Kind bigraphs — Static theory

Shane O'Conchuir

Foundations and Methods Group, Department of Computer Science
Trinity College Dublin, Ireland
Shane.OConchuir@cs.tcd.ie

Abstract

We present a refinement, suggested by Jensen and Milner under the term *kind*, on the pure place graphs in bigraph theory. We duly name the result *kind place graphs*. 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. We classify its idem pushouts and define the static theory for the variation. We then combine kind place graphs with pure link graphs in the usual way to achieve kind bigraphs and we reason about their static theory. Next, we develop a new *useful link-sorting*, named *tile-sorting*, for our purposes. Finally, we represent a sub-s-category of kind bigraphs, the *fitting* kind bigraphs, as a *useful place-sorting* and use known results to derive a dynamic behaviour for these bigraphs.

Keywords: bigraph, pushout, kind, place-sorting, link-sorting

Contents

1 Introduction

This report details a generalisation of pure bigraphs, whose theory has been developed by Leifer [8], Milner [11] and Jensen and Milner [7]. Our generalisation is on the pure bigraphs as defined in [7]. We assume familiarity with the theory of pure bigraphs throughout this report but, for convenience, some of the original definitions are repeated in the appendix.

Bigraph theory takes many notions from previous work in various process calculi. Its two (hence 'bi') basic structures represent mobile connectivity and mobile locality in an orthogonal manner. These structures may be separately altered and recombined, in a sensible manner, without breaking the central theory. The orthogonality between them may also be relaxed to some degree. In fact, once any changes to the pure theory satisfy certain properties, a different theory, tailored to some particular need or goal, may sometimes be realised. Bigraph theory therefore seems to have a great power as a general framework in which to study mobility.

Bigraph theory also rests upon category theory. Many useful tools and intuitions from category theory may then be applied in this setting. For example, imagine we have sensibly altered pure bigraph theory. We will then have a functor from some (pre)category based on this alteration to a pure (pre)category based on the pure theory. Properties of our altered theory can then be proved based on the category theoretic properties that this functor has. Category theory can be used as a tool to let us play around with the pure theory and verify that we have created something sensible.

Some of the existing alterations to pure bigraph theory are discussed in the next section. In this report, we are interested in generalising place graph atomicity. As pure place graphs and link graphs are essentially independent structures, we are able to concentrate our initial efforts solely on place graphs. This independence greatly simplified the technical process, as did some basic categorical notions.

1.1 Motivation

Previous generalisations of pure bigraphs seem to have been aimed at encoding various existing process calculi. Binding bigraphs were introduced in [7] where the notion of binders allowed an encoding of the asynchronous π-calculus. Binding bigraphs were generalised by Milner [13] to local bigraphs where names can be located at many places. This facilitated an encoding of the λ-calculus. Another generalisation by Leifer and Milner [9] introduced the notion of sorting on link graphs. These sorted link graphs were used for an encoding of Petri nets — place graphs were not needed as Petri nets do not use the notions of nesting and name scoping. Milner then introduced place-sorting [14], allowing an encoding of finite pure CCS. Place-sorting has also been proposed by Jensen and Milner [6] as a tool for encoding the full π-calculus using bigraphs.

So to date, most of the bigraph literature concerns encoding an existing calculus within this framework. Another possible use of bigraph theory is to model other abstract systems. There are at least two strategies for this form of modelling. In the usual strategy, one creates a syntax which defines the set of terms of a language which models the abstract system. A translation from this language into bigraphs is then given and only those bigraphs which are images of the translation are studied. An alternative strategy which we propose is to build a model directly as a bigraphical reactive system without the use of an interim language.

Using this second strategy, the modelling of real-life objects as controls poses a problem. Some expressiveness is needed to specify that certain controls cannot contain certain other controls. For example, we may wish to express that a building can contain a person but not that a person can contain a building. This notion of restricting place graphs (and also link graphs) was identified by Milner early in the history of bigraphs [11] as something which would be likely to be needed in significant applications. The general setting for the solution to this problem was introduced in [14] where the notion of *place sorting* was formally defined. In this recent work, the importance of restriction place graphs is repeated — *"in significant applications we are quite likely to employ a rich signature"*[14]. Leifer and Milner have also recently developed a theory of sorting for link graphs [9].

We begin this report with by introducing a generalisation of place graphs which allows us to specify which controls can be contained in others. We do not initially present it as a place-sorting, but later show that it is in fact an example of a place-sorting. We feel that it may be useful for directly modelling certain systems where some sort of containment relation is necessary – for example, the abstract model of a sentient built environment mentioned in [14].

We present a notion of *kind*, suggested by Jensen and Milner [7], which allows us to state the set of controls that one control may contain at top level. In terms of bigraphs, this lets us state that a node of control K can be the parent of nodes of control K', K'', *etc.* Note that this restriction is weak in that it only affects parent-child relationships. We can specify that a node of control K cannot have a child of control K' but, besides children, it does not enable us to specify that a node of control K cannot have *any ancestor* nodes of type K'. The stronger restriction on ancestry is discussed in Section 4.4 and Appendix B but is not investigated here.

We feel that these *kind bigraphs* (named after the suggestion in [7]) are useful in certain models. In particular, they should be applicable to models based on containment in the physical world. As an example, take the abstract model depicted in Figure 1 with four types of entities and the relationships between them. This model can be represented by sets for the entities and functions for the relationships, where the dotted relationships are represented by function composition. Instances of this model can also be represented as kind bigraphs as we will discuss below. It is at least for these sorts of hierarchies, where worrying about one level of nesting is sufficient, that kind bigraphs seem useful. Our definition of kind signature will not, however, restrict us to such hierarchies — indeed, doing so would prevent us to model many systems which are possible in pure bigraphs.

1.2 Expressiveness of kind bigraphs

Before introducing the theory, we present some informal examples of kind bigraphs to demonstrate their expressiveness.

Figure 1 is an example of an abstract structured model. We construct a kind signature to encode this model as follows: define the signature $\mathcal{K} =$

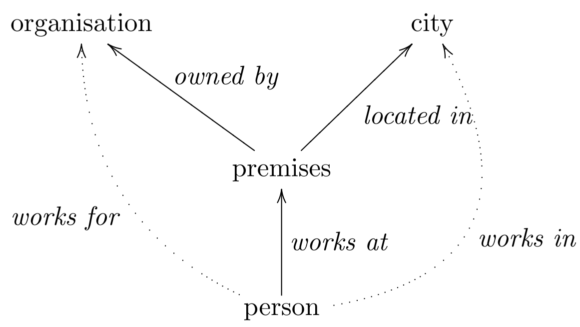


Figure 1: An abstract model of workers

$\{\mathbf{organisation}, \mathbf{city}, \mathbf{premises}, \mathbf{person}\}$ and a mapping

$$\begin{array}{rcl}\mathbf{organisation} & \mapsto & \{\mathbf{premises}\}\\ \mathbf{city} & \mapsto & \{\mathbf{premises}\}\\ \mathbf{premises} & \mapsto & \{\mathbf{person}\}\\ \mathbf{person} & \mapsto & \emptyset\end{array}$$

which states which kinds of control each control can contain. We will not define the *ar* map or state whether controls are active or passive for this example. Each bigraph in a s-category based on this signature will respect the hierarchy depicted in the figure. This notion of respecting a containment relation is one way in which kind bigraphs are more expressive (statically) than pure bigraphs. Kind bigraphs allow us to express some form of structure in place graphs.

Such static properties may be of some use on their own but they are more useful in a dynamic system. One aim of kind bigraphs is to be able to capture changes in such a model as Figure 1 *e.g.* 'Tom moves to a new premises', 'Bob gets fired', with bigraphical reaction rules which ensures that the reactum is a valid encoding of a model, *i.e.* that it respects the structure. This is not just to say that a parametric reaction rule itself respects this structure, but also that any ground reaction rules derived from the parametric reaction rule are structurally sound. This is another form of expressiveness we expect to gain from kind bigraphs — that the structure expressed by the signature is preserved through reaction.

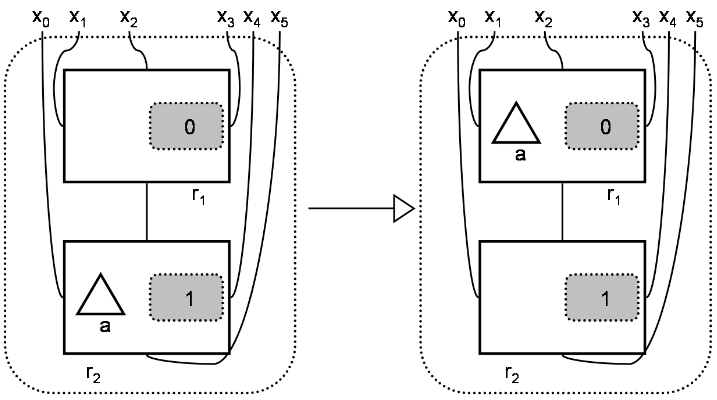
Presumably, the most useful property of kind bigraphs is the extra expressiveness they allow when defining reaction rules. Let $\mathcal{K} = \{\mathbf{r}, \mathbf{a}, \mathbf{e}\}$ where the elements represent rooms, agents and enemies respectively, $ar(\mathbf{r}) = 4$ and $ar(\mathbf{a}) = ar(\mathbf{e}) = 0$. Define *kind* as

$$\begin{array}{rcl}\mathbf{r} & \mapsto & \{\mathbf{a}, \mathbf{e}\}\\ \mathbf{a} & \mapsto & \emptyset\\ \mathbf{e} & \mapsto & \emptyset.\end{array}$$

We use this signature to model a system with rooms containing agents and enemies, where the rooms can be connected via edges (doors). We can think of this as a tile-based game where the tiles are square-shaped[1]. We now add reaction

[1]A model of such a game would also employ sorted link graphs [9] with four sorts (north,

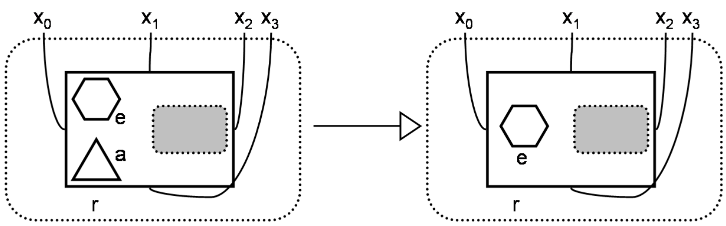
rules to make agents mobile. A ‘move north’ rule is shown in Figure 2. In the



$$(/l|\, {}^{\vec{x}}\!/\!_{\vec{x}}\, |\square)(\mathbf{r}_{x_2x_3lx_1}\square_0|\mathbf{r}_{lx_4x_5x_0}(\mathbf{a}|\square_1)) \;\rightarrow\!\!\rhd\; (/l|\, {}^{\vec{x}}\!/\!_{\vec{x}}\, |\square)(\mathbf{r}_{x_2x_3lx_1}(\mathbf{a}|\square_0)|\mathbf{r}_{lx_4x_5x_0}\square_1)$$

Figure 2: An agent moves between two rooms

pure theory, such rules would let agents move around the board regardless of what was contained inside the holes of the source and destination square. However, using kind bigraphs we can further specify the rule by stating what kinds of controls the holes can contain as follows; let $R, R' : \langle 2, (\{\mathbf{a}, \mathbf{e}\}, \{\mathbf{a}\}), \emptyset\rangle \rightarrow \langle 1, (\{r\}), \vec{x}\rangle$. The rule specifies that the second site, the sibling site of the agent, can only contain other agents and not enemies. We can describe this rule as ‘an agent may move north if there is a room to the north and no enemies in the current room’. This could be combined with a rule as in Figure 3 stating



$$\mathbf{r}_{x_1x_2x_3x_0}(\mathbf{a}|\mathbf{e}|\square) \;\longrightarrow\!\!\rhd\; \mathbf{r}_{x_1x_2x_3x_0}(\mathbf{e}|\square)$$

Figure 3: An enemy eliminates an agent

that if an agent occupies the same room as an enemy, the agent is eliminated. We have now described (informally) a bigraphical reactive system (Brs) which allows agents to move freely until they run into enemies. If we compare this kind reactive system to the pure reactive system, we see that a kind reactive system can limit or restrict the set of possible reactions that can take place by imposing

south, east and west) where every open link contains exactly one port and every closed link at most two points of opposite direction.

conditions on a redex that a pure reactive system can not. The expressiveness of kind bigraphs with respect to reactions can thus be described as follows: by specifying which controls can lie directly under the roots of parameters, parametric rules in a kind bigraphical reactive system can be used to specify ground rules where the *absence* of certain kinds of controls directly under those roots is guaranteed[2].

More examples of kind reaction rules can be found in Section 4.3.

1.3 Structure of this report

In Section 2, we introduce kind place graphs, a generalisation of place graphs. We explore the static theory of 'KPG, the s-category of kind place graphs, up to the point where we combine kind place graphs and pure link graphs to obtain kind bigraphs, where the nesting and linking structures remain orthogonal as for pure bigraphs. We also introduce some new concepts particular to kind bigraphs. These are *inflations*, *deflations* and the notion of *fitting*. It will transpire that the interface of a relative pushout (RPO) is a fitting interface.

There is a forgetful functor from 'KPG to pure place graphs. We show that this functor creates relative pushouts (RPOs). The proof is routine and — as a further example of a proof of RPO creation — adds to the large body of variants of bigraphs which have this desirable property. We then classify the idem pushouts (IPOs) of an s-category of kind place graphs and define hard kind place graphs in the same manner as hard *pure* place graphs.

Section 3 discusses kind bigraphs — pure bigraphs where the place graphs are kind. We present all the necessary definitions and propositions to complete the static theory of kind bigraphs. We conjecture that an axiomatisation of kind bigraphs may use inflations and certain fitting bigraphs (ions and atoms particularly). Whether such an axiomatisation is possible is not answered here.

Section 4 begins with a brief look at link-sorting [9]. We define a specific *useful link-sorting* ('useful' has a particular meaning here) to help formalise our examples later in the section. This link-sorting is called *tile-sorting* and is one of the few published 'useful' link-sortings to date. We show that homomorphic place-sortings [14] can be encoded with kind bigraphs. We then reformulate kind bigraphs as a place-sorting and use results from [14] to derive some of the dynamic theory of kind bigraphs. In particular, we focus on fitting bigraphs, a special case of kind bigraphs which satisfy additional properties. We give more examples of the expressiveness that kind bigraphs lends to reaction rules and we discuss how the use of fitting bigraphs restricts the Brss we can define. We end the section with some discussion on how to avoid these restrictions.

Appendix A contains the original notation and definitions from [7] for convenience. Appendix B discusses an alternative formulation for kind bigraphs which is less general but may be more expressive for certain applications. Appendix C contains the proof of a tile-sorting lemma.

[2]The stronger kind bigraphs discussed in the appendix would express a stronger property, namely: by specifying which controls lie under the roots of parameters, parametric rules in a stronger kind bigraphical reactive system can be used to specify ground rules where the absence of certain kinds of controls under those roots is guaranteed.

1.4 Notation

For most of the report, we use the notation (and abuses of same) of [ibid.], summarised in Appendix A.1. The bulk of the definitions and theorems presented here are taken verbatim or with minor changes from [ibid.], mainly in the order presented therein. In particular, any references to pure bigraphs, pure place graphs or pure link graphs pertain to that text. Where a section here is the same, or based on, a section from that text, we will label that section with its original numbering *e.g.* **Definition 1.1 (kind signature [6.1])** defines a signature for kind bigraphs based on Definition 6.1 (pure signature) of [7]. Most of the original definitions on which this work is based are copied from [ibid.] in Appendix A.2 for quick reference. The constructions of relative pushouts for pure place graphs and link graphs and the proofs regarding the constructions are not reproduced in this report.

We adopt a labelling convention in this paper as follows. In Section 2, we use H, I, J and L to denote kind interfaces and h, m, n and l respectively to denote the width of those interfaces so that $H = \langle h, \vec{C_H} \rangle, I = \langle m, \vec{C_I} \rangle, J = \langle n, \vec{C_J} \rangle$ and $L = \langle l, \vec{C_L} \rangle$. When we are discussing sites or roots s, r and t, we assume $s \in m, r \in n$ and $t \in L$. In Section 3, we extend the convention to names, letting $H = \langle h, \vec{C_H}, W \rangle, I = \langle m, \vec{C_I}, X \rangle, J = \langle n, \vec{C_J}, Y \rangle$ and $L = \langle l, \vec{C_L}, Z \rangle$.

We will talk about some enriched signatures $\mathcal{K}$ which have more structure than a pure signature. However, we denote the pure signature underlying an enriched signature $\mathcal{K}$ simply as $\mathcal{K}$ as well. It should be clear which type of signature (pure or enriched) is under discussion.

2 Kind place graphs

In this report we define a generalisation of pure bigraphs. The generalisation is solely on the notion of place graph atomicity and so placing and linking remain independent structures. We therefore proceed by introducing kind place graphs. In the next section, we rejoin the stuctures and discuss kind bigraphs.

2.1 Elementary notions

Definition 2.1 (kind signature [6.1]). *A* kind signature $\mathcal{K}$ *is like a pure signature except that it also provides a* function $kind : \mathcal{K} \to \mathcal{P}(\mathcal{K})$*. 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'*.*

This definition allows us to specify the set of controls that another control may contain (in [7], this was described as assigning a *kind* to each control of a node.) It is a simple generalisation of pure signatures. An atomic control K is one which can not contain any others *i.e.* $kind(K) = \emptyset$. A pure non-atomic control K is represented by a control which can contain all controls *i.e* $kind(K) = \mathcal{K}$. This definition also allows us to specify a control which can contain *some* other controls. The *kind* function maps a control K to the set of controls it can contain.

Definition 2.2 (kind interface [7.1]). *A kind interface* $I = \langle m, \vec{C_I} \rangle$*, where the width* m *is a finite ordinal and* $\vec{C_I} = (C_{I,0}C_{I,1} \cdots C_{I,m-1})$ *is a vector of subsets of* $\mathcal{K}$ *where* $C_{I,s}$ *is the set of controls that* s *can be a parent of.* $I^{\mathrm{u}} = m$ *denotes the pure place graph interface* underlying I*.*

We will usually abbreviate $C_{I,s}$ to C_s when it is clear what interface s belongs to.

Definition 2.3 (kind place graph [7.2]). *A kind place graph* $G : I \to J$ *consists of an underlying pure place graph* $G^{\mathrm{u}} : I^{\mathrm{u}} \to J^{\mathrm{u}}$ *with extra structure as follows. G must satisfy the following:*

Kind rules:

KR1 *if* $r = prnt(v)$ *then* $ctrl(v) \in C_r$

KR2 *if* $r = prnt(s)$ *then* $C_s \subseteq C_r$

KR3 *if* $v = prnt(v')$ *then* $ctrl(v') \in kind(ctrl(v))$

KR4 *if* $v = prnt(s)$ *then* $C_s \subseteq kind(ctrl(v))$

where r *is a root,* s *a site,* v *and* v' *are nodes and no node of control* K *can be a parent of a site when* $kind(K) = \emptyset$*.*

The rules state what we would expect. A root (resp. node) may only be parent of a node when the interface (resp. signature) allows it. A root (resp. node) may only be parent of a site when it can contain at least what the site can contain.

The s-category $\mathcal{K}$Kpg($\mathcal{K}$) of (concrete) kind place graphs over $\mathcal{K}$ has kind interfaces as objects and kind place graphs as arrows. Composition and identities are defined as for the underlying pure place graphs. Identities respect the kind rules.

Proposition 2.4. *Composition in* $\text{'KPG}(\mathcal{K})$ *respects the kind rules*

Proof. Given two kind place graphs $A : I \to J$ and $B : J \to K$, we have to check that the *prnt* map of $B^{\mathrm{u}} \circ A^{\mathrm{u}}$ respects the kind rules. Recall that $ctrl_{B \circ A} = ctrl_A \uplus ctrl_B$. The proof is split on the cases where insertion of roots into sites occurs:

1. in B, a site r lies beneath a root t ($t = B(r)$). By KR2 in B,

$$C_r \subseteq C_t \qquad \text{(I)}$$

   (a) in A, a node v lies beneath the root r ($r = A(v)$). We have $t = (B^{\mathrm{u}} \circ A^{\mathrm{u}})(v)$.
   By KR1 in A, $ctrl_A(v) \in C_r$. By (I), we have $ctrl_A(v) \in C_t$ and KR1 holds in $B \circ A$.

   (b) in A, a site s lies beneath the root r ($r = A(s)$). We have $t = (B^{\mathrm{u}} \circ A^{\mathrm{u}})(s)$.
   By KR2 in A, $C_s \subseteq C_r$. By (I), we have $C_s \subseteq C_t$. Hence, KR2 holds in $B \circ A$.

2. in B, a site r lies beneath a node v ($v = B(r)$). By KR4 in B,

$$C_r \subseteq kind(ctrl_B(v)) \qquad \text{(II)}$$

   (a) in A, a node v' lies beneath the root r ($r = A(v')$). We have $v = (B^{\mathrm{u}} \circ A^{\mathrm{u}})(v')$.
   By KR1 in A, $ctrl_A(v') \in C_r$. By (II), we have $ctrl_A(v') \in kind(ctrl_B(v))$ and KR3 holds in $B \circ A$.

   (b) in A, a site s lies beneath the root r ($r = A(s)$). We have $v = (B^{\mathrm{u}} \circ A^{\mathrm{u}})(s)$.
   By KR2 in A, $C_s \subseteq C_r$. By (II), we have $C_s \subseteq kind(ctrl_B(v))$. Hence, KR4 holds in $B \circ A$.

□

Definition 2.5 (tensor product [7.5]). *The tensor product of two kind interfaces* $I = \langle m, \vec{C_I} \rangle$ *and* $J = \langle n, \vec{C_J} \rangle$ *is*

$$I \otimes J = \langle m + n, \vec{C_I}\vec{C_J} \rangle$$

The tensor product $G : I \to J$ *of two kind place graphs* $G_i : I_i \to J_i (i = 0, 1)$ *is defined when the two node sets are disjoint in the same manner as the tensor product of two pure place graphs. The kind rules hold and we have* $G^{\mathrm{u}} = G_0^{\mathrm{u}} \otimes G_1^{\mathrm{u}}$*. Thus* $\mathcal{U}$ *preserves tensor product.*

The definitions of *barren*, *sibling*, *active* and *passive* remain as in the pure theory.

The functor $\mathcal{U} : \text{'KPG}(\mathcal{K}) \to \text{'PLG}(\mathcal{K})$ sends each I to I^{u} and each G to G^{u}. $\mathcal{U}$ is a forgetful functor in that it forgets about the vectors $\vec{C_I}$ in the interfaces — the extra structure. $\mathcal{U}$ is surjective on objects as given a pure interface $I' = m$, we can always form a kind interface $I = \langle m, (\emptyset \cdots \emptyset)\rangle$ such that $I^{\mathrm{u}} = I'$. Given a pair of kind place graphs A and B in the same homset where $A \neq B$, we have that their *prnt* maps are different. Therefore, A^{u} and B^{u} have different *prnt* maps and thus $A^{\mathrm{u}} \neq B^{\mathrm{u}}$. Hence, $\mathcal{U}$ is also faithful. This last fact simplifies many of the following proofs.

Proposition 2.6 (isomorphisms in kind place graphs [7.4]). *An arrow $\iota : I \to J$ in ´Kpg is an isomorphism iff it has no nodes, its parent map is a bijection and this bijection respects kind meaning that if $\iota(s) = r$ then $C_{I,s} = C_{J,r}$.*

Proof. $\Rightarrow$. We just show the necessity of the last property, that the parent bijection respects kind. Assume we have an isomorphism $\iota : \langle m, (C_{I,0} \cdots C_{I,m-1})\rangle \to \langle m, (C_{J,0} \cdots C_{J,m-1})\rangle$ with no nodes and a bijective *prnt* map. This implies that an isomorphism $\iota^{-1} : \langle m, (C_{J,0} \cdots C_{J,m-1})\rangle \to \langle m, (C_{I,0} \cdots C_{I,m-1})\rangle$ exists. We illustrate the proof with an example, depicted in Figure 4 below. In ι, by KR2,

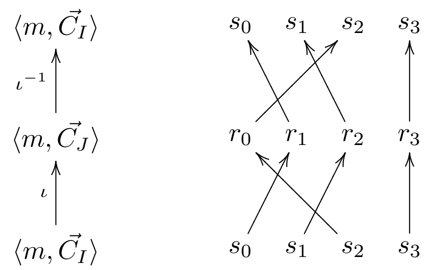


Figure 4: An isomorphism in ´Kpg and an example of two *prnt* maps

$C_{s_0} \subseteq C_{r_1}$. In ι^{-1}, by KR2, $C_{r_1} \subseteq C_{s_0}$. Hence, $C_{s_0} = C_{r_1}$. More generally, in an isomorphism $\iota : \langle m, \vec{C_I}\rangle \to \langle m, \vec{C_J}\rangle$, $C_{I,s} = C_{J,\iota(s)}$.

$\Leftarrow$. An inverse is found trivially. □

$\mathcal{U}$ preserves isomorphisms but does not reflect them.

Example 2.7 ($\mathcal{U}$ does not reflect isomorphisms). *Take the kind place graph $G : \langle 1, (\{\})\rangle \to \langle 1, (\{0, K\})\rangle$ with no nodes, where the site cannot contain any controls and the root can contain a control K. This is not an iso in ´Kpg. However, $\mathcal{U}(G)$ is an iso in ´Plg.*

Unlike isomorphisms, epis and monos remain unchanged from the pure place graph theory.

Proposition 2.8 (epis [7.6]). *In ´Kpg, a kind place graph is epi iff no root is barren.*

Proof. ($\Leftarrow$) Given the diagram

$$I \xrightarrow{A} J \underset{B_1}{\overset{B_0}{\rightrightarrows}} K,$$

let no root in A be barren and $B_0 \circ A = B_1 \circ A$. Then $(B_0 \circ A)^{\mathrm{u}} = (B_1 \circ A)^{\mathrm{u}}$. A^{u} is epi so $B_0^{\mathrm{u}} = B_1^{\mathrm{u}}$. Hence, $prnt_{B_0} = prnt_{B_1}$ and so $B_0 = B_1$. Hence, A is epi.

(⇒) Let $A : I \to \langle n, (C_0 \cdots C_{n-1})\rangle$ be epi. Assume a root $r \in n$ in A is barren. Define $B_0 : \langle n, (C_0 \cdots C_{n-1})\rangle \to \langle n+1, (C_0 \cdots C_{n-1} C_r)\rangle$ as a place graph with no nodes where the *prnt* map is defined as $B_0(i) = i$. Define $B_1 : \langle n, (C_0 \cdots C_{n-1})\rangle \to \langle n+1, (C_0 \cdots C_{n-1} C_r)\rangle$ as a place graph with no nodes where the *prnt* map is defined by $B_1(i) = i$ when $i \neq r$ and $B_1(r) = n$. Then $B_0 \circ A = B_1 \circ A$ since the barren root r disappears in the composition but $B_0 \neq B_1$. Hence, A is not epi and so our assumption is false. □

Proposition 2.9 (monos [7.6]). *In ´Kpg, a kind place graph is mono iff no two sites are siblings.*

Proof. (⇐) Given the diagram

$$I \overset{B_0}{\underset{B_1}{\rightrightarrows}} J \xrightarrow{A} K,$$

let no two sites in A be siblings and $A \circ B_0 = A \circ B_1$. Then $(A \circ B_0)^{\mathrm{u}} = (A \circ B_1)^{\mathrm{u}}$. A^{u} is mono so $B_0^{\mathrm{u}} = B_1^{\mathrm{u}}$. Hence, $prnt_{B_0} = prnt_{B_1}$ and so $B_0 = B_1$. Hence, A is mono.

(⇒) Let $A : \langle n, (C_0 \cdots C_{r_0} \cdots C_{r_1} \cdots C_{n-1})\rangle \to K$ be mono. Assume two sites r_0 and r_1 are siblings so $A(r_0) = A(r_1)$. Define B_0 as the identity kind place graph. Define

$$B_1 : \langle n, (C_0 \cdots C_{r_1} \cdots C_{r_0} \cdots C_{n-1})\rangle \to \langle n, (C_0 \cdots C_{r_0} \cdots C_{r_1} \cdots C_{n-1})\rangle$$

as a place graph with no nodes where the *prnt* map is defined by $B_1(r) = r$ when $r \notin \{r_0, r_1\}$ and $B_1(r_i) = r_{\bar{\imath}}, i \in \{0, 1\}$. Then $B_0 \circ A = B_1 \circ A$ by the definition of composition but $B_0 \neq B_1$. Hence, A is not mono and so our assumption is false. □

Corollary 2.10. *$\mathcal{U}$ both preserves and reflects epis and monos*

Note that if a kind place graph is both an epimorphism and a monomorphism then it may not be an isomorphism. Thus, ´Kpg is not a balanced category.

The following definition will be used when we discuss changing the inner or outer face of a kind bigraph.

Definition 2.11 (pointwise disjoint vector union, pointwise vector difference). *Given two vectors $\vec{A}$ and $\vec{B}$ of subsets of a set X with equal length l, we define the pointwise disjoint vector union $\vec{A} \mathbin{\dot{\uplus}} \vec{B}$ as*

$$\vec{A} \mathbin{\dot{\uplus}} \vec{B} \stackrel{\text{def}}{=} (C_0 C_1 \cdots C_l) \text{ where } C_i = A_i \uplus B_i.$$

On any two such vectors, we also define the pointwise vector difference $\vec{A} \mathbin{\dot{\setminus}} \vec{B}$ as

$$\vec{A} \mathbin{\dot{\setminus}} \vec{B} \stackrel{\text{def}}{=} (C_0 C_1 \cdots C_l) \text{ where } C_i = A_i \setminus B_i.$$

We now introduce the concept of 'fitness' for kind place graphs. Note that for any given kind place graph A, the pure place graph A^{u} may have other preimages under $\mathcal{U}$ *i.e.* there may be some A' in a different homset to A where $A'^{\mathrm{u}} = A^{\mathrm{u}}$. When constructing RPOs and IPOs, we will need to identify certain preimages under $\mathcal{U}$ which satisfy some notion of 'least'. We present this below.

Definition 2.12 (fitting). *A kind place graph $G : I \to \langle n, \vec{C_J}\rangle$ is said to be* fitting *if $\vec{C_J}$ is the vector of least subsets of $\mathcal{K}$ which satisfies KR1 and KR2. We will call a fitting kind place graph a fitting place graph, omitting 'kind'.*

In some sense, a fitting place graph has the least outer interface that satisfies the kind rules. Fitting place graphs obey some nice properties. It can be shown simply that the composition of two fitting place graphs is a fitting place graph (however, if $B \circ A$ is fitting, B and A are not both fitting in general). Further, $A \otimes B$ is fitting iff both A and B are fitting. Thus, fitting place graphs form a monoidal sub-s-category 'Fpg($\mathcal{K}$) of 'Kpg($\mathcal{K}$).

Definition 2.13 (fitting interface, fitting bound). *Given a set of place graphs $\{A_0, \ldots, A_{n-1}\}$ with a common codomain $I = \langle m, \vec{C_I}\rangle$, we say I is a* fitting interface *for $A_0, \ldots, A_{n-1}$ if $\vec{C_I}$ is the vector of least subsets of $\mathcal{K}$ which satisfies KR1 and KR2 for all $A_i \in \{A_0, \ldots, A_{n-1}\}$.*

If $\vec{B}$ is a bound for $\vec{A}$ and the outer interface of $\vec{B}$ is a fitting interface for $\vec{B}$, then we say that $\vec{B}$ is a fitting bound *for $\vec{A}$.*

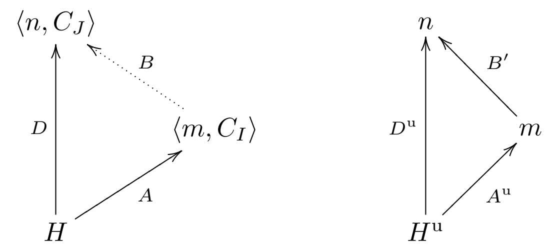
This notion of fitting interface could also be described as "least outer interface such that...". This is a useful description and it fits with the intuition that colimits are special constructions with this 'least such' property as fitting bounds will turn out to be important in constructing RPOs for kind bigraphs.

Since fitting place graphs form a sensible sub-s-category of 'Kpg, one may wonder why we currently concentrate on arbitrary kind place graphs. The reason is that certain reaction rules may be expressed in 'Kpg but not in 'Fpg, due to the fact that in the latter, a redex and reactum of a rule must both have fitting place graphs *and* have the same outer face. To keep our options open, we will continue to define the static theory of kind bigraphs. We will return to 'Fpg in Section 4.2 and show that it is an example of a place-sorting, defined by Milner in [14]. It turns out 'Fpg is a certain, well-behaved place-sorting and we can derive some of its dynamic theory 'for free' by invoking results from [14].

For now, we prepare for that section by proving some other nice properties of fitting place graphs in the following lemmas.

Lemma 2.14 (some properties of fitting place graphs).

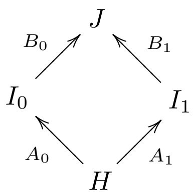
1. *Given a kind place graph $D : H \to \langle n, C_J\rangle$ and a* fitting *place graph $A : H \to \langle m, C_I\rangle$, let $B' : m \to n$ be a pure place graph such that $D^{\mathrm{u}} = B' \circ A^{\mathrm{u}}$. Then there exists a unique kind place graph $B : \langle m, C_I\rangle \to \langle n, C_J\rangle$ such that $prnt_B = prnt_{B'}$ (*i.e. *$B^{\mathrm{u}} = B'$) and $D = B \circ A$.*



2. *Let $D = B \circ A$ and D and A be fitting place graphs. Then B is a fitting place graph.*

3. *Let* $A : H \to \langle m, C_I\rangle$ *and* $D : H \to \langle n, C_J\rangle$ *be two* fitting *place graphs with common domain and let* $B' : m \to n$ *be a pure place graph such that* $D^{\mathrm{u}} = B' \circ A^{\mathrm{u}}$*. Then there exists a unique fitting place graph* $B : \langle m, C_I\rangle \to \langle n, C_J\rangle$ *such that* $B^{\mathrm{u}} = B'$ *and* $D = B \circ A$*.*

4. *Given a consistent pair of fitting place graphs* $\vec{A} : H \to \vec{I}$ *with common domain, let* $\vec{B} : \vec{I} \to J$ *be a bound for* $\vec{A}$ *such that* $\vec{B}$ *is a fitting bound for* $\vec{A}$*. Then both* B_0 *and* B_1 *are fitting place graphs.*



Proof. 1. We prove that B, with $prnt_B = prnt_{B'}$, is a kind place graph. $D = B \circ A$ follows trivially and uniqueness follows from the faithfulness of $\mathcal{U}$.

Note that $prnt_D = (\mathrm{id}_{V_A} \uplus prnt_B) \circ (prnt_A \uplus \mathrm{id}_{V_B})$ and $ctrl_D = ctrl_A \uplus ctrl_B$. The proof is broken into four main cases where we check that KR1-KR4 hold in B, knowing that KR1-KR4 hold in A and D.

KR1 Let $t = B(v)$. Then $t = D(v)$ and by KR1 in D, $ctrl_D(v) \in C_t$. Hence $ctrl_B(v) \in C_t$.

KR2 Let $t = B(r)$. We must show that when $K \in C_r$, $K \in C_t$. A is fitting and so the following two subcases cover all the elements of C_r.

- Let $r = A(s)$. By KR2 in A, if $K \in C_s$ then $K \in C_r$. By composition, $t = D(s)$. By KR2 in D, if $K \in C_s$ then $K \in C_t$.
- Let $r = A(v)$. By KR1 in A, $ctrl_A(v) \in C_r$. By composition, $t = D(v)$. By KR1 in D, $ctrl_D(v) = ctrl_A(v) \in C_t$.

KR3 Let $v = B(v')$. Then $v = D(v')$ and by KR3 in D, $ctrl_B(v') = ctrl_D(v') \in kind(ctrl_D(v)) = kind(ctrl_B(v))$.

KR4 Let $v = B(r)$. As above, we must show that when $K \in C_r$, $K \in kind(ctrl_B(v))$. A is fitting and so the following two subcases cover all the elements of C_r.

- Let $r = A(s)$. By KR2 in A, if $K \in C_s$ then $K \in C_r$. By composition, $v = D(s)$. By KR4 in D, if $K \in C_s$ then $K \in kind(ctrl_D(v)) = kind(ctrl_B(v))$.
- Let $r = A(v')$. By KR1 in A, $ctrl_A(v') \in C_r$. By composition, $v = D(v')$. By KR3 in D, $ctrl_A(v') = ctrl_D(v') \in kind(ctrl_D(v)) = kind(ctrl_B(v))$.

2. We will prove that when $K \in C_t$, where t is a root of both B and D, $K \in C_t$ is required for KR1 and KR2 to hold in B.

Let $K \in C_t$. Since D is fitting, we know that one or both of the following hold:

$$
\begin{array}{lll}
t = D(v) & \text{where } K = ctrl_D(v) & (i) \\
t = D(s) & \text{where } K \in C_s & (ii)
\end{array}
$$

or both. We examine these cases.

(i) Let $t = D(v)$ and $K = ctrl_D(v)$. The node v is located in either B or A. If v is in B, then $t = B(v)$ and $K = ctrl_B(v) \in C_t$ is necessary in B. If v is in A, then $r = A(v)$ and $t = B(r)$. By KR1 in A, $K = ctrl_A(v) \in C_r$ and by KR2 in B we need $K \in C_t$ to be true.

(ii) Let $t = D(s)$ and $K \in C_s$. Then $r = A(s)$ and $t = B(r)$. By KR2 in A, $K \in C_r$. By KR2 in B, we require $K \in C_t$.

3. By Lemma 2.14.1 above, there exists such a unique kind place graph B. By Lemma 2.14.2 above, B is fitting.

4. We follow the usual conventions and let the nodes of A_0 be V_0 and V_2, the nodes of A_1 be V_1 and V_2, where V_2 are the nodes shared by A_0 and A_1. The node sets of B_0 are then $V_1 - V_2$ and V_3 and the node sets of B_1 are $V_0 - V_2$ and V_3, where V_3 is some set of fresh nodes. It is given that $B_0 \circ A_0 = B_1 \circ A_1$. Our strategy is as follows.

   We know that J is a fitting interface for the pair $\vec{B}$. This means that when $K \in C_t$, for some place $t \in J$, then either $K \in C_t$ is necessary for KR1 and KR2 to hold in B_0, $K \in C_t$ is necessary for KR1 and KR2 to hold in B_1 or $K \in C_t$ is necessary for KR1 and KR2 to hold in both B_0 and B_1. We will assume w.l.o.g. that if $K \in C_t$ then it is necessary for KR1 and KR2 to hold in B_0. We then show that it is also necessary that $K \in C_t$ in order that KR1 and KR2 holds in B_1. We can then infer that for each of B_0 and B_1, J is the least interface such that KR1 and KR2 hold and hence both place graphs are fitting.

   Let $K \in C_t$ and assume that this is necessary in order that KR1 or KR2 holds in B_0. We have two cases depending on which kind rule requires the condition.

   - Let $t = B_0(v)$, $ctrl(v) = K$. We have two subcases, depending on which set of nodes v belongs to.
     - $\mathbf{v} \in \mathbf{V_3}$
       Then $t = B_1(v)$ and $K \in C_t$ is necessary for KR1 to hold in B_1.
     - $\mathbf{v} \in \mathbf{V_1} - \mathbf{V_2}$
       Then $r_1 = A_1(v)$ and $t = B_1(r_1)$. By KR1 in A_1, $K \in C_{r_1}$. Thus, $K \in C_t$ is necessary for KR2 to hold in B_1.
   - Let $t = B_0(r_0)$, $K \in C_{r_0}$. Recall that A_0 is fitting. We have three subcases.
     - $\mathbf{r_0} = \mathbf{A_0}(\mathbf{v}), ctrl(\mathbf{v}) = \mathbf{K}, \mathbf{v} \in \mathbf{V_0} - \mathbf{V_2}$
       Then v is in B_1 and so $t = B_1(v)$. $K \in C_t$ is then necessary for KR1 to hold in B_1.
     - $\mathbf{r_0} = \mathbf{A_0}(\mathbf{v}), ctrl(\mathbf{v}) = \mathbf{K}, \mathbf{v} \in \mathbf{V_2}$
       Then v is in A_1 and so $r_1 = A_1(v)$ and $t = B_1(r_1)$. By KR1 in A_1, $K \in C_{r_1}$. Thus, $K \in C_t$ is necessary for KR2 to hold in B_1.
     - $\mathbf{r_0} = \mathbf{A_0}(\mathbf{s}), \mathbf{K} \in \mathbf{C_s}$
       Then $r_1 = A_1(s)$ and $t = B_1(r_1)$. By KR2 in A_1, $K \in C_{r_1}$. Thus, $K \in C_t$ is necessary for KR2 to hold in B_1.

That completes the case split and so we have that when $K \in C_t$ is required for KR1 or KR2 to hold in B_i, then it is required so that KR1 or KR2 holds in $B_{\bar{\imath}}$. Thus, B_0 and B_1 are fitting place graphs.

□

These lemmas will aid in proving that given a bound $\vec{D}$ for $\vec{A}$, where the four kind bigraphs are fitting, the RPO construction for kind bigraphs (introduced in the next section) will yield a kind RPO $(\vec{B}, B)$ for the square, and each bigraph in the RPO triple will be a fitting bigraph. Further, given a second candidate $(\vec{C}, C)$ for an RPO, the unique mediating kind arrow from $(\vec{B}, B)$ to $(\vec{C}, C)$ will be fitting. We will then have shown that the subcategory of fitting bigraphs creates RPOs, which is a very useful result.

We continue introducing some new concepts for kind place graphs called inflations and deflations. Given a preimage $A : I \to J$ of A^{u} under $\mathcal{U}$, there are some operations that we can perform on A to yield a place graph A' such that $A'^{\mathrm{u}} = A^{\mathrm{u}}$. These operations on A involve altering the inner or outer interface with respect to their respective vectors $\vec{C_I}$ and $\vec{C_J}$ and will make use of Definition 2.11.

Definition 2.15 (inflations, deflations). *Given an interface* $I = \langle m, \vec{C_I} \rangle$ *and a vector* $\vec{V}$ *of of subsets of* $\mathcal{K}$ *and of length* m*, we define the* inflation

$$\uparrow\!\vec{V}_I : \langle m, \vec{C_I} \rangle \to \langle m, \vec{C_I} \dot{\uplus} \vec{V} \rangle$$

when $V_i \cap C_{I,i} = \emptyset, i \in m$*, and the* deflation

$$\downarrow\!\vec{V}_I : \langle m, \vec{C_I} \mathbin{\dot{\setminus}} \vec{V} \rangle \to \langle m, \vec{C_I} \rangle$$

when $V_i \subseteq C_{I,i}, i \in m$*, as place graphs with no nodes and identity prnt maps.*

We now define related operations on a place graph $G : \langle m, \vec{C_I} \rangle \to \langle n, \vec{C_J} \rangle$.

Definition 2.16 (outer inflation, inner deflation). *Let* $\vec{V}$ *be a vector of subsets of* $\mathcal{K}$.

Let $\vec{V} = (V_0 \cdots V_{n-1})$ *such that* $V_i \cap C_{J,i} = \emptyset, i \in n$*. We define the* outer inflation $G \nwarrow\!\!\searrow \vec{V} \stackrel{\mathrm{def}}{=} \uparrow\!\vec{V}_J \circ G$.

Let $\vec{V} = (V_0 \cdots V_{m-1})$ *such that* $V_i \subseteq C_{I,i}, i \in m$*. We define the* inner deflation $G \searrow \vec{V} :\stackrel{\mathrm{def}}{=} G \circ \downarrow\!\vec{V}_I$.

These two operations seem natural for kind place graphs since they do not break the kind rules. We continue to define two partially-defined operations. These operations alter place graphs without using composition and seem more related to the *extension* operator, $\oplus$, introduced in [12].

Definition 2.17 (inner inflation, outer deflation). *Let* $\vec{V}$ *be a vector of subsets of* $\mathcal{K}$.

Let $\vec{V} = (V_0 \cdots V_{m-1})$ *such that* $V_i \cap C_{I,i} = \emptyset, i \in m$*. The* inner inflation $G \nearrow \vec{V} : \langle m, \vec{C_I} \dot{\uplus} \vec{V} \rangle \to \langle n, \vec{C_J} \rangle$ *is defined when the pointwise disjoint union of* $\vec{V}$ *and* $\vec{C_I}$ *does not violate KR2 or KR4 in* G.

Let $\vec{V} = (V_0 \cdots V_{n-1})$ *such that* $V_i \subseteq C_{J,i}, i \in n$*. The* outer deflation $G \swarrow\!\!\nearrow \vec{V} : \langle m, \vec{C_I} \rangle \to \langle n, \vec{C_J} \mathbin{\dot{\setminus}} \vec{V} \rangle$ *is defined when removing the subsets in* $\vec{V}$ *from* $\vec{C_J}$ *does not violate KR1 or KR2 in* G.

When the operations are defined, $prnt_{G \nearrow \vec{V}} = prnt_G$ *and* $prnt_{G \swarrow\!\nearrow \vec{V}} = prnt_G$.

Some examples are depicted in Figure 5. In the figure, we include the control of a node as a label directly below the node. The set above r_0 and below s_0 depicts the set of controls that r_0 and s_0 can repectively hold.

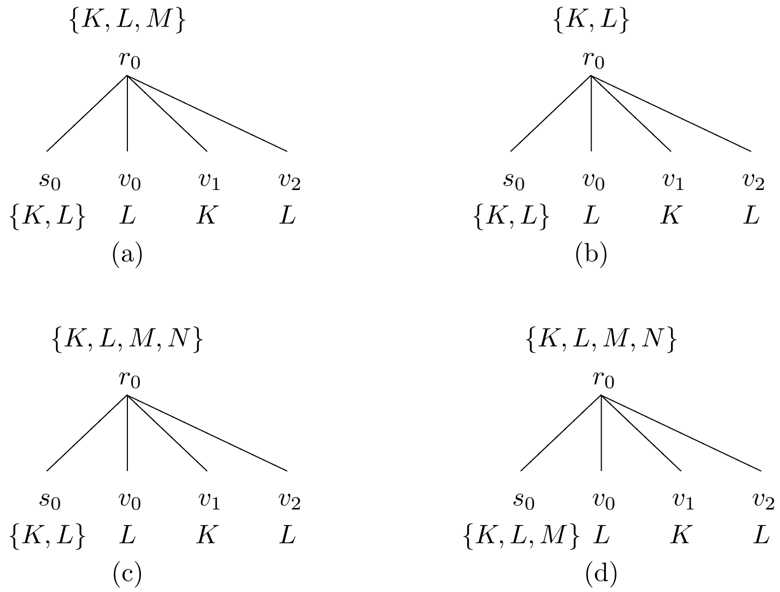


Figure 5: (a) A place graph G (b) Outer deflation $G\swarrow\!\!\!\!\nearrow\vec{V}, \vec{V} = (\{M\})$ (c) Outer inflation $G\nwarrow\!\!\!\!\searrow\vec{V}, \vec{V} = (\{N\})$ (d) Inner inflation $G\nearrow\vec{V}, \vec{V} = (\{M\})$

One use of these operations is to alter interfaces in order to allow composition. This idea is similar to adding a wiring, via tensor product, to a bigraph for the same means. They may play a part in the dynamic theory of kind bigraphs. Specifically, if we allow the redex of a reaction rule not to be fitting, then a bigraph may have to be composed with an inflation in order for a reaction rule to fire.

Inflations may also be used to decompose a place graph G into a composition $\uparrow\vec{V}_J \circ G'$ where G' is fitting. Finally, if a place graph G' is obtained by inflating or deflating G then $G'^{\mathrm{u}} = G^{\mathrm{u}}$.

2.2 RPOs and IPOs

We now show how to construct a kind RPO given a pure RPO in much the same manner as was done for binding bigraphs. It may help to note that in the pure theory, an RPO $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$ was constructed by leaving all the nodes common to both D_0 and D_1 as part of B. Thus, $\vec{B}$ was minimal and a mediating arrow to any candidate RPO was easily found. Our construction is necessarily similar except that we also require that the outer *interface* of $\vec{B}$ be minimal in some sense — we require that $\vec{B}$ is a fitting bound for $\vec{A}$.

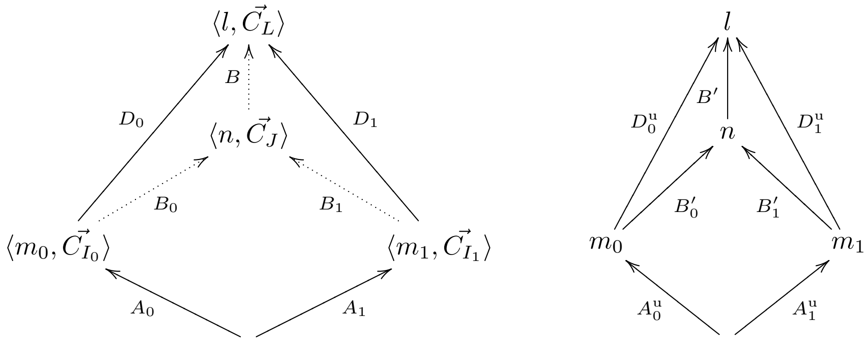
Construction 2.18 (building a kind RPO [11.4]).

Proof. Let $\vec{A} : H \to \vec{I}$ have a bound $\vec{D} : \vec{I} \to L$ in kind place graphs. We wish to build a kind RPO

$$(\vec{B} : \vec{I} \to J, B : J \to L).$$

We start by building a pure RPO $(\vec{B'}, B')$ for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$ via [7, Construction 7.7]. From this we shall construct a kind bound $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$, such that $(\vec{B}, B)^{\mathrm{u}} = (\vec{B'}, B')$. Then in the next proposition we shall show that it is a kind RPO.

Let $J' = n$ be the interface of this pure RPO. We now construct a kind interface $J = \langle n, \vec{C_J} \rangle$ with $J^{\mathrm{u}} = J'$ by providing the vector $\vec{C_J}$ as follows.



Letting r range over n, we define two sets C^0_r and C^1_r as:

$$C^0_r = \{ ctrl_{B_i}(v) | r = B_i(v), i \in \{0,1\} \},$$

$$C^1_r = \{ K | r = B_i(s) \wedge K \in C_{I_i,s}, i \in \{0,1\} \}.$$

Define $\vec{C_J} = (C_0 \cdots C_n)$ where C_i is defined as $C^0_i \cup C^1_i$. Clearly this is the least vector of subsets of $\mathcal{K}$ that satisfies KR1 and KR2 in B_i. We have now defined J and hence $(\vec{B}, B)$. J is defined to be a fitting interface for $\vec{B}$.

We now show that B_0, B_1 and B are kind place graphs. KR1 and KR2 hold in B_0 and B_1 by the construction of $\vec{C_J}$. We present the cases for KR3 and KR4 for B_0 — the proofs for B_1 follow similarly. In the following, $s \in m_i, r \in n$ and $t \in l$. We will need the fact that

$$ctrl_{D_i} = ctrl_B \uplus ctrl_{B_i}.$$

First, note that KR1-KR4 hold in $\vec{D}$. Also, $B' \circ B'_0 = D^{\mathrm{u}}_0$ and $prnt_{D_0} = prnt_{D^{\mathrm{u}}_0}$ so

$$prnt_{D_0} = (\mathrm{id}_{V_{B_0'}} \uplus prnt_{B'}) \circ (prnt_{B_0'} \uplus \mathrm{id}_{V_{B'}}) \qquad \text{(I)}$$

Let $v = {B_0}'(v')$. By (I), $v = D_0(v')$. Since KR3 holds in D_0, $ctrl_{D_0}(v') \in kind(ctrl_{D_0}(v))$ and KR3 holds in B_0.

Let $v = {B_0}'(s)$. By (I), $v = D_0(s)$. Since KR4 holds in D_0, $C_s \subseteq kind(ctrl_{D_0}(v))$ and KR4 holds in B_0.

We now show that KR1-KR4 holds in B. Note that a node v in B is also a node in both D_0 and D_1.

Let $t = B(v)$. By (I), $t = D_0(v)$. Since KR1 holds in D_0, $ctrl_{D_0}(v) \in C_t$ and KR1 holds in B.

Let $t = B(r)$. We must show $C_r \subseteq C_t$. We do a case-split:

- if $r = B_0(s)$ then $t = D_0(s)$. By the definition of C^1_r, if $K \in C_s$ then $K \in C_r$. Since KR2 holds in D_0, if $K \in C_s$ then $K \in C_t$ also.
- if $r = B_0(v)$ then $t = D_0(v)$. By the definition of C^0_r, $\mathit{ctrl}_{B_0}(v) \in C_r$. Since KR1 holds in D_0, $\mathit{ctrl}_{D_0}(v) \in C_t$ also.

The cases for B_1 are similar. Hence, if $K \in C_r$ then $K \in C_t$ and KR2 holds in B. Combining these cases covers all the sets in $\vec{C_J}$ given by the construction.

Let $v = B(v')$. By (I), $v = D_0(v')$. Since KR3 holds in D_0, $\mathit{ctrl}_{D_0}(v') \in \mathit{kind}(\mathit{ctrl}_{D_0}(v))$ and KR3 holds in B.

Let $v = B(r)$. We must show $C_r \subseteq \mathit{kind}(\mathit{ctrl}_B(v))$. We do a case-split:

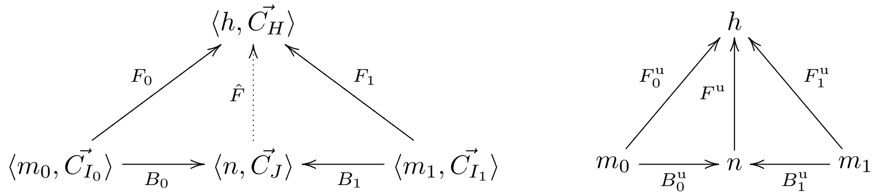
- if $r = B_0(s)$ then $v = D_0(s)$. By the definition of C^1_r, if $K \in C_s$ then $K \in C_r$. Since KR4 holds in D_0, if $K \in C_s$ then $K \in \mathit{kind}(\mathit{ctrl}_{D_0}(v))$.
- if $r = B_0(v')$ then $v = D_0(v')$. By the definition of C^0_r, $\mathit{ctrl}_{B_0}(v') \in C_r$. Since KR3 holds in D_0, $\mathit{ctrl}_{D_0}(v') \in \mathit{kind}(\mathit{ctrl}_{D_0}(v))$.

The cases for B_1 are similar. Hence, if $K \in C_r$ then $K \in \mathit{kind}(\mathit{ctrl}_B(v))$ and KR4 holds in B. Combining these cases covers all the sets in $\vec{C_J}$ given by the construction.

Hence, $(\vec{B}, B)$ is a kind bound for $\vec{A}$ to $\vec{D}$. □

Proposition 2.19 (kind RPOs [11.5]). *A kind RPO for $\vec{A}$ to $\vec{D}$ is provided by Construction 2.18.*

Proof. Let $(\vec{B}, B)$ be as in the construction. Let $(\vec{F}, F)$ be any other bound for $\vec{A}$ relative to $\vec{D}$. We must find a unique mediating arrow $\hat{F} : J \to H$.



From the construction, $(\vec{B}, B)^{\mathrm{u}}$ is a pure RPO for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$. $(\vec{F}, F)^{\mathrm{u}}$ is also a bound for $\vec{A}^{\mathrm{u}}$ relative to $\vec{D}^{\mathrm{u}}$ and so there is a unique mediating arrow $F' : J^{\mathrm{u}} \to H^{\mathrm{u}}$ between these two relative bounds.

We claim that $\hat{F} : J \to H$ where $\hat{F}^{\mathrm{u}} = F'$ obeys the kind rules. The proof follows similarly to the case for B in the Construction 2.18. We present it here for completeness. In the following, $s \in m_i, r \in n$ and $t \in h$. We will need the fact that

$$\mathit{ctrl}_{F_i} = \mathit{ctrl}_{B_i} \uplus \mathit{ctrl}_{\hat{F}}.$$

It is given that the kind rules hold in B_i and F_i. Also, $F' \circ B^{\mathrm{u}}_i = F^{\mathrm{u}}_i$ and $\mathit{prnt}_{F_i} = \mathit{prnt}_{F^{\mathrm{u}}_i}$ so

$$\mathit{prnt}_{F_i} = (\mathrm{id}_{V_{B_i}} \uplus \mathit{prnt}_{\hat{F}}) \circ (\mathit{prnt}_{B_i} \uplus \mathrm{id}_{V_{\hat{F}}}) \qquad \text{(I)}$$

Let $t = \hat{F}(v)$. By (I), $t = F_i(v)$. Since KR1 holds in F_i, KR1 holds in $\hat{F}$.

Let $t = \hat{F}(r)$. We must show $C_r \subseteq C_t$. We do a case-split:

- if $r = B_i(s)$ then $t = F_i(s)$. By the definition of C^1_r, if $K \in C_s$ then $K \in C_r$. Since KR2 holds in F_i, if $K \in C_s$ then $K \in C_t$ also.

- if $r = B_i(v)$ then $t = F_i(v)$. By the definition of C^0_r, $ctrl_{B_i}(v) \in C_r$. Since KR1 holds in F_i, $ctrl_{F_i}(v) \in C_t$ also.

Combining these cases covers all the pairs in $\vec{C_J}$ given by the construction. Hence, if $K \in C_r$ then $K \in C_t$ and KR2 holds in $\hat{F}$.

Let $v = \hat{F}(v')$. By (I), $v = F_i(v')$. Since KR3 holds in F_i, KR3 holds in $\hat{F}$. Let $v = \hat{F}(r)$. We must show $C_r \subseteq kind(ctrl_{\hat{F}}(v))$. We do a case-split:

- if $r = B_i(s)$ then $v = F_i(s)$. By the definition of C^1_r, if $K \in C_s$ then $K \in C_r$. Since KR4 holds in F_i, if $K \in C_s$ then $K \in kind(ctrl_{F_i}(v))$.

- if $r = B_i(v')$ then $v = F_i(v')$. By the definition of C^0_r, $ctrl_{B_i}(v') \in C_r$. Since KR3 holds in F_i, $ctrl_{F_i}(v') \in kind(ctrl_{F_i}(v))$.

Combining these cases covers all the pairs in $\vec{C_J}$ given by the construction. Hence, if $K \in C_r$ then $K \in kind(ctrl_{\hat{F}}(v))$ and KR4 holds in $\hat{F}$.

Hence, there exists a mediating arrow $\hat{F} : I \to J$ as shown, where $\mathcal{U}(\hat{F}) = F'$. Its uniqueness follows as F' is unique and $\mathcal{U}$ is faithful. □

Corollary 2.20 (preserving RPOs [11.6]). *The forgetful functor $\mathcal{U}$ preserves RPOs.*

Proof. The proof is identical to [7, Corollary 11.6]. □

Proposition 2.19 proves that the forgetful functor $\mathcal{U}$ creates RPOs. Thus, we have given an example of another functor which satisfies this property mentioned following Corollary 11.6 in [7]. The steps we took to prove RPO creation were 1) construct such a 'least' interface I based on the pure interface, 2) prove that the bigraphs $(\vec{B}, B)$ are of the generalised form (i.e. binding bigraphs, kind bigraphs), and 3) prove that $\hat{F}$ is of the generalised form. These steps are the same as those taken for binding bigraphs.

Jensen and Milner [7] ask the question as to whether a set of conditions on a functor exist which are sufficient to ensure RPO creation. Much work has been done recently in the literature [9, 14] to address this question. In our case, we note that that the similarity between the functors $\mathcal{U}' : \text{'BBG} \to \text{'BIG}$ and $\mathcal{U} : \text{'KPG} \to \text{'PLG}$ were that they were forgetful. The fact that composition is defined in 'KPG and 'BBG as for the underlying pure bigraphs seemed crucial in the proofs. A property which makes $\mathcal{U}$ forgetful is that the objects of 'KPG have extra structure which is forgotten. Faithfulness is easy to establish, as we have argued above, as A^{u} has the same *prnt* map as A.

Note also that J in Construction 2.18 was defined by defining $\vec{C_J}$ as the vector of *least* subsets of $\mathcal{K}$ such that KR1 and KR2 hold (*i.e.* J is fitting for $\vec{B}$.) This notion of 'least such' is the intuition behind the universal property of a pushout and thus may provide a guiding tactic for constructing relative bounds which are RPOs.

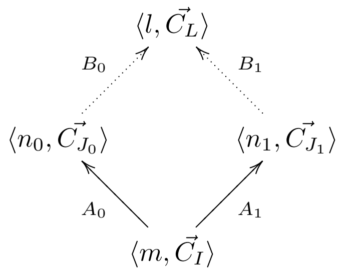
We now turn to the question of when a pair $\vec{A} : I \to \vec{J}$ is consistent. Note that in pure place graphs, the conditions CP0 – CP2 ensure that the *ctrl* maps agree and that the *prnt* maps of $\vec{A}$ are in some way compatible. Here we will also require that the vectors $\vec{C_{J_0}}$ and $\vec{C_{J_1}}$ are also in some way compatible. We will first justify the conditions we will use by examining the four cases where a place is the parent of another place. We will use the fact that bounds exist in pure place graphs when CP0 – CP2 are satisfied. A rigourous case analysis is given first but the reader may skip to the proceeding informal reasoning.

Example 2.21 (deriving consistency conditions — case analysis).

We now derive the consistency conditions necessary for a kind pair $\vec{A} : I \to \vec{J}$ to have a bound $\vec{B} : \vec{J} \to L$. We assume that CP0 – CP2 must hold for $\vec{A}$. In the following, $s \in m$ and $r_i \in n_i$ and $t \in l$.

Remember that we can always define $\vec{C_L}$ such that each component of the vector $C_{L,i} = \mathcal{K}$ so that given a pure bound, we can always define L such that KR1 and KR2 hold. In the arguments below, we examine all the parental cases of A_0. The arguments for A_1 are similar. Again, it may help to remember that

$$ctrl_{B_i} \uplus ctrl_{A_i} = ctrl_{B_i \circ A_i} = ctrl_{B_{\bar{\imath}} \circ A_{\bar{\imath}}}$$



Let $v = A_0(s)$**.** Then $v = (B_0 \circ A_0)(s) = (B_1 \circ A_1)(s)$ (I).

- If $v \in V_2$, then, by CP1, $v = A_1(s)$ and since KR4 holds here in $\vec{A}$, it will hold in $B_0 \circ A_0 = B_1 \circ A_1$. No change is necessary.
- If $v \notin V_2$, then, by CP2, $r_1 = A_1(s)$ and by (I), $v = B_1(r_1)$. Thus, we require that $C_{r_1} \subseteq \mathit{kind}(\mathit{ctrl}_{A_0}(v))$ to satisfy KR4 in $\vec{B}$. The remainder of CP2 does not add any new restrictions.

Our condition becomes:

If $A_i(s) \in V_i - V_2$ then $A_{\bar{\imath}}(s) \in n_{\bar{\imath}}$ and

$$C_{A_{\bar{\imath}}(s)} \subseteq \mathit{kind}(\mathit{ctrl}_{A_i}(A_i(s))),$$

and if also $A_{\bar{\imath}}(w) = A_{\bar{\imath}}(s)$ then $w \in m \uplus V_2$ and $A_i(w) = A_i(s)$.

Let $v = A_0(v')$**.** Then $v = (B_0 \circ A_0)(v') = (B_1 \circ A_1)(v')$ (II).

- If $v \in V_2$, then, by CP1, $v' \in V_2$ and $v = A_1(v')$ and since KR3 holds in $\vec{A}$, it will hold in $B_0 \circ A_0 = B_1 \circ A_1$. No change is necessary.
- If $v \notin V_2$, then, by CP2, $r_1 = A_1(v')$ and by (II), $v = B_1(r_1)$. As above, KR4 must be respected and so we have a similar requirement as before. The remainder of CP2 does not add any new restrictions.

Our condition is generalised ($w_2 \in m \uplus V_2$) to become:

If $A_i(w_2) \in V_i - V_2$ then $A_{\bar{\imath}}(w_2) \in n_{\bar{\imath}}$ and

$$C_{A_{\bar{\imath}}(w_2)} \subseteq \mathit{kind}(\mathit{ctrl}_{A_i}(A_i(w_2))),$$

and if also $A_{\bar{\imath}}(w) = A_{\bar{\imath}}(w_2)$ then $w \in m \uplus V_2$ and $A_i(w) = A_i(w_2)$.

Let $r_0 = A_0(s)$**.** We have two cases.

- If $r_1 = A_1(s)$ then we can have $B_0(r_0) = B_1(r_1) = t$, where t can contain at least the union of the set of controls that r_0 and r_1 can contain.
- If $v = A_1(s)$ then, by CP1, $v \notin V_2$ and this case has been covered.

Let $r_0 = A_0(v)$**.** We have three cases.

- $v \notin V_2$.
  If $r_0 = A_0(w_2)$ then $A_1(w_2) \in V_1 \cup n_1$ by CP1. If $A_1(w_2) \notin n_1$ then by CP2, siblings of w_2 are members of V_2. Hence $v \in V_2$. This leads to a contradiction, therefore either all such w_2 have roots as parents in A_1 or r_0 has no shared children.
  If all such w_2 have roots as parents then they must all map to the same parent in B_1. We choose this to be a root t where $B_0(r_0) = t$.
  Else, since all of the children of r_0 are in V_0, we let $t = B_0(r_0)$ and make $t \in l$ the parent of all these children in B_1, defining t to be able to contain all these controls.
- $v \in V_2, A_1(v) = v'$. By CP1, v' is not shared. This case has been covered.
- $v \in V_2, A_1(v) = r_1$.
  Since $(B_0 \circ A_0) = (B_1 \circ A_1)$, $B_0(r_0) = B_1(r_1)$. Hence, $B_0(r_0)$ can be a root defined to be able to contain r_0 and r_1.

Example 2.21b (deriving consistency conditions — in short).

Given a pure bound $\vec{B}$ for $\vec{A}$ and since $\vec{C_I}$ can be equal to $(\mathcal{K} \cdots \mathcal{K})$, we need only to ensure that KR3 and KR4 hold in $\vec{B}$.

If $v = prnt(v')$ in B_i then neither node is in V_2 or V_i. If both are in $V_{\bar{\imath}} - V_2$ then KR3 is satisfied since it holds in $A_{\bar{\imath}}$. Otherwise, v' at least is a fresh node (*i.e.* $v' \notin V_0 \cup V_1 \cup V_2$) and its parent in B_0 and B_1 must be either a fresh node or a root. We can remove any such branches of these fresh nodes in $\vec{B}$ (mapping their non-fresh children to the root above the branch) and ensure a different bound where KR3 holds.

If $v = prnt(r_i)$ in B_i then it is either a member of $V_{\bar{\imath}} - V_2$ or a fresh node. The latter can be turned into a bound as above. The former case is covered by our extra condition. We need only modify rule CP2 as follows:

Definition 2.22 (consistency conditions for kind place graphs [7.9]). *Let $\vec{A}$ be a pair of kind place graphs with common inner face. We define two conditions for $\vec{A}$ to be consistent.*

CP *Conditions* CP0 *and* CP1 *for the underlying place graphs [7, Definition 7.9].*

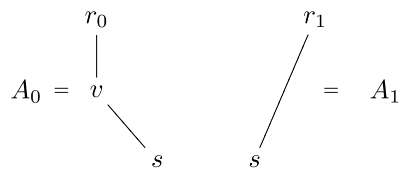
KC *If* $A_i(w_2) \in V_i - V_2$

$$\textit{then } A_{\bar{\imath}}(w_2) \in n_{\bar{\imath}} \textit{ and } C_{A_{\bar{\imath}}(w_2)} \subseteq kind(ctrl_{A_i}(A_i(w_2))),$$

$$\textit{and if also } A_{\bar{\imath}}(w) = A_{\bar{\imath}}(w_2) \textit{ then } w \in m \uplus V_2 \textit{ and } A_i(w) = A_i(w_2).$$

Note that if CP and KC hold for a pair $\vec{A}$, then since CP and KC imply CP0 − CP2, $\vec{A}^{\mathrm{u}}$ is also consistent. Thus, bounds are preserved by $\mathcal{U}$.

Example 2.23 (Bounds are not reflected by $\mathcal{U}$). *Let $A_0, A_1 : \langle 1, (\{L\})\rangle \to \langle 1, (\{L, K\})\rangle$ be a pair of place graphs with the following prnt maps:*



Let $ctrl_{A_0}(v) = K$ and $kind(v) = L$. Then KC *does not hold as $C_{r_1} = \{K, L\} \supset \{L\} = kind(v)$ and so $\vec{A}$ has no bound. However, $\vec{A}^{\mathrm{u}}$ has many bounds.*

Theorem 2.24 (kind IPOs [11.8]).

1. *The consistency conditions* CP *and* KC *are necessary for the existence of bounds in kind place graphs.*

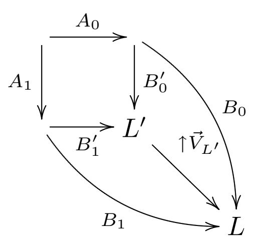
2. *If $\vec{A}$ has a kind IPO $\vec{B}$ with outer interface L then L is fitting for $\vec{B}$.*

3. *Let $\vec{A}$ satisfy the consistency conditions and $\vec{A}^{\mathrm{u}}$ have a pure IPO $\vec{B}'$. Then $\vec{A}$ has a kind IPO $\vec{B}$, with $\vec{B}^{\mathrm{u}} = \vec{B}'$.*

4. *If $\vec{A}$ has a kind IPO $\vec{B}$, then $\vec{A}^{\mathrm{u}}$ has a pure IPO $\vec{B}^{\mathrm{u}}$.*

Proof. 1. If $\vec{B}$ is a bound for $\vec{A}$ in 'Kpg then $\vec{B}^u$ is a bound for $\vec{A}^u$ in 'Plg so CP is necessary. As in the previous argument, KC is necessary in order for KR4 to hold in $\vec{B}$.

2. Let $\vec{A}$ has a kind IPO $\vec{B}$ with outer interface L. Assume L is not fitting for $\vec{B}$. Then we can construct a pair $\vec{B}'$ with outer interface L' where B_i and B'_i have the same *prnt* map and L' is fitting for $\vec{B}'$. We also define an inflation $\uparrow\vec{V}_{L'}$ such that the diagram below commutes.



$(\vec{B}', \uparrow\vec{V}_{L'})$ is a relative bound for $\vec{A}$. $(\vec{B}, \mathrm{id})$ is an RPO for $\vec{A}$ to $\vec{B}$. Thus, there must exist a unique arrow $F : L \to L'$ from $(\vec{B}, \mathrm{id})$ to $(\vec{B}', \uparrow\vec{V}_{L'})$. However, this arrow can not exist as since L is not fitting for $\vec{B}$ and L' is fitting for $\vec{B}'$, the vector $\vec{V}$ used in the inflation has at least one pair $s \in L', r \in L$ such that $C_s \subset C_r$. Thus, any such F would break KR2 as it would map r to s and $C_r \supset C_s$. We have reached a contradiction and so L must be fitting.

3. KC $\implies$ CP2 and so $\vec{A}^u$ is ensured to have at least one IPO $\vec{B}'$[3]. We construct a bound $\vec{B}$ such that $\vec{B}^{\mathrm{u}} = \vec{B}'$. As in Construction 2.18, we need only define $\vec{C_L}$ in the outer face $\langle l, \vec{C_L} \rangle$ of $\vec{B}$ such that $\vec{C_L}$ is fitting for $\vec{B}$ (from the argument above, KR3 holds in $\vec{B}$ and KR4 is ensured by KC.) Then $(\vec{B}^{\mathrm{u}}, \mathrm{id})$ is an RPO for $\vec{A}^{\mathrm{u}}$ to $\vec{B}^{\mathrm{u}}$ (as $\vec{B}' = \vec{B}^{\mathrm{u}}$ is an IPO for $\vec{A}^{\mathrm{u}}$). Using the argument of Proposition 2.19, $(\vec{B}, \mathrm{id})$ is an RPO for $\vec{A}$ to $\vec{B}$ and hence a kind IPO for $\vec{A}$.

4. This is a special case of Corollary 2.20.

□

Thus, by Theorems 3 and 4 above, when a pair $\vec{A}$ of kind place graphs is consistent, there is a precise correspondence between its kind IPOs and the pure IPOs of $\vec{A}^{\mathrm{u}}$. Theorem 2.24.2 now becomes important as when $\vec{A}$ is a consistent pair of kind IPOs and we are given a pure IPO $\vec{B}'$ for $\vec{A}^{\mathrm{u}}$, then the IPO for $\vec{A}$ is given by taking the *prnt* maps of the pure IPO $\vec{B}'$ and defining the outer interface of $\vec{B}'$ to be fitting for $\vec{B}'$.

2.3 Hard kind place graphs

We finish this section by discussing hard place graphs in ´KPG.

Definition 2.25 (hard kind place graphs [7.13]). *A* hard *kind place graph is one in which no root or non-atomic node is barren. They form a sub-s-category denoted by* ´KPG$_{\mathrm{h}}$.

Hard kind place graphs are epi by definition. As in the pure theory, if a composition $B \circ A$ in ´KPG is hard then so are both A and B. $\mathcal{U}$ maps hard kind place graphs to hard place graphs.

Proposition 2.26 (pushouts for hard place graphs [7.14]). *If $\vec{A}$ is a consistent pair of hard kind place graphs, then it has a pushout $\vec{B}$ which is hard and is a pushout in* ´KPG$_{\mathrm{h}}$.

Proof. $\vec{A}$ is a consistent pair of hard kind place graphs. Thus, $\vec{A}^{\mathrm{u}}$ is a consistent pair of hard place graphs. By [7, Proposition 7.14], $\vec{A}^{\mathrm{u}}$ has a unique IPO (a pushout) $\vec{B}'$, which is a pushout in ´PLG$_{\mathrm{h}}$. By Theorem 2.24.3, we have an IPO $\vec{B}$ in ´KPG, where $\vec{B}^{\mathrm{u}} = \vec{B}'$. By Theorem 2.24, this IPO is unique and so we have a pushout $\vec{B}$ in ´KPG. Further, $\vec{B}$ is a pushout in ´KPG$_{\mathrm{h}}$ since, as $\vec{B}^{\mathrm{u}}$ is hard, $\vec{B}$ is hard. □

We now find another connection between ´KPG and ´KPG$_{\mathrm{h}}$, generalising the similar connection in pure place graphs. Let $\mathcal{K}$ be a kind signature. We define a new atomic control $\triangle$ with zero arity; we adjoin $\triangle$ to $\mathcal{K}$ to form $\mathcal{K}^{\triangle} = \mathcal{K} \uplus \triangle$ and define $kind_{\mathcal{K}^{\triangle}}$ by

$$kind_{\mathcal{K}^{\triangle}}(K) = \begin{cases} \emptyset & \text{if } kind_{\mathcal{K}}(K) = \emptyset \\ kind_{\mathcal{K}}(K) \cup \triangle & \text{else} \end{cases}$$

[3]Note that the nodes of an IPO $\vec{B}$ are a subset of the nodes of $\vec{A}$ *i.e.* no fresh nodes are introduced.

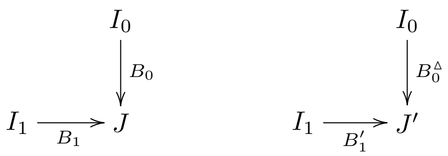
so that all non-atomic controls of $\mathcal{K}$ can contain $\triangle$. We can make any arrow G of $\acute{\mathrm{K}}\mathrm{PG}(\mathcal{K})$ into a hard kind place graph in $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}(\mathcal{K}^{\triangle})$ by adding a $\triangle$-node as a child of any barren root or non-atomic node, adding $\triangle$ to the set of controls that the root can contain.

Place equivalence, $\equiv_{\triangle}$, is now defined as in the pure theory and is a static congruence[4]. There is also a quotient s-category $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}(\mathcal{K}^{\triangle})/\equiv_{\triangle}$. Unlike the pure theory, $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}(\mathcal{K}^{\triangle})/\equiv_{\triangle}$ is not isomorphic with $\acute{\mathrm{K}}\mathrm{PG}(\mathcal{K})$. This can be seen by noting that the objects of $\acute{\mathrm{K}}\mathrm{PG}(\mathcal{K})$ are a strict subset of the objects of $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}(\mathcal{K}^{\triangle})/\equiv_{\triangle}$, as the latter s-category has all the interfaces of the former as well as interfaces I where $C_r \in \vec{C_I}$ may contain $\triangle$.

In the pure theory, the operation of adding a new place node $\triangle$ to a place graph was easily defined. Since we have to ensure that our place graphs obey the kind rules, we must define the operation of adding a new place node $\triangle$ to a kind place graph more carefully. Given a kind place graph $G : I \to \langle n, \vec{C_J}\rangle$, if we add a $\triangle$-node as a child of any non-atomic node, then we have a new place graph $G' : I \to \langle n, \vec{C_J}\rangle$ in the same homset as G. If we add a $\triangle$-node as a child of any root $r \in n$ in G, then we have a new place graph $G' : I \to \langle n, \vec{C_{J'}}\rangle$ which may have a different outer interface to G. We define

$$\vec{C_{J'}} \stackrel{\text{def}}{=} (C_{J,0} \ldots C_{J,r} \cup \triangle \ldots C_{J,n-1}).$$

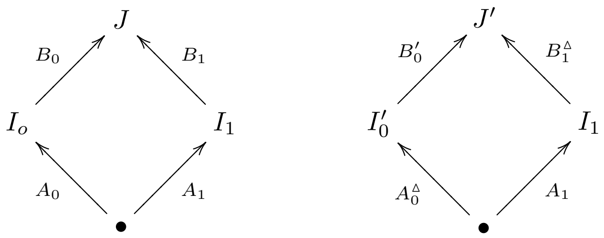
With this definition, given a fitting interface J for a pair $\vec{B} : \vec{I} \to J$, if we add a place node to B_0 to yield the pair $(B_0^{\triangle}, B_1')$ as in the figure below, with outer interface J', and where $prnt_{B_1'} = prnt_{B_1}$, then J' is a fitting interface for $B_0^{\triangle}, B_1'$. J' is fitting as either $J' = J$ or else J has been altered as above such that KR1 is satisfied in $B_0^{\triangle}$.



In the following final propositions of this section, we shall use $\triangle$ to mean a fresh place node $\triangle_u$, of control $\triangle$, distinct from all others present. The final two propositions in this section are not exactly the same as in the pure theory. The extra complications lie in the fact that a bound must have a fitting interface in order to be an IPO for some pair. Note, however, that if all the place graphs in the propositions are fitting then the extra complication does not arise and we retrieve the original propositions exactly.

Proposition 2.27 (first pushout variation [7.15]). *Let $\vec{B}$ be a bound for $\vec{A}$ in $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}(\mathcal{K}^{\triangle})$. Add a new place node $\triangle$ to both A_0 and B_1, yielding $A_0^{\triangle}$ and $B_1^{\triangle}$ such that (i) if $\triangle$ is added as a child of root r in B_1 and $\triangle \in C_{J,r}$, then $\triangle \in C_{J,r}$ is necessary for KR1 or KR2 to hold in $\vec{B}$ and (ii) $B_0' \circ A_0^{\triangle} = B_1^{\triangle} \circ A_1$. Then $\vec{B}$ is a pushout for $\vec{A}$ iff $(B_0', B_1^{\triangle})$ is a pushout for $(A_0^{\triangle}, A_1)$.*

[4] That is, if $f \equiv_{\triangle} f'$ and $g \equiv_{\triangle} g'$ then (when both sides of the equations are defined), $f \otimes g \equiv_{\triangle} f' \otimes g'$ and $f \circ g \equiv_{\triangle} f' \circ g'$.



Proof. $\Leftarrow$ $(B'_0, B_1^\vartriangle)$ is a pushout for $(A_0^\vartriangle, A_1)$. By Theorem 2.24, $({B'_0}^{\mathrm{u}}, {B_1^\vartriangle}^{\mathrm{u}})$ is a pushout for $({A_0^\vartriangle}^{\mathrm{u}}, {A_1}^{\mathrm{u}})$. By [7, Proposition 7.15], $\vec{B}^{\mathrm{u}}$ is a pushout for $\vec{A}^{\mathrm{u}}$. $\vec{A}$ is a consistent pair of kind place graphs. By Theorem 2.24.2, J' is fitting for $(B_0, B_1^\vartriangle)$. Given condition (i), J is fitting for (B_0, B_1). By Theorem 2.24, $\vec{B}$ is a pushout for $\vec{A}$.

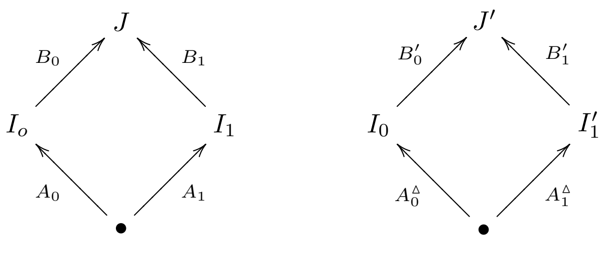
$\Rightarrow$ The proof is similar except that condition (i) is not necessary as if $\vec{B}$ is an IPO for $\vec{A}$, then J is fitting for $\vec{B}$ and so J' is fitting for $(B'_0, B_1^\vartriangle)$.

□

Again, the following proposition needs to be altered due to the kind rules. This results in the same kind of asymmetry with regards the fitness of J and J' to $\vec{B}$ and $\vec{B'}$ respectively.

Proposition 2.28 (second pushout variation [7.16]). *Let $\vec{B}$ be a bound for $\vec{A}$ in $\acute{\mathrm{K}}\mathrm{PG_h}(\mathcal{K}^\vartriangle)$. Let a fresh place node $\vartriangle$ be added to both members of $\vec{A}$, yielding $\vec{A}^\vartriangle$ and $\vec{B'}$ (where $prnt_{B'_i} = prnt_{B_i}$ and J' is equal to J except for any change needed so that KR2 holds in B'_1) such that $\vec{B'}$ is also a bound for $\vec{A}^\vartriangle$, and with $A_0^\vartriangle(\vartriangle)$ a node (not a root). Also, (i) if $\vartriangle$ is added as a child of root s in A_1, $B_1(s) = r$ and $\vartriangle \in C_{J,r}$, then $\vartriangle \in C_{J,r}$ is necessary for KR2 to hold in B_1. Then*

1. *If $\vec{B}$ is a pushout for $\vec{A}$, $\vec{B'}$ is a pushout for $\vec{A}^\vartriangle$.*
2. *Let $\vartriangle$ have a sibling w in both $A_0^\vartriangle$ and $A_1^\vartriangle$. Then if $\vec{B'}$ is a pushout for $\vec{A}^\vartriangle$, $\vec{B}$ is a pushout for $\vec{A}$.*



Proof. 1. $\vec{B}$ is a pushout for $\vec{A}$. By Theorem 2.24, $\vec{B}^{\mathrm{u}}$ is a pushout for $\vec{A}^{\mathrm{u}}$. By Proposition [7, Proposition 7.16], $\vec{B}^{\mathrm{u}} = \vec{B'}^{\mathrm{u}}$ is a pushout for $(A_0^\vartriangle, A_1^\vartriangle)^{\mathrm{u}}$. $\vec{A}^\vartriangle$ is a consistent pair of kind place graphs. Since $\vec{B}$ is a pushout for $\vec{A}$, J is fitting for $\vec{B}$. By construction, J' is fitting for $\vec{B'}$. Thus, by Theorem 2.24, $\vec{B'}$ is a pushout for $\vec{A}^\vartriangle$.

2. $\vec{B'}$ is a pushout for $\vec{A}^{\triangle}$. By Theorem 2.24, $\vec{B'}^{\mathrm{u}}$ is a pushout for $\vec{A}^{\triangle\mathrm{u}}$. By Proposition [7, Proposition 7.16], $\vec{B'}^{\mathrm{u}} = \vec{B}^{\mathrm{u}}$ is a pushout for $\vec{A}^{\mathrm{u}}$. $\vec{A}$ is a consistent pair of kind place graphs. Since $\vec{B'}$ is a pushout for $\vec{A}^{\triangle}$, J' is fitting for $\vec{B'}$. By condition (i), J is fitting for $\vec{B}$. Thus, by Theorem 2.24, $\vec{B}$ is a pushout for $\vec{A}$.

□

Finally, given that hard kind place graphs have no barren roots or non-atomic nodes, we may also consider altering a non-hard kind place graph to be hard by assigning a site s, where $C_s = \emptyset$, as a child to each barren root or non-atomic node. Such a site is essentially useless as no node can ever be planted inside it. This direction has not been pursued to date as adding such a useless site alters the homset of the place graph.

3 Kind bigraphs

We now combine kind place graphs and pure link graphs to define kind bigraphs. We will omit the adjective 'pure' when referring to link graphs. Interfaces for kind bigraphs are of the form $I = \langle m, \vec{C}_I, X \rangle$ where the first two components are from kind place graphs and the third is from link graphs. We call $I^{\mathrm{u}} = \langle m, X \rangle$ the pure interface *underlying* I.

Definition 3.1 (kind bigraph). *A (concrete) kind bigraph $G : I \to J$ consists of an* underlying *pure bigraph $G^{\mathrm{u}} : I^{\mathrm{u}} \to J^{\mathrm{u}}$ which respects the kind rules of the last section. We also describe a kind bigraph G as a pair $\langle G^{\mathrm{K}}, G^{\mathrm{L}} \rangle$, where G^{K} is the kind place graph and G^{L} is the link graph. We then have $G^{\mathrm{u}} = \langle (G^{\mathrm{K}})^{\mathrm{u}}, G^{\mathrm{L}} \rangle$.*

Definition 3.2 (precategory of kind concrete bigraphs [9.1]). *The precategory $\acute{\mathrm{K}}\mathrm{BG}(\mathcal{K})$ of kind concrete bigraphs over $\mathcal{K}$ has kind interfaces $I = \langle m, \vec{C}_I, X \rangle$ 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, their 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 $\langle \mathrm{id}_{\langle m, \vec{C}_I \rangle}, \mathrm{id}_X \rangle : I \to I$, where $I = \langle m, \vec{C}_I, X \rangle$. The subprecategory $\acute{\mathrm{K}}\mathrm{BG}_{\mathrm{h}}$ consists of hard *kind bigraphs — those with kind place graphs in $\acute{\mathrm{K}}\mathrm{PG}_{\mathrm{h}}$.*

We extend the forgetful functor $\mathcal{U}$ of the last section. $\mathcal{U} : \acute{\mathrm{K}}\mathrm{BG}(\mathcal{K}) \to \acute{\mathrm{B}}\mathrm{IG}(\mathcal{K})$ sends each I to I^{u} and each G to G^{u}.

We proceed with the definitions of elementary notions for kind bigraphs.

3.1 Elementary notions

isomorphisms [9.2] The isomorphisms in $\acute{\mathrm{K}}\mathrm{BG}$ are all combinations $\iota = \langle \iota^{\mathrm{K}}, \iota^{\mathrm{L}} \rangle$ of a kind place graph isomorphism and a link graph isomorphism.

tensor product [9.3] The tensor product of interfaces $I = \langle m, \vec{C}_I, X \rangle$ and $J = \langle n, \vec{C}_J, Y \rangle$, where X and Y are disjoint, is

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

The tensor product $G : I \to J$ of two kind bigraphs $G_i : I_i \to J_i$ $(i = 0, 1)$ with disjoint node and edge sets is defined when $I = I_0 \otimes I_1$ and $J = J_0 \otimes J_1$ are defined, and then $G^{\mathrm{u}} = G_0^{\mathrm{u}} \otimes G_1^{\mathrm{u}}$. Thus $\mathcal{U}$ preserves tensor product.

Theorem 3.3 (kind bigraphs are wide monoidal [9.4]). *For any kind signature $\mathcal{K}$, the precategories $\acute{\mathrm{K}}\mathrm{BG}(\mathcal{K})$ and $\acute{\mathrm{K}}\mathrm{BG}_{\mathrm{h}}(\mathcal{K})$ are wide monoidal; the origin is $\epsilon = \langle 0, (), \emptyset \rangle$, and the interface $\langle m, \vec{C}_I, X \rangle$ has width m.*

Proof. $\acute{\mathrm{K}}\mathrm{BG}(\mathcal{K})$ and $\acute{\mathrm{K}}\mathrm{BG}_{\mathrm{h}}(\mathcal{K})$ are wide precategories, being built from $\acute{\mathrm{B}}\mathrm{IG}(\mathcal{K})$ and $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\mathcal{K})$ respectively with no change to the notions of support or width. For each permutation π of the ordinal width(I), there is an isomorphism $\pi_I : I \to I$ as in Proposition 2.6 with width$(\pi_I) = \pi$.

In $\mathcal{K}\text{BG}(\mathcal{K})$ and $\mathcal{K}\text{BG}_\text{h}(\mathcal{K})$, width preserves tensor product as we define the tensor product of kind place graphs in terms of pure place graphs. We now show that $\mathcal{K}\text{BG}(\mathcal{K})$ and $\mathcal{K}\text{BG}_\text{h}(\mathcal{K})$ are monoidal with unit ϵ to complete the proof.

Let $I = \langle m, \vec{C}_I, X\rangle$. By definition, $I \otimes \epsilon = \langle m+0, \vec{C}_I(), X \uplus \emptyset\rangle = \langle m, \vec{C}_I, X\rangle = I$. Similarly, $\epsilon \otimes I = I$.

Let $J = \langle n, \vec{C}_J, Y\rangle$. We define

$$\gamma_{I,J} : \langle m+n, \vec{C}_I\vec{C}_J, X \uplus Y\rangle \to \langle n+m, \vec{C}_J\vec{C}_I, Y \uplus X\rangle$$

as the isomorphism (no nodes or edges) with an identity *link* map and where

$$\begin{aligned} s &\mapsto \text{width}(J) + s, \quad s \in m \\ s &\mapsto s - \text{width}(I), \quad s \in n \end{aligned}$$

in the *prnt* map. This is indeed an isomorphism as the parent map is a bijection and the kind rules hold. We now prove the five required equations:

(1) and (2) The proofs follow from the facts that the two kind bigraphs have the same interface by definition of tensor product, that $\mathcal{U}$ preserves tensor product and is faithful and that $\mathcal{B}\text{IG}(\mathcal{K})$ and $\mathcal{B}\text{IG}_\text{h}(\mathcal{K})$ are monoidal. The following diagrams sketch the proof.

$$\begin{array}{ccc} f \otimes (g \otimes h) & & (f \otimes g) \otimes h \\ \downarrow \mathcal{U} & & \downarrow \mathcal{U} \\ f^\text{u} \otimes (g^\text{u} \otimes h^\text{u}) & \overset{\text{associativity}}{=\!=\!=} & (f^\text{u} \otimes g^\text{u}) \otimes h^\text{u} \end{array}$$

$$\begin{array}{ccc} (f_1 \otimes g_1) \circ (f_0 \otimes g_0) & & (f_1 \circ f_0) \otimes (g_1 \circ g_0) \\ \downarrow \mathcal{U} & & \downarrow \mathcal{U} \\ ({f_1}^\text{u} \otimes {g_1}^\text{u}) \circ ({f_0}^\text{u} \otimes {g_0}^\text{u}) & \overset{\text{monoidal}}{=\!=\!=} & ({f_1}^\text{u} \circ {f_0}^\text{u}) \otimes ({g_1}^\text{u} \circ {g_0}^\text{u}) \end{array}$$

(3) $\gamma_{I,\epsilon} = \text{id}_I$

$\gamma_{I,\epsilon}$ has an identity *link* map. By definition, for $s \in m$, $s \mapsto \text{width}(\epsilon) + s = 0 + s = s$ and $\vec{C}_I() = ()\vec{C}_I = \vec{C}_I$.

(4) $\gamma_{J,I} \circ \gamma_{I,J} = \text{id}_{I \otimes J}$

By definition, the composition has an identity *link* map. We show that the *prnt* map is an identity.

$$\begin{array}{ll} s \in m, & s \overset{\gamma_{I,J}}{\longmapsto} \text{width}(J) + s \overset{\gamma_{J,I}}{\longmapsto} (\text{width}(J) + s) - \text{width}(J) = s \\ s \in n, & s \overset{\gamma_{I,J}}{\longmapsto} s - \text{width}(I) \overset{\gamma_{J,I}}{\longmapsto} \text{width}(I) + (s - \text{width}(I)) = s \end{array}$$

(5) The proof is similar to (1) and (2). First, note that by the definitions of $\gamma_{I,K}, \gamma_{H,J}$ and tensor product, both bigraphs are in the same homset.

Next, note that $\mathcal{U}(\gamma_{I,J})$ is precisely the symmetry defined for pure bigraphs in [7, Section 10]. The proof is then:

$$\begin{array}{rll} & \mathcal{U}(\gamma_{I,K} \circ (f \otimes g)) & \\ = & \mathcal{U}(\gamma_{I,K}) \circ \mathcal{U}(f \otimes g) & \text{(functor laws)} \\ = & \mathcal{U}(\gamma_{I,K}) \circ (\mathcal{U}(f) \otimes \mathcal{U}(g)) & (\mathcal{U} \text{ preserves } \otimes) \\ = & (\mathcal{U}(g) \otimes \mathcal{U}(f)) \circ \mathcal{U}(\gamma_{H,J}) & (\text{'Big is monoidal}) \\ = & \mathcal{U}(g \otimes f) \circ \mathcal{U}(\gamma_{H,J}) & (\mathcal{U} \text{ preserves } \otimes) \\ = & \mathcal{U}((g \otimes f) \circ \gamma_{H,J}) & \text{(functor laws)} \end{array}$$

The proof follows since $\mathcal{U}$ is faithful.

□

Thus, 'Kbg($\mathcal{K}$) is an s-category, and 'Kbg$_{\mathrm{h}}$($\mathcal{K}$) is a sub-s-category of 'Kbg($\mathcal{K}$).

epis and monos [9.5] A kind bigraph is epi (resp. mono) iff its underlying pure bigraph is epi (resp. mono) *i.e.* $\mathcal{U}$ both preserves and reflects epis and monos.

fitting, fitting interface We say a bigraph G is fitting if its place graph component G^{K} is. Given a set of bigraphs $\{A_0, \ldots, A_{n-1}\}$ with a common codomain $I = \langle m, \vec{C}_I, X\rangle$, we say I is a *fitting interface for* $A_0, \ldots, A_{n-1}$ iff $\langle m, \vec{C}_I\rangle$ is a fitting interface for $A_0^{\mathrm{K}}, \ldots, A_{n-1}^{\mathrm{K}}$. If $\vec{B}$ is a bound for $\vec{A}$ and the outer interface of $\vec{B}$ is a fitting interface for $\vec{B}$, then we say that $\vec{B}$ is a *fitting bound* for $\vec{A}$.

The composition of two fitting bigraphs is a fitting bigraph and the tensor product of two fitting bigraphs is a fitting bigraph. Fitting bigraphs form a sub-s-category 'Fkb($\mathcal{K}$) of 'Kbg($\mathcal{K}$).

We extend Lemmas 2.14.1-2.14.4 to kind bigraphs. The proofs are trivial.

Lemma 3.4 (some properties of fitting bigraphs).

1. *Given a kind bigraph $D : H \to \langle n, C_J, Y\rangle$ and a* fitting *bigraph $A : H \to \langle m, C_I, X\rangle$, let $B' : \langle m, X\rangle \to \langle n, Y\rangle$ be a pure bigraph such that $D^{\mathrm{u}} = B' \circ A^{\mathrm{u}}$. Then there exists a unique kind bigraph $B : \langle m, C_I, X\rangle \to \langle n, C_J, Y\rangle$ such that $prnt_B = prnt_{B'}$ (*i.e. *$B^{\mathrm{u}} = B'$) and $D = B \circ A$.*

2. *Let $D = B \circ A$ and D and A be fitting bigraphs. Then B is a fitting bigraph.*

3. *Let $A : H \to \langle m, C_I, X\rangle$ and $D : H \to \langle n, C_J, Y\rangle$ be two* fitting *bigraphs with common domain and let $B' : \langle m, X\rangle \to \langle n, Y\rangle$ be a pure bigraph such that $D^{\mathrm{u}} = B' \circ A^{\mathrm{u}}$. Then there exists a unique fitting bigraph $B : \langle m, C_I, X\rangle \to \langle n, C_J, Y\rangle$ such that $B^{\mathrm{u}} = B'$ and $D = B \circ A$.*

4. *Given a consistent pair of fitting bigraphs $\vec{A} : H \to \vec{I}$ with common domain, let $\vec{B} : \vec{I} \to J$ be a bound for $\vec{A}$ such that $\vec{B}$ is a fitting bound for $\vec{A}$. Then both B_0 and B_1 are fitting bigraphs.*

inflations, deflations We extend these notions to bigraphs. A bigraph is an inflation (resp. deflation) if its place graph is an inflation (resp. deflation) and its link graph is an identity. We write $\uparrow \vec{V}_I$ to mean $\uparrow \vec{V}_I \otimes \mathrm{id}_X$ where $I = \langle m, \vec{C}_I, X \rangle$, and similarly for deflations. We use these definitions to define the outer inflation and inner deflations of a bigraph G as $\uparrow\vec{V}_J \circ G$ and $G \circ \downarrow\vec{V}_I$ respectively.

Given a bigraph $G : \langle m, \vec{C}_I, X \rangle \to \langle n, \vec{C}_J, Y \rangle$, inner inflations and outer deflations of G — $G \nearrow \vec{V} : \langle m, \vec{C}_I \dot{\uplus} \vec{V}, X \rangle \to \langle n, \vec{C}_J, Y \rangle$ and $G \swarrow \vec{V}' : \langle m, \vec{C}_I, X \rangle \to \langle n, \vec{C}_J \mathbin{\dot{\setminus}} \vec{V}', Y \rangle$ respectively — have link maps $link_G$, parent maps $prnt_G$ and are defined when $G^{\mathrm{K}} \nearrow \vec{V}$ and $G^{\mathrm{K}} \swarrow \vec{V}'$ are defined.

Inflations may be used to decompose a bigraph G into a composition $\uparrow\vec{V}_J \circ G'$ where G' is fitting. We conjecture that a likely use for inflations and deflations would be as elementary bigraphs in an axiomatisation of kind bigraphs. As with place graphs, if a bigraph G' is obtained by inflating or deflating G then $G'^{\mathrm{u}} = G^{\mathrm{u}}$.

The following is a corollary of [7, Theorem 8.9] and Proposition 2.19, noting that RPOs for kind place graphs (Construction 2.18) are built from pure place graph RPOs.

Corollary 3.5 (RPOs for kind bigraphs [9.6]). *In the s-categories* ´KBG *and* ´KBG$_{\mathrm{h}}$*, an RPO for $\vec{A}$ to $\vec{D}$ is provided by the triple*

$$(\langle B_0^{\mathrm{K}}, B_0^{\mathrm{L}} \rangle, \langle B_1^{\mathrm{K}}, B_1^{\mathrm{L}} \rangle, \langle B^{\mathrm{K}}, B^{\mathrm{L}} \rangle)$$

where $(\vec{B}^{\mathrm{K}}, B^{\mathrm{K}})$ is a kind place graph RPO for $\vec{A}^{\mathrm{K}}$ to $\vec{D}^{\mathrm{K}}$ and $(\vec{B}^{\mathrm{L}}, B^{\mathrm{L}})$ is a link graph RPO for $\vec{A}^{\mathrm{L}}$ to $\vec{D}^{\mathrm{L}}$.

Proof. By Corollary 2.20, $(\vec{B}^{\mathrm{K}}, B^{\mathrm{K}})^{\mathrm{u}}$ is a place graph RPO for $(\vec{A}^{\mathrm{K}})^{\mathrm{u}}$ to $(\vec{D}^{\mathrm{K}})^{\mathrm{u}}$. By [7, Theorem 7.8], this pure place graph RPO was constructed by [7, Construction 7.7]. The link graph RPO is constructed by [7, Construction 8.8]. The proof follows from [7, Corollary 9.6]. □

We can see that given an pure RPO in ´BIG, a kind RPO can be constructed in ´KBG via Construction 2.18. Corollary 2.20 then carries through to ´KBG, with the proof being identical to that of [7, Corollary 11.6].

Corollary 3.6 (preserving RPOs [11.6]). *The forgetful functor $\mathcal{U}$ preserves RPOs.*

We now bring the IPO theory of kind place graph IPOs to kind bigraphs. The consistency conditions for kind place graphs imply CP and CL.

Definition 3.7 (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 *Conditions* CP0 *and* CP1 *for the underlying place graphs [7, Definition 7.9].*

CL *Conditions* CL0 – CL2 *for link graphs [7, Definition 8.10].*

KC *If* $A_i(w_2) \in V_i - V_2$

$$\textit{then } A_{\bar{\imath}}(w_2) \in n_{\bar{\imath}} \textit{ and } C_{A_{\bar{\imath}}(w_2)} \subseteq \textit{kind}(\textit{ctrl}_{A_i}(A_i(w_2))),$$

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 3.8 (kind IPOs [11.8]).

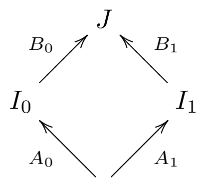
1. *The consistency conditions* CP, CL *and* KC *are necessary for the existence of bounds in kind bigraphs.*
2. *If* $\vec{A}$ *has a kind IPO* $\vec{B}$ *with outer interface* L *then* L *is fitting for* $\vec{B}$.
3. *Let* $\vec{A}$ *satisfy the consistency conditions and* $\vec{A}^{\mathrm{u}}$ *have a pure IPO* $\vec{B}'$. *Then* $\vec{A}$ *has a kind IPO* $\vec{B}$, *with* $\vec{B}^{\mathrm{u}} = \vec{B}'$.
4. *If* $\vec{A}$ *has a kind IPO* $\vec{B}$, *then* $\vec{A}^{\mathrm{u}}$ *has a pure IPO* $\vec{B}^{\mathrm{u}}$.

Proof. Similar to Theorem 2.24. □

Similarly to Theorem 3.8.2, it can also be shown that if $\vec{A}$ has a kind pushout $\vec{B}$ with outer interface L then L is fitting for $\vec{B}$.

The next proposition is particular to kind bigraphs and will have a use in their dynamic theory. The proposition itself looks similar to many such propositions in category theory if we replace 'bound' with 'pushout' and 'fitting' for some property of an arrow.

Proposition 3.9 (semi-fitting bound). *Let* (B_0, B_1) *be a fitting bound for* (A_0, A_1) *and* A_1 *be fitting. Then* B_0 *is fitting.*



Proof. We know that J is fitting for the pair (B_0, B_1) and that $B_0 \circ A_0 = B_1 \circ A_1$. In the following, t is a root in J, r_i is a site of B_i and s is a site of A_0 and A_1. v ranges over nodes. We will show that if $K \in C_t$ then $K \in C_t$ is necessary for KR1 and KR2 to hold in B_0.

Let $K \in C_t$. We will assume that $K \in C_t$ is not necessary for KR1 and KR2 to hold in B_0. Since J is fitting, it must then be that $K \in C_t$ is necessary for KR1 and KR2 to hold in B_1. We then have a case split:

- Let $t = B_1(v)$ and $ctrl(v) = K$. If $v \in B_0$ then $t = B_0(v)$ and $K \in C_t$ is necessary for B_0. Otherwise, we have $r_0 = A_0(v)$ and $t = B_0(r_0)$. But by KR1 in A, $K \in C_{r_0}$ and then $K \in C_t$ is necessary for KR2 to hold in B_0.
- Let $t = B_1(r_1)$ and $K \in C_{r_1}$. Now, A_1 is fitting and therefore $K \in C_{r_1}$ is necessary for KR1 or KR2 to hold in A_1. We have a further case split:

– Say $K \in C_{r_1}$ is necessary for KR1 to hold in A_1. Then $r_1 = A_1(v)$ and $ctrl(v) = K$. Therefore, either $t = B_0(v)$ and so $K \in C_t$ is required or else $t = B_0(r_1)$ and $r_1 = A_0(v)$ in which case by KR1 in A_0, $K \in C_{r_1}$ and by KR2 in B_0, $K \in C_t$ is required.

– Say $K \in C_{r_1}$ is necessary for KR2 to hold in A_1. Then $r_1 = A_1(s)$ and $K \in C_s$. But then $t = B_0(r_0)$ and $r_0 = A_0(s)$. By KR2 in A_0, $K \in r_0$ and so $K \in C_t$ is required for B_0.

Thus, for all t in J and all $K \in C_t$, $K \in C_t$ is necessary for KR1 and KR2 to hold in B_0. Hence, B_0 is a fitting bigraph. □

This proof becomes significant when A_1 is a ground redex. Then if the set of reaction rules of some kind Brs (a definition of which is not presented here) has fitting redexes and fitting parameters, all labels in the standard transition systems will be fitting. This may be a desirable property.

3.2 Further properties

special IPOs We now present proofs that the special IPOs propositions [7, Propositions 9.8, 9.9, 9.11] hold for kind bigraphs. The propositions themselves are taken verbatim from [ibid.] and so are not repeated in the appendices. We also present a new IPO proposition particular to kind bigraphs.

Proposition 3.10 (containment pushout [9.8]). *Let A be epi. Then the pair $(A, F \circ A)$ has the pair (F, id) as a pushout. In particular, by taking $A = \mathrm{id}$ and $F = \mathrm{id}$ respectively: (1) any pair (id, F) has (F, id) as a pushout, and (2) if A is epi then (A, A) has $(\mathrm{id}, \mathrm{id})$ as a pushout.*

Proof. Let $A :\to I$ is epi and $F \circ A :\to J$ be defined. Then A^{u} is epi and by [7, Proposition 9.8], the pair $(A^{\mathrm{u}}, F^{\mathrm{u}} \circ A^{\mathrm{u}})$ has the pair $(F^{\mathrm{u}}, \mathrm{id})$ as a pushout. The pair $(A, F \circ A)$ is consistent with bound (F, id_J). J is a fitting interface for (F, id_J) as J satisfies KR1 and KR2 exactly for id_J. Thus, by Theorem 3.8, (F, id_J) is an IPO for $(A, F \circ A)$. □

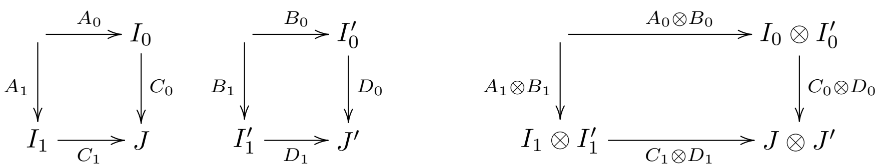
It seems that any pair $(\uparrow\vec{V}_J \circ R, R)$ will have a unique IPO (pushout) (C_0, C_1), where C_0 and C_1 have no nodes and injective link and parent maps. However, we are especially interested in the case where R is epi since it is usually required that redexes in a Brs are epi. We present a related proposition below which will be useful for the dynamic theory of kind bigraphs.

Proposition 3.11 (inflation IPO). *Let $R : I \to J$ be epi. Then the pair $(\uparrow\vec{V}_J \circ R, R)$ has the pair $(\mathrm{id}, \uparrow\vec{V}_J)$ as a pushout.*

Proof. We have $(\uparrow\vec{V}_J \circ R)^{\mathrm{u}} = (\uparrow\vec{V}_J)^{\mathrm{u}} \circ R^{\mathrm{u}} = \mathrm{id} \circ R^{\mathrm{u}} = R^{\mathrm{u}}$. Since R is epi, R^{u} is epi. By Proposition 9.8 [7], $((\uparrow\vec{V}_J \circ R)^{\mathrm{u}}, R^{\mathrm{u}})$ has $(\mathrm{id}, \mathrm{id})$ as a pushout. By Theorem 3.8, $(\uparrow\vec{V}_J \circ R, R)$ has a pushout (C_0, C_1), where both bigraphs have a bijective *prnt* map and an identity *link* map. By Theorem 3.8, the outer interface of (C_0, C_1) is fitting for $(\uparrow\vec{V}_J \circ R, R)$ – this implies that $C_0 = \mathrm{id}$ and $C_1 = \uparrow\vec{V}_J$. □

The proposition above may be useful when discussing the standard transition system of a kind Brs (to be defined in a future report.) For example, let $a = (\uparrow\vec{V}_J \circ r)$ for some ground redex r *i.e.* a is an 'inflated redex'. Then there is a unique transition in ST with respect to that redex and the label is id.

Proposition 3.12 (tensor IPO [9.9]). *In any of ´KPG, ´KPG$_h$, ´LIG, ´KBG or ´KBG$_h$, let $\vec{C}$ be an IPO for $\vec{A}$ and $\vec{D}$ be an IPO for $\vec{B}$, where the supports of the two IPOs are disjoint. Then, provided the tensor products exist, $\vec{C} \otimes \vec{D}$ is an IPO for $\vec{A} \otimes \vec{B}$.*



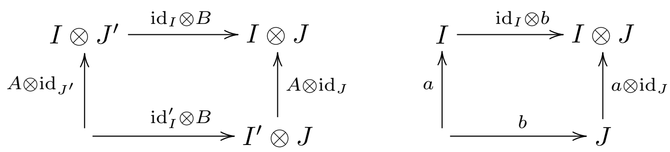
Proof. We present the proof for ´KBG. Let $\vec{C}$ be an IPO for $\vec{A}$ and $\vec{D}$ be an IPO for $\vec{B}$, where the supports of the two IPOs are disjoint. By Theorem 3.8, $\vec{C}^{\mathrm{u}}$ is an IPO for $\vec{A}^{\mathrm{u}}$ and $\vec{D}^{\mathrm{u}}$ is an IPO for $\vec{B}^{\mathrm{u}}$. By [7, Proposition 9.9], $\vec{C}^{\mathrm{u}} \otimes \vec{D}^{\mathrm{u}}$ is an IPO for $\vec{A}^{\mathrm{u}} \otimes \vec{B}^{\mathrm{u}}$. ´KBG is wide monoidal so $\vec{C} \otimes \vec{D}$ is a bound for $\vec{A} \otimes \vec{B}$ as

$$\begin{aligned}(C_0 \otimes D_0) \circ (A_0 \otimes B_0) &= (C_0 \circ A_0) \otimes (D_0 \circ B_0) \\ = (C_1 \circ A_1) \otimes (D_1 \circ B_1) &= (C_1 \otimes D_1) \circ (A_1 \otimes B_1).\end{aligned}$$

J is fitting for $\vec{C}$ as $\vec{C}$ is an IPO for $\vec{A}$. Likewise, J' is fitting for $\vec{D}$. Thus, $J \otimes J'$ is fitting for $\vec{C} \otimes \vec{D}$. By Theorem 3.8, $\vec{C} \otimes \vec{D}$ is an IPO for $\vec{A} \otimes \vec{B}$. ☐

Corollary 3.13 (tensor IPOs with identities [9.10]). *Let $A : I' \to I$ and $B : J' \to J$ share no nodes, and let the names of I', I be disjoint from those of J', J. Then the pair $(A \otimes \mathrm{id}_{J'}, \mathrm{id}_{I'} \otimes B)$ has an IPO $(\mathrm{id}_I \otimes B, A \otimes \mathrm{id}_J)$. See diagram (a).*

In particular if $I' = J' = \epsilon$ then $A = a$ and $B = b$ are ground bigraphs, and the IPO as in diagram (b).



(a) (b)

As in the pure theory, we shall (1) call a kind bigraph *lean* if its link graph is lean and (2) denote the result of adding a set E of fresh, idle edges to a kind bigraph A as A^E.

For the following proofs, note that U preserves and reflects leanness *i.e.* A is lean iff A^{u} is lean, as $A^{\mathrm{L}} = (A^{\mathrm{u}})^{\mathrm{L}}$. Note also that A and A^E are in the same homset — in particular, their outer interfaces are equal — and that adding or removing idle edges to a consistent pair of bigraphs (pure or kind) yields a consistent pair of bigraphs.

Proposition 3.14 (IPOs, idle edges and leanness [9.11]). *For any two pairs $\vec{A}$ and $\vec{B}$:*

1. *If $\vec{B}$ is an IPO for $\vec{A}$, and A_1 is lean, then B_0 is lean.*
2. *For any fresh set E of edges, $\vec{B}$ is an IPO for $\vec{A}$ iff (B_0, B_1^E) is an IPO for (A_0^E, A_1).*

Proof. 1. Let $\vec{B}$ be an IPO for $\vec{A}$ and A_1 be lean. Then $\vec{B^{\mathrm{u}}}$ is an IPO for $\vec{A^{\mathrm{u}}}$ and ${A_1}^{\mathrm{u}}$ is lean. By [7, Proposition 9.11.1], ${B_0}^{\mathrm{u}}$ is lean. ${B_0}^{\mathrm{L}} = ({B_0}^{\mathrm{u}})^{\mathrm{L}}$. Thus, B_0 is lean.

2. $\Rightarrow$ Let $\vec{B}$ be an IPO for $\vec{A}$. Then $\vec{B^{\mathrm{u}}}$ is an IPO for $\vec{A^{\mathrm{u}}}$. By [7, Proposition 9.11.1], $({B_0}^{\mathrm{u}}, {B_1^E}^{\mathrm{u}})$ is an IPO for $({A_0^E}^{\mathrm{u}}, {A_1}^{\mathrm{u}})$. $\vec{A}$ is a consistent pair of kind bigraphs. Thus, (A_0^E, A_1) is a consistent pair of kind bigraphs. Let $\vec{B}$ have outer interface J. J is fitting for $\vec{B}$ as $\vec{B}$ is an IPO for $\vec{A}$. (B_0, B_1^E) also has outer interface J, which is fitting for (B_0, B_1^E). Thus, by Theorem 3.8, (B_0, B_1^E) is an IPO for (A_0^E, A_1).

   $\Leftarrow$ Similar.

□

Lean-support equivalence for kind bigraphs remains unchanged from the pure theory *i.e.* "two concrete (kind) bigraphs A and B are *lean-support equivalent*, written $A \risingdotseq B$, if after discarding any idle edges they are support equivalent" [7]. Lean-support equivalence is a static congruence and so we have quotient functors as follows.

abstract kind bigraphs [9.12] An abstract kind bigraph is a lean-support equivalence class of concrete kind bigraphs. We can define the category $\mathrm{KBG}(\mathcal{K})$ (resp. $\mathrm{KBG_h}(\mathcal{K})$) of *abstract kind (hard) bigraphs* as having the same objects as $\acute{\mathrm{K}}\mathrm{BG}(\mathcal{K})$ (resp. $\acute{\mathrm{K}}\mathrm{BG_h}(\mathcal{K})$), where its arrows are lean-support equivalence classes of concrete kind bigraphs. For any signature $\mathcal{K}$, we have quotient functors $[\![\cdot]\!] : \acute{\mathrm{K}}\mathrm{BG}(\mathcal{K}) \to \mathrm{KBG}(\mathcal{K})$ and $[\![\cdot]\!] : \acute{\mathrm{K}}\mathrm{BG_h}(\mathcal{K}) \to \mathrm{KBG_h}(\mathcal{K})$ sending a concrete (hard) kind bigraph to its lean-support equivalence class.

ground bigraphs A *ground* bigraph is one with inner face ϵ, $\langle 0, (), \emptyset \rangle$.

prime interfaces, bigraphs An interface $I = \langle m, \vec{C_I}, X \rangle$ is prime if it has width $m = 1$. A prime kind bigraph has no inner names and a prime outer face.

merge The definition of the prime *merge* is modified to exactly satisfy KR2. It is defined as $\mathit{merge} : \langle m, \vec{C_I}, \emptyset \rangle \to \langle 1, (C_0), \emptyset \rangle$ where $C_0 = C_{I,0} \cup \cdots \cup C_{I,m-1}$. *merge* has no nodes and maps m sites to a single root which can contain the union of the controls that its children can contain. *merge* is a fitting bigraph.

wirings and discreteness Wirings and discreteness are as before and pertain only to link graphs.

ions, atoms, molecules We define an ion as in the pure theory except that the site can contain all controls that the node can. For any non-atomic control K with arity k and sequence $\vec{x}$ of k distinct names, we define the *discrete ion* $K_{v,\vec{x}} : \langle 1, (kind(K)), \emptyset\rangle \to \langle 1, (\{K\}), \vec{x}\rangle$ to have a single K-node v, whose ports are severally linked to $\vec{x}$. The site is a child of the node. For atomic K, a *discrete atom* is $K_{v,\vec{x}} : \epsilon \to \langle 1, (\{K\}), \vec{x}\rangle$, again containing a single K-node v whose ports are severally linked to $\vec{x}$.

For any prime discrete P with outer face $\langle 1, (C_0), Y\rangle = \langle 1, (kind(K)), Y\rangle$ we call $(K_{v,\vec{x}} \otimes \mathrm{id}_Y) \circ P$ a *discrete molecule*.

Ions, atoms and molecules are defined to be discrete and fitting. Arbitrary (non-discrete, non-fitting) ions, molecules and atoms are constructed by composing $\omega \otimes \uparrow \vec{V}_I$, where ω is a wiring, with a discrete ion, molecule or atom. Other arbitrary ions and molecules can be constructed by deflating the site of the ion.

Our final section of the static theory for kind bigraphs defines some operations and decompositions of pure bigraphs for kind bigraphs.

3.3 Operations and decompositions

parallel product [9.13, 9.14] We extend the definition of parallel product for pure bigraphs. The *parallel product* of two interfaces is defined as

$$\langle m, \vec{C_0}, X\rangle \| \langle n, \vec{C_1}, Y\rangle \stackrel{\text{def}}{=} \langle m+n, \vec{C_0}\vec{C_1}, X \cup Y\rangle.$$

The parallel product of two bigraphs is identical to the pure definition. We repeat it here:

$$G_0 \| G_1 \stackrel{\text{def}}{=} \langle G_0^{\mathrm{K}} \otimes G_1^{\mathrm{K}}, G_0^{\mathrm{L}} | G_1^{\mathrm{L}}\rangle : I_0 \otimes I_1 \to J_0 \| J_1$$

when the interfaces exist and the node sets are disjoint. The parallel product can also be defined as in [7, Proposition 9.14].

Again, $\|$ is associative (we need only check that $(\vec{C_0}\vec{C_1})\vec{C_2} = \vec{C_0}(\vec{C_1}\vec{C_2})$), with unit ϵ. Note that, as for pure bigraphs, the parallel product requires that node sets are disjoint but does not require that the edge sets are disjoint. This is because the *link* map of $G_0^{\mathrm{L}} | G_1^{\mathrm{L}}$ is defined as the union of the constituent link maps.

The parallel product of two fitting bigraphs is a fitting bigraph.

prime product [9.15] The *prime product* of two interfaces is defined as

$$\langle m_0, \vec{C_{I_0}}, X_0\rangle | \langle m_1, \vec{C_{I_1}}, X_1\rangle \stackrel{\text{def}}{=} \langle 1, (C_0), X_0 \cup X_1\rangle$$

where

$$C_0 = C_{I_0,0} \cup \cdots \cup C_{I_0,m_0-1} \cup C_{I_1,0} \cup \cdots \cup C_{I_1,m_1-1}.$$

For two prime bigraphs $\vec{P} : \vec{I} \to \vec{J}$ with disjoint support, if $P_0 || P_1$ is defined and n is the sum of the widths of J_0 and J_1, the *prime product* is defined, as in pure bigraphs, as

$$P_0 | P_1 \stackrel{\text{def}}{=} merge_n \circ (P_0 \| P_1) : I_0 \otimes I_1 \to J_0 | J_1.$$

$|$ is associative, with unit $\langle 1, (\emptyset), \emptyset\rangle$ when applied to primes. The prime product of two fitting bigraphs P_0, P_1 is a fitting bigraph. This can also be seen by observing that both $P_0 \| P_1$ and $merge_n$ are fitting.

underlying discrete bigraph [9.16] The factorisation of a bigraph G to a discrete normal form presented in the pure theory remains unchanged as it pertains only to the link graph.

As an aside, in the equation $G = (\omega \otimes \mathrm{id}_n) \circ D$, D can be described as 'taking G and pulling the wiring to the outer interface' and $(\omega \otimes \mathrm{id}_n)$ can be described as 'rejoining these wires as open or closed links'.

It may be necessary, perhaps for an axiomatisation of kind bigraphs, to define something akin to discreteness for kind bigraphs. This notion follows.

Proposition 3.15 (underlying discrete, fitting (kind) bigraph). *Every kind bigraph G in ´KBG or ´KBG$_{\mathrm{h}}$ can be expressed uniquely (up to iso) as $G = (\omega \otimes \uparrow\vec{V}) \circ D$, where ω is a wiring, D is discrete and fitting, and $\uparrow\vec{V}$ has no names.*

Composition and tensor product preserve discreteness, as for pure bigraphs, and thus in ´KBG and ´KBG$_{\mathrm{h}}$, the discrete bigraphs form a monoidal sub-s-category. Likewise, the composition of two fitting bigraphs is a fitting bigraph and the tensor product of two fitting bigraphs is a fitting bigraph. Thus we have that in ´KBG and ´KBG$_{\mathrm{h}}$, the discrete, fitting bigraphs form a sub-s-category ´FKB (as $\gamma_{I,J}$ is a discrete, fitting bigraph.)

We have extended the next proposition slightly to give another factorisation of a discrete bigraph which includes fitness. Parts 1, 3 and 4 of the proposition are as in the pure theory (omitting the vectors $\vec{C}$) and so are not reproduced in the appendices. For the following proofs, note that A is discrete iff A^{u} is discrete as they have the same link graph.

Proposition 3.16 (synthesis and analysis of discrete bigraphs [9.17]). *In ´KBG or ´KBG$_{\mathrm{h}}$ the discrete kind bigraphs form a monoidal sub-s-category. Moreover*

1. *Every discrete $D : \langle m, \vec{C_I}, X\rangle \to \langle n, \vec{C_J}, Y\rangle$ may be factored uniquely, up to isomorphism on the domain of each factor D_i, as*

$$D = \alpha \otimes ((D_0 \otimes \cdots \otimes D_{n-1}) \circ \pi)$$

   with α a renaming, each D_i prime and discrete, and π a permutation.

2. *Every discrete $D : \langle m, \vec{C_I}, X\rangle \to \langle n, \vec{C_J}, Y\rangle$ may be factored uniquely, up to isomorphism on the domain of each factor D_i, as*

$$D = \alpha \otimes (\uparrow\vec{V} \circ (D_0 \otimes \cdots \otimes D_{n-1}) \circ \pi)$$

   with α a renaming, $\uparrow\vec{V}$ an inflation, each D_i prime, discrete and fitting, and π a permutation.

3. *If (D', G') is an IPO for (G, D) and D is discrete, then D' is discrete.*

4. *If $D' \circ G = \omega D$ with D and D' discrete, then (D', ω) is an IPO for (G, D).*

Proof. 1. We will explain this factorisation for pure bigraphs and then it will be obvious that the same factorisation applies to kind bigraphs. Note that for any kind bigraph $G = G_0 \otimes \cdots \otimes G_{n-1}$, each G_i is also a kind bigraph *i.e.* the kind rules hold.

D is discrete. This means that it has no edges and a bijective *link* map $link_D : X \uplus P \rightarrow Y$, where P is the set of ports of nodes of D. We can split the set of ports P into n disjoint subsets $P_0, \ldots, P_{n-1}$, such that

$$P_i \stackrel{\text{def}}{=} \sum_{G(v)=i} ar(ctrl(v)), \qquad P = \biguplus_{i \in n} P_i,$$

i.e. P_i is the set of ports of nodes under root i. We can also split Y into $n+1$ disjoint subsets $Y_X, Y_{P_0}, \ldots, Y_{P_{n+1}}$ such that

$$Y = Y_X \uplus Y_{P_0} \uplus \cdots \uplus Y_{P_{n-1}}$$

where Y_X is the image of X under $link_D$ and Y_{P_i} is the image of P_i under $link_D$. We can now form n bijections $link_{D_i} : P_i \rightarrow Y_{P_i}, i \in n$, from ports to names, defined by

$$link_{D_i}(p) = y \text{ when } link_D(p) = y.$$

We define one final bijection $\alpha : X \rightarrow Y_X$ as $\alpha(x) = link_D(x)$. We have now essentially split up the bijection $link_D$ into n bijections for each root of D and one bijection α mapping the inner names of D to Y_X. α is the renaming in the factorisation.

Let m_i be the set of sites under root $i \in n$. Let $\vec{C_{I_i}}$ be the vector of subsets of $\mathcal{K}$ corresponding to m_i. Each D_i is prime and discrete and is defined as $D_i : \langle m_i, \vec{C_{I_i}}, \emptyset\rangle \rightarrow \langle 1, (C_i), Y_{P_i}\rangle$ with the obvious *prnt* map and $link_{D_i}$ as the link map. We can picture D_i as the root i with all its nodes and each port of the nodes having a unique wire which is pulled to a name on the outer interface. There are no other 'wires' in D_i.

Now, each D_i is prime so they will have no inner names. Hence, the tensor product $D_0 \otimes \cdots \otimes D_{n-1}$ has no inner names. This is why α is necessary — α 'supplies' the inner names X and the outer names Y_X for D. This can best be seen by looking at Figure 6. It is necessary to separate out this bijection from inner names X to Y_X from the discrete D_i as otherwise a *unique* factorisation would not be possible — each D_i could have some subset of X in its inner interface and some bijective (to X) subset of Y_X in its outer interface and still be discrete.

The permutation (isomorphism) $\pi : \langle m, \vec{C_I}\rangle \rightarrow \langle m, \vec{C_{I'}}\rangle$ is defined as for pure bigraphs, where if $\pi(i) = j$ then $C_{I',j} = C_{I,i}$. This permutation is needed for a simple reason. By the definition of tensor product, the sites of $D_0 \otimes \cdots \otimes D_{n-1}$ are ordered so that a site in D_i is 'less than' (in terms of the ordinal m) a site in D_j where $i < j$. The permutation π merely reorders the sites as necessary.

2. This factorisation is the same as the last except that we make each D_i fitting. $\uparrow\vec{V}$ is then the obvious, unique inflation.

3. Let (D', G') be an IPO for (G, D) and D be discrete. Then $({D'}^{\mathrm{u}}, {G'}^{\mathrm{u}})$ is an IPO for $(G^{\mathrm{u}}, D^{\mathrm{u}})$ and D^{u} is discrete. By [7, Proposition 9.17.2], ${D'}^{\mathrm{u}}$ is discrete. Thus, D' is discrete.

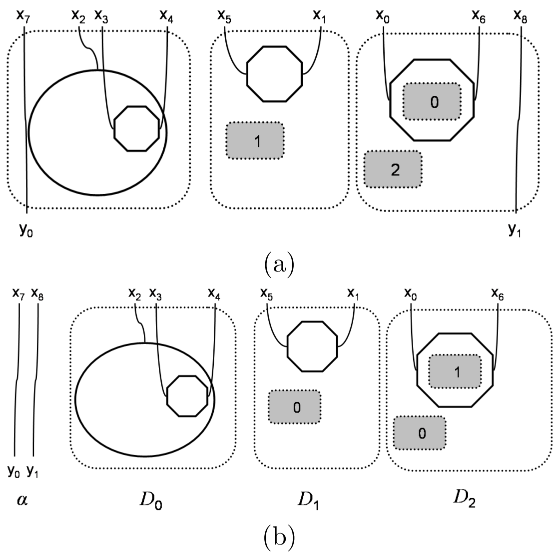
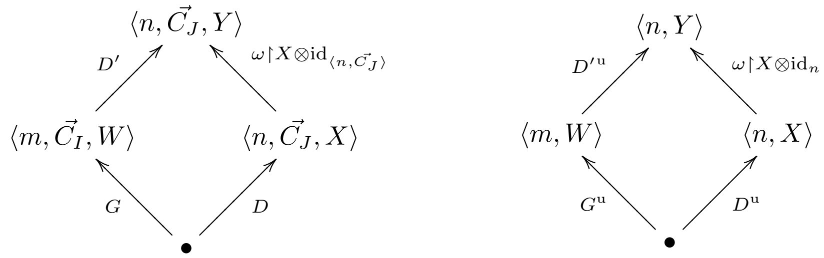


Figure 6: (a) A discrete bigraph D (b) The elements of the factorisation (where $\pi(0) = 2, \pi(1) = 0, \pi(2) = 1$)

4.



D and D' are discrete so D^{u} and D'^{u} are discrete. Hence $(D'^{\mathrm{u}}, \omega)$ is an IPO for $(G^{\mathrm{u}}, D^{\mathrm{u}})$ by [7, Theorem 9.17.3].

(D', ω) is a bound for (G, D). Thus, the pair (G, D) is consistent. The place graph component of ω is an identity and so J is fitting for (D', ω). By Theorem 3.8, (D', ω) is an IPO for (G, D).

□

The remainder of this section is self-contained and any propositions are copied directly from the pure theory. The reader is encouraged to refer to [7] for some more details on the following propositions.

instantiation [9.18] An instantiation ϱ from m to n, which is written $\varrho :: m \to n$, is determined by a function $\bar{\varrho} : n \to m$. For any X this function defines the map

$$\varrho : \mathrm{Gr}\langle m, (C_0 \cdots C_{m-1}), X\rangle \to \mathrm{Gr}\langle n, (C_{\bar{\varrho}(0)} \cdots C_{\bar{\varrho}(n-1)}), X\rangle$$

as follows. Decompose $g : \epsilon \to \langle m, (C_0 \cdots C_m), X\rangle$ into $g = \omega(d_0 \otimes \cdots \otimes d_{m-1})$, with $Y = \biguplus_{i \in m} Y_i$, $\omega : Y \to X$ and each $d_i : \epsilon \to \langle 1, (C_i), Y_i\rangle$ prime and discrete. Then define

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

where $e_j \simeq d_{\bar{\varrho}(j)}$ for $j \in n$.

Recall that support equivalence is defined on homsets and so if $G \simeq G'$ then the inner and outer $\vec{C}$ vectors of both bigraphs are respectively identical.

Note also that since each d_i is discrete, it has no edges and so the e_j in the definition also have no edges. Support equivalence here is just a renaming of the node set of d_i. This also means that in the bigraph $\varrho(g)$, multiple copies of a bigraph d_i will share the same edges.

wiring an instance [9.19] The proof that wiring commutes with instantiation *i.e*,

$$\omega\varrho(a) \simeq \varrho(\omega a),$$

remains as before. We present a longer version of the original proof below. For brevity, we write id_m and $\mathrm{id}_{m'}$ for $\mathrm{id}_{\langle m, \vec{C_I}\rangle}$ and $\mathrm{id}_{\langle m', \vec{C_I'}\rangle}$ respectively.

Proof. Let $a : \langle m, \vec{C_I}, X\rangle$, with $\varrho :: m \to m'$. Take the DNF

$$a = (\omega' \otimes \mathrm{id}_m)d = (\omega' \otimes \mathrm{id}_m)(d_0 \otimes \cdots \otimes d_{m-1}).$$

Then

$$\varrho(a) = (\omega' \otimes \mathrm{id}_{m'})a' = (\omega' \otimes \mathrm{id}_{m'})(d'_0 \| \cdots \| d'_{m'-1})$$

with each $d'_i \simeq d_{\bar{\varrho}(i)}$. So

$$\begin{aligned}
\varrho(\omega a) &= \varrho((\omega \otimes \mathrm{id}_m)a) \\
&\quad \langle \text{definition of } a\rangle \\
&= \varrho((\omega \otimes \mathrm{id}_m)(\omega' \otimes \mathrm{id}_m)(d_0 \otimes \cdots \otimes d_{m-1})) \\
&\quad \langle\, \text{'}\textsc{Kbg} \text{ is monoidal}\rangle \\
&= \varrho\Big(\big((\omega \circ \omega') \otimes (\mathrm{id}_m \circ \mathrm{id}_m)\big)(d_0 \otimes \cdots \otimes d_{m-1})\Big) \\
&\quad \langle \text{identity law}\rangle \\
&= \varrho\big(((\omega \circ \omega') \otimes \mathrm{id}_m)(d_0 \otimes \cdots \otimes d_{m-1})\big) \\
&\quad \langle \text{definition of } \varrho\rangle \\
&\simeq ((\omega \circ \omega') \otimes \mathrm{id}_{m'})(d'_0 \| \cdots \| d'_{m'-1}) \\
&\quad \langle \text{identity law}\rangle \\
&= ((\omega \circ \omega') \otimes (\mathrm{id}_{m'} \circ \mathrm{id}_{m'}))(d'_0 \| \cdots \| d'_{m'-1}) \\
&\quad \langle\, \text{'}\textsc{Kbg} \text{ is monoidal}\rangle \\
&= ((\omega \otimes \mathrm{id}_{m'}) \circ (\omega' \otimes \mathrm{id}_{m'}))(d'_0 \| \cdots \| d'_{m'-1}) \\
&\quad \langle \text{associativity}\rangle \\
&= (\omega \otimes \mathrm{id}_{m'}) \circ ((\omega' \otimes \mathrm{id}_{m'}))(d'_0 \| \cdots \| d'_{m'-1})) \\
&\quad \langle \text{definition of } \varrho(a)\rangle \\
&\simeq (\omega \otimes \mathrm{id}_{m'}) \circ \varrho(a) = \omega\varrho(a)
\end{aligned}$$

□

Proposition 3.17 (wiring a product [9.20]). *Wiring commutes with parallel and prime product; that is,*

$$\omega(F \parallel G) = \omega F \parallel \omega G \text{ and } \omega(F \mid G) = \omega F \mid \omega G$$

Proof. These properties affect the link graph and remain true. □

Proposition 3.18 (instantiating a product [9.21]). *Let* $a_i : \epsilon \to \langle 1, (C_i), Y_i \rangle$ *be prime and ground* ($i \in m$)*, and let* $Y = \bigcup_{i \in m} Y_i$*. Let* $\varrho :: m \to n$ *be an instantiation. Then*

$$\varrho(a_0 \parallel \cdots \parallel a_{m-1}) = Y \parallel b_0 \parallel \cdots \parallel b_{n-1}$$

where $b_j \simeq a_{\bar{\varrho}(j)}$ *for* $j \in n$.

Proof. The proof proceeds as for pure bigraphs. We present it here.

Express a_i in DNF as $\omega_i d_i$ where

$$d_i : \epsilon \to \langle 1, (C_i), X_i \rangle, \omega_i : X_i \to Y_i,$$

and the discrete d_i have disjoint name sets.

A crucial point is that each d_i has a name set disjoint from any other d_j. We can then define a wiring $\omega : X_0 \uplus \cdots \uplus X_{m-1} \to Y = \omega_0 \| \cdots \| \omega_{m-1}$.

$$\begin{array}{cc} & \varrho(a_0 || \cdots || a_{m-1}) \\ = & \langle \text{DNF} \rangle \\ & \varrho(\omega_0 d_0 || \cdots || \omega_{m-1} d_{m-1}) \\ = & \langle \text{definition of wiring} \rangle \\ & \varrho(\omega d_0 || \cdots || \omega d_{m-1}) \\ = & \langle \text{Proposition 9.20} \rangle \\ & \varrho(\omega(d_0 || \cdots || d_{m-1})) \\ = & \langle \text{Proposition 9.19} \rangle \\ & \omega(\varrho(d_0 || \cdots || d_{m-1})) \\ = & \langle \text{the name sets } X_j \text{ are disjoint} \rangle \\ & \omega(\varrho(d_0 \otimes \cdots \otimes d_{m-1})) \\ = & \langle \text{definition of instantiation} \rangle \\ & \omega(e_0 \| \cdots \| e_{n-1}) \\ = & \langle \text{Proposition 9.20} \rangle \\ & \omega e_0 \| \cdots \| \omega e_{n-1} \\ = & Y \| \omega_{\bar{\varrho}(0)} e_0 \| \cdots \| \omega_{\bar{\varrho}(n-1)} e_{n-1} \end{array}$$

Since $e_j \simeq d_{\bar{\varrho}(j)}$, we have $\omega_{\bar{\varrho}(j)} e_j \simeq \omega_{\bar{\varrho}(j)} d_{\bar{\varrho}(j)} \simeq a_{\bar{\varrho}(j)}$. □

Note that the empty substitution $Y : \epsilon \to Y$ is added in the last line of equations in the proof since some names in Y may have been discarded by the instantiation.

Proposition 3.19 (instantiating with prime component [9.22]). *Let*

$$G : \langle 1, (C_{B,i}), X \rangle \to \langle m, (C_0 \cdots C_{m-1}), Y \rangle$$

be arbitrary with prime inner face, and $\varrho :: m \to n$ *be an instantiation. Then for some* $k \geq 0$*, if we choose disjoint renamings* $\alpha_i : X \to X_i (i \in k)$*, there exists a context* $C : \langle k, (C_{B,i} \cdots C_{B,i}), \bigcup_i X_i \rangle \to \langle n, (C'_0 \cdots C'_{n-1}), Y \rangle$ *such that*

$$\varrho(G \circ a) \simeq C \circ (a_0 \otimes \cdots \otimes a_{k-1})$$

whenever $G \circ a$ *is defined, where* $a_i \simeq \alpha_i a$.

Moreover for any pair $a, b : \epsilon \to \langle 1, (C_s), X \rangle$ *we have* $(\varrho(G \circ a), \varrho(G \circ b)) \in (\mathcal{S}^{\simeq})^*$*, where*

$$\mathcal{S} = \{(H \circ a, H \circ b) | H \text{ any context}\}.$$

Proof. The proof proceeds as for pure bigraphs. We present below a longer version than originally presented, with some notational changes to suit our conventions.

We can express G as a product of prime discrete factors where all but one factor is ground. We call this non-ground, discrete factor $B : \langle 1, (C_{B,i}), X \rangle \to \langle 1, (C_{B,o}), Z_B \rangle$ where $C_{B,o} = C_j$ for some $j \in m$. We call the remaining factors $d_1, \ldots, d_{m-1}$. The factorisation of G is

$$G = (\omega \otimes \pi) \circ D$$

$$D = (B \otimes d_1 \otimes \cdots \otimes d_{m-1}).$$

If we compare the factorisation of D to [7, Proposition 9.16] and Proposition 3.16 we notice that two components of the discrete factorisation are missing. First, there is no inner permutation. This permutation is not necessary since there is only one site in G — the permutation then becomes an identity. Second, the renaming which usually supplies the inner names of D is missing. This renaming is in fact part of the factor B, whose inner names are X.

The permutation (isomorphism) $\pi : \langle m, \vec{C'_I} \rangle \to \langle m, \vec{C_I} \rangle$ in the above factorisation is used for convenience so that we can list B as the first factor.

We can compose G with a to get

$$G \circ a = (\omega \otimes \pi) \circ ((B \circ a) \otimes d_1 \otimes \cdots \otimes d_{m-1}),$$

where $B \circ a, d_1, \ldots, d_{m-2}$ and d_{m-1} are prime and ground. The only edges in the factors $((B \circ a), d_1, \ldots, d_{m-2}$ and $d_{m-1})$ are in $B \circ a$ and the sets of outer names of the factors as disjoint. We can therefore rewrite the above equation as

$$G \circ a = (\omega \otimes \pi) \circ ((B \circ a) \| d_1 \| \cdots \| d_{m-1})$$

and, by Proposition [7, Proposition 9.21],

$$\varrho(G \circ a) \simeq (\omega \otimes \pi') \circ (e_0 \| \cdots \| e_{k-1} \| d'_k \| \cdots \| d'_{n-1} \| Z)$$

where π' is used only for a convenient ordering, $d'j, k \leq j < n$ is support-equivalent to some $d_l, 1 \leq l < m$, and

$$e_i \simeq \epsilon \xrightarrow{a} \langle 1, (C_{B,i}), X \rangle \xrightarrow{B} \langle 1, (C_{B,o}), Z_B \rangle.$$

We rewrite each $e_i : \epsilon \to \langle 1, (C_{B,o}), Z_B \rangle$ as

$$e_i : \epsilon \xrightarrow{a} \langle 1, (C_{B,i}), X \rangle \xrightarrow{\alpha_i} \langle 1, (C_{B,i}), X_i \rangle \xrightarrow{B_i} \langle 1, (C_{B,o}), Z_B \rangle$$

to obtain an expression for $\varrho(G \circ a)$ involving several support-disjoint copies of a. The required context is then

$$C \simeq (\omega \otimes \pi') \circ (B'_0 \| \cdots \| B'_{k-1} \| d'_k \| \cdots \| d'_{n-1} \| Z)$$

where $B'_i \simeq B_i$.

The second part of the proof proceeds as the original proof, noting that $\langle \alpha_i \rangle$ has the type $\alpha_i : \langle 1, (C_{B,i}), X \rangle \to \langle 1, (C_{B,i}), X_i \rangle$. □

We have taken the static theory of kind bigraphs up to the point where it seems feasible that the dynamic theory should work out. We will continue this direction in a future technical report. For the remainder of this report, we will investigate the connection between fitting bigraphs and place-sorted bigraphs [14].

4 Fitting bigraphs

In this section, we will concentrate on the s-categories $\text{'}\mathrm{FKB}(\mathcal{K})$ of fitting bigraphs over a kind signature $\mathcal{K}$, each of which is a sub-s-category of $\text{'}\mathrm{KBG}(\mathcal{K})$. There are many reasons why we wish to restrict our attention to fitting bigraphs. One reason is simply to reduce the number of arrows in the homsets, leading to a reduction in the number of labels in some transition systems, specifically the full transition system. Another reason is to give rise to a more elegant static theory — it is unnecessary to state that a root of a kind bigraph G is able to be a parent of a certain control if it never will be for any composition $G \circ H$. As we shall see, the most compelling reason to study fitting bigraphs is that they have certain properties that kind bigraphs do not.

We begin in Section 4.1 by introducing a new link-sorting called *tile-sorting*. It was derived so that some tile-based games where the tiles have an even number of sides may be directly represented as kind bigraphs over this link-sorting. We do not formally combine the tile-sorting with kind place graphs in this work but we prove that it has two properties which guarantee its usefulness. The intuition behind tile sorting is that it has undirected points as in pure link graph theory which may link together at will. It also has directed points, which may only link together in pairs of opposite directions. We have thus, in some sense, orthogonally combined a generalised *directed linear* link graph [9] with a pure link graph. The application presented here may be trivial but such a link-sorting intuitively seems to have wider applications.

In Section 4.2, we summarise Milner's recent theory of place-sorting [14] and compare the notion of 'sort' to that of 'kind'. We then show how a useful place-sorting, known as a homomorphic sorting, can be described using a kind signature. A homomorphic sorting was used to facilitate the encoding of finite CCS in [ibid.]. We finally explain how the fitting bigraphs over a kind signature are a well-behaved place sorting which, as a consequence of results in [ibid.], immediately gives us some nice congruence properties.

In Section 4.3, we give some examples of Brss of fitting bigraphs over a kind signature, showing how their reaction rules can, in some sense, be designed to fire only if certain logical statements regarding the nesting of places are satisfied.

The main disadvantage of restricting ourselves to fitting bigraphs is that reaction rules with fitting redexes and reactums are less expressive in general than reaction rules with kind redexes and reactums. In Section 4.4, we briefly discuss this problem and present a conjectured solution to some cases.

For the remainder of this section, we will assume that we are working with hard bigraphs. This section mainly serves as a motivation for studying kind bigraphs. Therefore, we will discuss and use definitions from the dynamic theory of bigraphs and wide reactive systems which we will not explain in this document, leaving any detailed explanations for the report on the dynamic theory of kind bigraphs. The reader is encouraged to refer to [7, 14] for the necessary material.

4.1 Tile sorting

This work on kind bigraphs arose due to a remark made by Jensen and Milner [7]. Since that document, Leifer and Milner [9] have introduced *sorted link* *graphs* and Milner [14] has introduced *place-sorted graphs* in order to provide the theory to deal with many future applications of bigraphs.

In sorted link graphs, names (both inner and outer) and ports[5] are assigned a *sort* and a *sorting discipline* enforces constraints on which link graphs are 'allowed'. Composition and tensor product of two link graphs, sorted under the same discipline, yield a similarly sorted link graph. One example of sorted link graphs given in [9] is *undirected linear* link graphs, where all names and ports have the same sort but where every outer name is linked to exactly one point, and every edge is linked to exactly two points. Another example, using two sorts, is *directed linear* link graphs, where a polarity is assigned to each port of a control and ports may only be linked when their polarities are opposite. A particular strain of sorted link graphs were used in [ibid.] to derive a behavioural congruence for condition-event Petri nets, using some very nice techniques to mimic experiments on the nets in the bigraphical setting.

We informally discussed a sorting similar to directed linear link graphs when we talked about the expressiveness of kind bigraphs earlier in this report where we used two pairs of sorts with opposite polarities. We will now briefly delve into the theory of link sorting in order to justify some of our examples in Section 4.3 by presenting a formal definition of the link sorting used in that section. We first repeat the original definitions of sorted link graphs and sorting, with their original numbering. We change some notation to fit our application. The original definitions are defined only in terms of link graphs. We work under the assumption that we may orthogonally combine sorted link graphs with kind place graphs.

In the following, Ψ will denote a non-empty set of *sorts*, and ψ will range over Ψ.

[Definition 8.1 (sorted link graphs) [9]] A signature $\mathcal{K}$ is Ψ*-sorted* if it is enriched by an assignment of a sort $\psi \in \Psi$ to each $i \in ar(K)$ for each control K. An interface X is Ψ-sorted if it is enriched by ascribing a sort to each name $x \in X$.

A link graph is Ψ*-sorted over* $\mathcal{K}$ if its interfaces are Ψ-sorted, and for each K, i the sort assigned by $\mathcal{K}$ to $i \in ar(\mathcal{K})$ is ascribed to the i^{th} port of every K-node.

$\text{'LIG}(\Psi, \mathcal{K})$ denotes the monoidal precategory of sorted link graphs whose identities, composition and tensor product are defined in the obvious way in terms of the underlying (unsorted) link graphs.

[Definition 8.2 ((link) sorting) [9]] A *(link)-sorting (discipline)* is a triple

$$\Sigma = (\Psi, \mathcal{K}, \Xi)$$

where $\mathcal{K}$ is Ψ-sorted, and Ξ is a condition on Ψ-sorted link graphs over $\mathcal{K}$. The condition Ξ must be satisfied by the identities and preserved by both composition and tensor product.

A link graph in $\text{'LIG}(\Psi, \mathcal{K})$ is said to be Σ*-(link-)sorted* if it satisfies Ξ. The Σ-sorted link graphs form a monoidal sub-precategory of $\text{'LIG}(\Psi, \mathcal{K})$ denoted

[5] Sorts are assigned to each port of a *control*. The i^{th} port of any node with control K has the sort assigned to the i^{th} port of K.

by $\acute{\mathrm{L}}\textsc{ig}(\Sigma)$. Further, if $\acute{\mathcal{R}}$ is a set of Σ-sorted reaction rules then $\acute{\mathrm{L}}\textsc{ig}(\Sigma, \acute{\mathcal{R}})$ is a Σ-*sorted* LRS.

Associated with a link-sorting is a forgetful functor

$$\mathcal{U} : \acute{\mathrm{L}}\textsc{ig}(\Sigma) \to \acute{\mathrm{L}}\textsc{ig}(\mathcal{K})$$

which discards sorts. Such a functor $\mathcal{U}$ is called a *sorting* functor, is surjective on objects, and is faithful.

We will not repeat the definition of link reactive system (LRS) here but we will define a link-sorting which we call *tile-sorting* as we will use it to build kind Brss with a 'tile' control to model tile-based games.

We presuppose a set Dir of *directions*, a second disjoint set ODir of *opposite* directions such that $|\mathsf{Dir}| = |\mathsf{ODir}|$, and a bijection f between these sets taking each $\mathsf{d} \in \mathsf{Dir}$ to its opposite $\mathsf{o} \in \mathsf{ODir}$. There is no real distinction between directions and their opposites so when we are discussing any particular direction $\mathsf{d} \in \mathsf{Dir} \uplus \mathsf{ODir}$, we use o to denote its opposite direction *i.e.* $\mathsf{o} = f\mathsf{d}$ if $f \in \mathsf{Dir}$, otherwise $\mathsf{o} = f^{-1}\mathsf{d}$.

Our set of link sorts will be $\Psi = \mathsf{Dir} \uplus \mathsf{ODir} \uplus \{\mathsf{a}\}$ where the element a represents an *undirected* sort.

For example, let $\mathsf{Dir} = \{\mathsf{n}, \mathsf{e}\}$ for 'north' and 'east'. Then define $\mathsf{ODir} = \{\mathsf{s}, \mathsf{w}\}$ for 'south' and 'west' and let $f\mathsf{n} = \mathsf{s}$ and $f\mathsf{e} = \mathsf{w}$. This is the tile-sorting we will use for our examples. Since there are four directions, we will call this specific tile-sorting *square-sorting.* First, we will formally define tile-sorting.

Definition 4.1 (tile-sorting). *In a* tile-sorting $\Sigma = (\Psi, \mathcal{K}, \Xi)$ *the sorts are* $\Psi = \mathsf{Dir} \uplus \mathsf{ODir} \uplus \{\mathsf{a}\}$ *for some sets* Dir *and* ODir *of directions and opposites, the signature* $\mathcal{K}$ *is arbitrary with an arbitrary assignment of sorts to control arities, and the condition* Ξ *is as follows:*

- *a closed link has either*
  - *exactly one* d*-point*
  - *exactly one* d*-point and one* o*-point*
  - *arbitrarily many* a *points*
- *an open* d*-link has exactly one* d*-point*
- *an open* a*-link has arbitrarily many* a *points*

The intuition behind our sorting is that the directions are used to sort certain ports so the signature can specify rooms with *e.g.* north, south, east, and west entrance/exit ports which can each be linked unique to some other room in the obvious way. We allow edges to have one directed point so that we can represent rooms where there is no exit in that direction *e.g.* no exit to the north. The last sort, the undirected sort a, is used to represent pure links and so these points and names can be linked together arbitrarily but may not connect to directions.

The sorting condition is clearly satisfied by the identities. It is preserved by tensor product — the definition can informally be described as placing the two link graphs side-by-side with no cross-wiring and thus each open and closed link of the tensor product has the same points as it had in whatever factor it belonged to. We now show that composition preserves tile-sorting.

In what follows, we use the usual convention and let p_0 range over P_0, p_i over $P_i (i = 0, 1)$, x_i over $X_i, (i = 0, 1)$. We will let q range over points of a link graph, using subscripting when it is useful. When discussing two link graphs $\vec{A}$ with a common domain W, we usually let p_2 range over $P_2 \uplus W$, where P_2 is the set of ports common to $\vec{A}$. The same conventions apply to node sets V and edge sets E.

Proposition 4.2 (tile sorting is preserved by composition). *If $A_i : X_i \to X_{i+1}, (i = (0, 1)$ are two tile-sorted link graphs with disjoint nodes and edges then $A_1 \circ A_0$ is a tile-sorted link graph.*

Proof. We first look at the cases of idle edges and names.

If e_0 is idle in A_0 then e_0 is idle in $A_1 \circ A_0$. If e_1 is idle in A_1 then e_1 is idle in $A_1 \circ A_0$. Both of these cases satisfy Ξ.

If x_1 is idle in A_0 then it has sort a. x_1 disappears in the composition but since it is linked, in A_1, to a link connecting only points of sort a, this is of no consequence. If x_2 is idle in A_1 then it has sort a and it is also idle in $A_1 \circ A_0$. Ξ is satisfied.

We now consider the images of elements in the domain of *link*. Recall that

$$link = link_{A_1 \circ A_0} = (\mathrm{Id}_{E_0} \uplus link_1) \circ (link_0 \uplus \mathrm{Id}_{P_1}).$$

- Let $A_0(q_0^{\mathsf{d}}) = e_0$. Then $(A_1 \circ A_0)(q_0^{\mathsf{d}}) = e_0$. Since A_0 is tile-sorted, there are two subcases:
  - e_0 has one d point in A_0 *i.e.* q_0^{d}. Then e_0 has one d point, q_0^{d}, in $A_1 \circ A_0$.
  - e_0 has one other d point in A_0, q_0^{o}. Then $(A_1 \circ A_0)(q_0^{\mathsf{o}}) = e_0$ and e_0 has two points of opposite direction in *link*.
- Let $A_0(q_0^{\mathsf{d}}) = x_1^{\mathsf{d}}$. Then q_0^{d} is the only element in A_0 that maps to x_1^{d}. We have two subcases:
  - $A_1(x_1^{\mathsf{d}}) = x_2^{\mathsf{d}}$ and so is the only element in $\mathrm{dom}(link_1)$ to do so. Thus, q_0^{d} is the only element in the domain of *link* to map to x_2^{d}.
  - $A_1(x_1^{\mathsf{d}}) = e_1$. We have three subcases.
    - No other element in $\mathrm{dom}(link_1)$ maps to e_1 and so q_0^{d} is the only element in the domain of *link* to map to e_1.
    - One other port maps to e_1. $A_1(p_1^{\mathsf{o}}) = e_1$ and so p_1^{o} is the only other element in the domain of *link* to map to e_1.
    - One other name maps to e_1. $A_1(x_1^{\mathsf{o}}) = e_1$. As A_0 is tile-sorted, this name has a unique preimage q_0^{o} in $link_0$ and so q_0^{o} is the only other point in $\mathrm{dom}(link)$ to map to e_1.
- Let $A_1(p_1^{\mathsf{d}}) = e_1$. Then $(A_1 \circ A_0)(p_1^{\mathsf{d}}) = e_1$. Since A_1 is tile-sorted, there are two subcases:
  - e_1 has one d point in $link_1$, p_1^{d}, and thus *link*.
  - e_1 has another o point in A_1. There are two subcases:
    - $A_1(p_1^{\mathsf{o}}) = e_1$. The only element in the domain of $A_1 \circ A_0$ that maps to p_1^{o} is itself. Hence, e_1 has two points of opposite direction in *link*.

- * $A_1(x_1^{\mathsf{o}}) = e_1$. As A_0 is tile-sorted and x_1 does not have sort a, there is exactly one point of sort o in dom(*link*) which maps to x_1 and hence e_1.

- Let $A_1(p_1^{\mathsf{d}}) = x_2^{\mathsf{d}}$. Then $(A_1 \circ A_0)(p_1^{\mathsf{d}}) = x_2^{\mathsf{d}}$ and since A_1 is tile-sorted, no other element in the dom(*link*) maps to x_2^{d}.

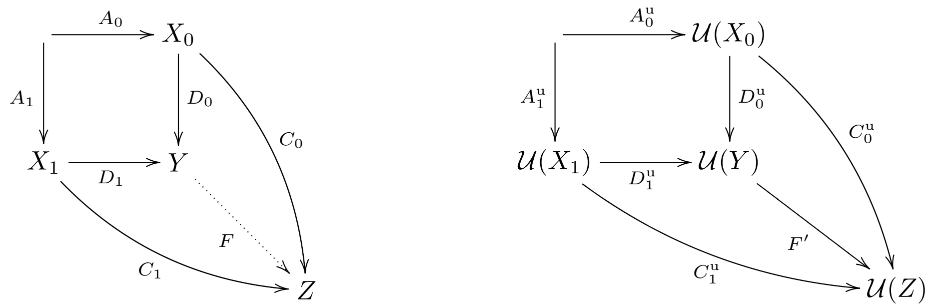
□

Proposition 8.8 [9] proves that a sorting presented in that work has a certain decomposition property. This property is then used to quite succintly prove that the forgetful functor from that sorting reflects pushouts. This decomposition property does not hold in tile sorting (a counter-example arises involving idle directed names) and so we do not have an immediate proof that our sorting functor reflects pushouts. However, we will show in the remainder of this section that a tile sorting functor both creates RPOs and reflect pushouts. The following lemma is invaluable to the proofs.

Lemma 4.3 (the Hammer lemma). *Let $\vec{D} : \vec{X} \to Y$ and $\vec{C} : \vec{X} \to Z$ be two bounds for $\vec{A} :\to \vec{X}$ in* $\text{'LIG}(\Sigma)$ *such that each $y \in Y$ is not idle in at least one of (B_0, B_1).*

If there exists a mediator $F' : \mathcal{U}(Y) \to \mathcal{U}(Z)$ such that the diagram on the right commutes, then there exists an $F : Y \to Z$ such that $\mathcal{U}(F) = F'$ and the diagram on the left commutes.

Further, if F' is unique as a mediating arrow between $\vec{D}$ and $\vec{C}$ then F is also a unique mediating arrow between $\vec{D}$ and $\vec{C}$.



The original proof is quite involved and so we defer it until Appendix C. The proof is followed by some discussion about relative pushouts.

We show that tile sorting has RPOs in a similar manner to that for binding bigraphs in [7] and kind place graphs. We first construct a relative bound based on a pure RPO, showing that it is sorted. We then prove that this RPO candidate is indeed the RPO.

In the following proofs, we will not annotate points with directions or a until we need to assume that they are (un)directed. We also write $\mathsf{ts}(G)$ to denote that a link graph G obeys tile sorting.

Construction 4.4 (building a tile-sorted RPO).

Proof. Let $\vec{A} : W \to \vec{X}$ have a bound $\vec{D} : \vec{X} \to Z$ in tile sorted link graphs. We wish to build a sorted RPO

$$(\vec{B} : \vec{X} \to \hat{X}, B : \hat{X} \to Z).$$

We start by building a pure RPO $(\vec{B'}, B')$ for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$ via [7, Construction 8.8]. From this we shall construct a sorted bound $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$, such that $(\vec{B}, B)^{\mathrm{u}} = (\vec{B'}, B')$. Then in the next proposition we shall show that it is a sorted RPO.

Let X' be the interface of this pure RPO. We will construct a sorted interface $\hat{X}$ with $\hat{X}^{\mathrm{u}} = X'$ by ascribing a sort to each $x \in X'$ during the proof. The first step is to show that B_0 and B_1 are sorted. Then proof then follows from Lemma 4.3.

We first examine the construction of X', B_0, and B_1 and show that for any point q_0 in B_0 such that $B_0(q_0) = x \in X'$, the sorting is obeyed. The proof for B_1 is similar.

Recall that the construction defines two sets

$$X'_i \stackrel{\mathrm{def}}{=} \{x \in X_i | D_i(x) \in E_3 \uplus Z\}$$

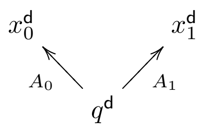
and defines the smallest equivalence relation $\cong$ on $X'_0 + X'_1$ such that $(0, x_0) \cong (1, x_1)$ whenever $A_0(p) = x_0$ and $A_1(p) = x_1$ for some point $p \in W \uplus P_2$. The RPO interface X' is then defined up to isomorphism as

$$X' \stackrel{\mathrm{def}}{=} (X'_0 + X'_1)/\cong .$$

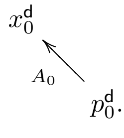
We subscript members of X'_i with i and let $\hat{x_i}$ denote the $\cong$-equivalence class of $(i, x_i), x_i \in X'_i$.

The proof that each $\hat{(}x_i) \in X'$ obeys the sorting is as follows.

- Let $x_0^{\mathsf{d}} \in X'_0$. Then, by $\mathsf{ts}(A_0)$, $A_0^{-1}(x_0^{\mathsf{d}}) = \{q^{\mathsf{d}}\}$. We have a case split.
  - Let $q^{\mathsf{d}} \in w \uplus P_2$. $(A_1)(q^{\mathsf{d}})$ must be a name in order for $\vec{B}^{\mathrm{u}}$ to be a bound for $\vec{A}^{\mathrm{u}}$ as $B_0(x_0^{\mathsf{d}})$ will be a name. Hence, $(A_1)(q^{\mathsf{d}}) = x_1^{\mathsf{d}}$ and so by $\mathsf{ts}(A_i)$,

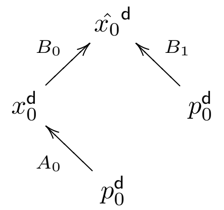


    depicts q^{d} as the unique preimage of x_0^{d} and x_1^{d} in A_0 and A_1 respectively. As $D_0(x_0^{\mathsf{d}})$ is a name or a fresh port and $\vec{D}$ is a bound for $\vec{A}$, $x_1^{\mathsf{d}} \in X'_1$. Thus, the $\cong$-equivalence class $\hat{x_0}^{\mathsf{d}}$ of x_0^{d} and x_1^{d} contains exactly those two elements and is of direction d. So, $B_i(x_i^{\mathsf{d}}) = \hat{x_0}^{\mathsf{d}}$ which satisfies the sorting. Also, there exists no $p' \in P_i - P_2$ such that $A_i(p) = x_i^{\mathsf{d}}$ as we have already identified the unique preimage of those names under A_i. By the construction, $\hat{x_0}^{\mathsf{d}}$ then has exactly one preimage in each of B_0 and B_1, *i.e.* x_0^{d} and x_1^{d} respectively, which is of the same direction and so $\mathsf{ts}(B_0)$ for this case.
  - Let $q^{\mathsf{d}} = p_0^{\mathsf{d}} \in P_0 - P_2$ be the unique preimage of x_0^{d} in A_0 and we can draw



By the construction, x_0^{d} is the only member of the $\cong$-equivalence class $\hat{x_0}^{\mathsf{d}}$ which is of sort d and thus is the only point of B_0 that maps to $\hat{x_0}^{\mathsf{d}}$, respecting the sorting.

p_0^{d} is also a point in B_1 and according to the construction, it is the only point in B_1 that maps to $\hat{x_0}^{\mathsf{d}}$. This information is depicted below.



Again, the sorting is obeyed and $\mathsf{ts}(B_0)$ for this case.

- Let $x_0^{\mathsf{a}} \in X_0'$.

  If $A_0^{-1}(x_0^{\mathsf{a}}) = \emptyset$ *i.e.* x_0^{a} is idle, then x_0^{a} forms its own equivalence class $\hat{x_0}^{\mathsf{a}}$ and is the only point in B_0 or B_1 to map to $\hat{x_0}^{\mathsf{a}}$. Thus, $\mathsf{ts}(B_0)$ and $\mathsf{ts}(B_1)$ holds for $\hat{x_0}^{\mathsf{a}}$.

  Otherwise, $A_0^{-1}(x_0^{\mathsf{a}}) = \{q_0, q_1, \ldots, q_{n-1}\}, n > 0$. By $\mathsf{ts}(A_0)$, all these points are undirected and so $A_0^{-1}(x_0^{\mathsf{a}}) = \{q_0^{\mathsf{a}}, q_1^{\mathsf{a}}, \ldots, q_{n-1}^{\mathsf{a}}$. By $\mathsf{ts}(A_1)$, each shared point in $\{q_0, q_1, \ldots, q_{n-1}\}$ will map to some x_1^{a} in X_1. As $D_0(x_0^{\mathsf{a}})$ is a name or a fresh port and $\vec{D}$ is a bound for $\vec{A}$, $x_1^{\mathsf{a}} \in X_1'$. We can repeat this process of getting inverse images of undirected names and then taking the images of the undirected points until we have two subsets $X_0'' \subset X_0'$ and $X_1'' \subset X_1'$, all of which contain undirected names. These elements form a $\cong$-equivalence class and they all map into undirected $\hat{x_0}^{\mathsf{a}}$ under B_0 and B_1 respectively, obeying the sorting.

  Let $p \in P_1 - P_2$. Then if $A_1(p) = {x'}_1^{\mathsf{a}} \in X_1''$, by $\mathsf{ts}(A_1)$, p is undirected *i.e.* p^{a}. By the construction, $B_0(p) = \hat{x_0}^{\mathsf{a}}$. Similarly for $p \in P_0 - P_2$. Hence, $\mathsf{ts}(B_i)$ for this case.

Thus, every outer name of B_0 and B_1 obeys the sorting.

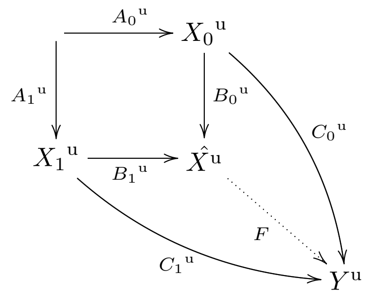
Next, we examine edges in B_i to ensure that they obeys the sorting. Since $B^{\mathrm{u}} \circ {B_i}^{\mathrm{u}} = {D_i}^{\mathrm{u}}$, by the definition of composition of link graphs, $B_i^{-1}(e_1) = D_i^{-1}(e_1)$ for any $e_1 \in E_1 - E_2$. But this covers all the edges of B_i. As $\mathsf{ts}(D_i)$, we immediately have that $\mathsf{ts}(B_i)$ for all edges in B_i. Thus, B_0 and B_1 are tile-sorted link graphs.

Now, we have proved that $\vec{B}$ and $\vec{D}$ are both sorted bounds for $\vec{A}$ such that each $x \in \hat{X}$ is not idle in at least one of (B_0, B_1). There also exists a mediator $\mathcal{U}(B) : X' = \mathcal{U}(\hat{X}) \to \mathcal{U}(Z)$ such $\mathcal{U}(B) \circ B_i^{\mathrm{u}} = D_i^{\mathrm{u}}$. Thus, by Lemma 4.3, there exists a sorted $B' : Y \to Z$ such that $\mathcal{U}(B') = \mathcal{U}(B)$. As $\mathcal{U}$ is faithful, B is this mediating arrow. Therefore, $(\vec{B}, B)$ is a candidate RPO for $\vec{A}$ to $\vec{D}$. □

The proof of the construction was a bit involved but the next two proofs follow easily.

Proposition 4.5 (tile sorted RPOs). *A tile-sorted RPO for $\vec{A}$ to $\vec{D}$ is provided by Construction 4.4.*

Proof. We must prove that this candidate $(\vec{B}, B)$ is the RPO. Given any other candidate $(\vec{C}, C)$, since $(\vec{B}, B)^{\mathrm{u}}$ is a pure RPO for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$, there is a unique mediating arrow F from $(\vec{B}, B)^{\mathrm{u}}$ to $(\vec{C}, C)^{\mathrm{u}}$. We represent this information in the diagram below.

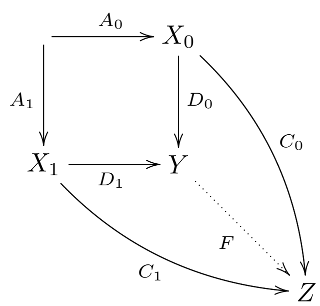


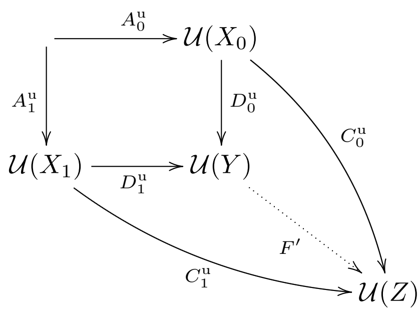
Now, $\vec{B}$ and $\vec{C}$ are both sorted bounds for $\vec{A}$ such that each $x \in \hat{X}$ is not idle in at least one of (B_0, B_1). There also exists a mediator $F : \mathcal{U}(\hat{X}) \to \mathcal{U}(Y)$ such $F \circ B_i^{\mathrm{u}} = C_i^{\mathrm{u}}$. Thus, by Lemma 4.3, there exists a sorted $\hat{F} : \hat{X} \to Y$ such that $\mathcal{U}(\hat{F}) = F$. As $\mathcal{U}$ is faithful and F is a unique mediating arrow, $\hat{F}$ is a unique mediating arrow from $(\vec{B}, B)$ to $(\vec{C}, C)$.

Thus, Construction 4.4 builds the RPO (up to isomorphism.) □

Thus, $\mathcal{U}$ creates RPOs. It can also easily be shown that $\mathcal{U}$ preserves RPOs.

Proposition 4.6 (tile sorting reflects pushouts). *Whenever $\vec{D}$ bounds $\vec{A}$ in* Lig(Σ) *and $\mathcal{U}(\vec{D})$ is a pushout for $\mathcal{U}(\vec{A})$, then $\vec{D}$ is a pushout for $\vec{A}$.*





Proof. To prove that $\vec{D}$ is a pushout for $\vec{A}$, we show that for any other bound $\vec{C}$ for $\vec{A}$, there exists a unique sorted mediator F such that $F \circ D_i = C_i$.

Assume that $\vec{C}$ is another bound for $\vec{A}$. Let F' be the unique (pure) mediating arrow from $\vec{D}^{\mathrm{u}}$ to $\vec{C}^{\mathrm{u}}$. Next, note that since $\mathcal{U}(\vec{D})$ is a pushout for $\mathcal{U}(\vec{A})$, then it is also an IPO for $\mathcal{U}(\vec{A})$. By the IPO construction, each $y \in \mathcal{U}(Y)$ is not idle in at least one of $(D_0^{\mathrm{u}}, D_1^{\mathrm{u}})$. Thus, each $y \in Y$ is not idle in at least one of (D_0, D_1).

We now invoke the Lemma 4.3, noting that F' is a unique mediating arrow. □

Note that the proofs in this section referred to a direction and the opposite of a direction. They did not specifically rely on the number of directions in Ψ, but merely on the fact that each direction had an opposite.

From the last two propositions, we immediately have the following corollary of Theorem 8.6 [9].

Corollary 4.7 (tile-sorting is useful). *Given a tile-sorting* Σ,

1. *Bisimilarity for the standard transition system* ST *over* $\acute{}\textsc{Lig}(\Sigma, \acute{}\mathcal{R})$ *is a congruence.*

2. *In addition, if* $\acute{}\mathcal{R}$ *is simple then the engaged transitions are adequate for* ST.

Finally, although we name these sortings *tile-sortings* as they were inspired by tile-based board games, we admit that they are very similar to a generalised form of the directed linear link graphs described by Leifer and Milner [9] once we forget about the undirected sort. Perhaps an intuitive description of tile-sortings would be as a hybrid of such a generalised directed linear sorting and pure link graphs.

4.2 Place sortings and kind bigraphs

The notion of link-sorting was subsequently applied to place graphs by Milner [14] and used to encode finite CCS in bigraphs. This *place-sorting* shares many properties with kind bigraphs (which were developed later) and so in the next few sections we will discuss the similarities and differences between both approaches as we see them.

4.2.1 Place sorting

A place-sorting is a sorting discipline which constrains the parent map of a bigraph, admitting only those bigraphs which satisfy the rules of the discipline. Place-sortings must satisfy certain conditions in order that they can be used to form an s-category. Some useful place-sortings have further properties such as that the RPO construction for place-sorted bigraphs should also be based on the RPO construction for pure bigraphs. There is also a forgetful functor associated with any sorting which forgets the sorts to yield pure bigraphs. This functor can be used to prove properties of the sorting based on the pure bigraphs[6].

The formal definitions (taken from [14]) are as follows, where Θ denotes a non-empty set of *sorts*, and θ ranges over Θ.

Definition 4.8 (place-sorted bigraphs [14]). *An interface with width* m *is* Θ-(place-)sorted *if it is enriched by ascribing a sort to each place* $i \in m$. *If* I *is place-sorted, its underlying unsorted interface is denoted by* $\mathcal{U}(I)$.

$\acute{}\textsc{Big}_{\mathrm{h}}(\mathcal{K}, \Theta)$ *denotes the s-category in which the objects are place-sorted interfaces, and each arrow* $G : I \to J$ *is a bigraph* $G : \mathcal{U}(I) \to \mathcal{U}(J)$. *The identities, and composition and tensor product are as in* $\acute{}\textsc{Big}_{\mathrm{h}}(\mathcal{K})$, *but with sorted interfaces.*

As with kind bigraphs and binding bigraphs, the interfaces are affected in the generalisation. Kind interfaces are similar to place-sorted interfaces in that a kind interface of width m contains a vector C_I of m subsets and a place-sorted

[6] This should be a familiar notion by now — in binding bigraphs and kind bigraphs, much of the pure theory is 'pulled back' along the respective forgetful functors.

interface of width m can be described as a vector (or sequence) of m singleton sets, each of which is a sort.

Definition 4.9 (place-sorting [14]). *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 $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\mathcal{K}, \Theta)$ is Σ-(place-)sorted *if it satisfies Φ. The Σ-sorted bigraphs form a sub-s-category of $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\mathcal{K}, \Theta)$ denoted by $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma)$. Further, if $\acute{\mathcal{R}}$ is a set of Σ-sorted reaction rules then $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma, \acute{\mathcal{R}})$ is a* Σ-sorted *Brs.*

Associated with a place-sorting is a forgetful functor

$$\mathcal{U} : \acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma) \to \acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\mathcal{K})$$

which discards sorts. Such a functor $\mathcal{U}$ is called a *sorting* functor, is surjective on objects, and is faithful.

4.2.2 'sort' versus 'kind'

The notion of 'sort' is generally separate to that of 'kind'. Sorts are members of a non-empty set Θ. In comparison, we think of the set of controls that one control may contain as its kind. However, we may blur this distinction. For a signature $\mathcal{K}$, let $\Theta = \mathcal{P}(\mathcal{K})$, the powerset of controls. We show in Section 4.2.5 that a kind signature can then be encoded as a specific place-sorting.

An important distinction is that though place-sorted interfaces contain sorting information, place-sorted bigraphs do not necessarily impose any constraints upon the structure of place graphs. On the other hand, kind bigraphs must obey the kind rules. A place-sorting may also impose constraints vastly different to the kind rules *e.g.* it may require that each root or node has at most one child. Such constraints do not employ sorts and also can not be expressed with kind place graphs.

Place-sorting is thus a more general concept than kindness — we show this in Section 4.2.5 — and there is no doubt that there are many different, useful place-sortings.

4.2.3 Homomorphic sorting

A homomorphic sorting has 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. The formal definition is:

Definition 4.10 (homomorphic sorting [14]). *In a* homomorphic *sorting $\Sigma = (\mathcal{K}, \Theta, \Phi)$ the condition Φ assigns a sort θ 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.*

In a homomorphic sorting, the assignment of a sort θ to each $K \in \mathcal{K}$ can also be described as a partition $f : \mathcal{K} \to \theta$ of $\mathcal{K}$ into disjoint subsets of controls with the same sort.

Leifer and Milner [9] identified two properties of a link-sorting which, if satisfied, provide sufficient structure for the transition theory of the link-sorting. Milner [14] then showed that these two properties, if satisfied for a place-sorting, similarly yield a transition theory for the place-sorting. The two properties are as follows:

Definition 4.11 (creating RPOs, reflecting pushouts [14]). *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 $\vec{D}$.*

$\mathcal{F}$ reflect 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 $\vec{A}$.*

Homomorphic sorting has these properties.

Proposition 4.12 (homomorphic sorting is well behaved [14]). *Every homomorphic sorting creates RPOs and reflects pushouts.*

The importance of these properties is that they can be used in conjunction with the following theorem of [14] (where a mono context is a context which is a monomorphism.) When the antecedents hold, the theorem can be used to show that 1) bisimilarity is a congruence and 2) that one may reduce some of the labelled transition systems (thus increasing efficiency) without affecting bisimilarity. We refer the reader to that text for the explanations of $\sim_{\mathrm{ST}}$, $\dot{\sim}_{\mathrm{ST}}$, PE and $\dot{}$PE.

Theorem 4.13 (useful place-sortings [14]). *In $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma, \acute{\mathcal{R}})$:*

1. *If Σ creates RPOs then $\sim_{\mathrm{ST}}$ is a congruence, and $\dot{\sim}_{\mathrm{ST}}$ is a congruence for mono contexts.*
2. *If in addition Σ reflects pushouts and $\acute{\mathcal{R}}$ is prime simple, then* PE *is adequate for* ST *and* $\dot{}$PE *is adequate for* $\dot{}$ST.

Before continuing, we will make a quick aside. The condition that a root must have the same sort as its children may initially seem to imply that the prime product operator is not well-defined. However, recall that the prime product is defined as

$$P_0|P_1 \stackrel{\text{def}}{=} \mathit{merge}_n \circ (P_0 \| P_1).$$

The parallel product, $P_0 \| P_1$, of two homomorphic-sorted bigraphs indeed respects the sorting. However, *merge* must also be sorted. merge_n has no nodes and maps n sites to a single root but, since it is sorted, it must be the case that the sort of the root is the same as the sort of all the sites. There is therefore a different merge_n for each sort θ. One could then write $\mathit{merge}_{n,\theta}$ as the bigraph with no nodes which maps n sites of sort θ to one root of sort θ. It is then clear that the prime product $P_0|P_1$ is defined iff the roots in both bigraphs all have the same sort and the underlying prime product is defined. This extra condition on prime product fits perfectly with the encoding of finite CCS in [14].

4.2.4 Homomorphic sortings as kind bigraphs

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 $kind(K) = \mathcal{K}_\theta$.

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 $C_r = \mathcal{K}_\theta$ for some sort θ. We therefore restrict ourselves to the full sub-s-category of kind bigraphs over $\mathcal{K}$ whose interfaces have this property. We will call this sub-s-category of $\mathcal{K}\text{BG}_\text{h}(\mathcal{K})$ the *homomorphic s-category* and denote it by $\mathcal{K}\text{BG}_\text{h}(\Sigma)$. $\mathcal{K}\text{BG}_\text{h}(\Sigma)$ is defined when $\mathcal{K}$ is a kind signature derived as above from a homomorphic sorting Σ.

As an example, we show how to describe the homomorphic sorting used for the encoding of finite CCS [14] as a kind signature. The encoding uses four controls. The nil and alt controls have sort p while the send and get controls have sort m. We define

$$\Theta_p = \{\text{nil}, \text{alt}\} \quad \text{and} \quad \Theta_m = \{\text{send}, \text{get}\}.$$

The $prnt$ map of the sorting is defined as $\{p \mapsto m, m \mapsto p\}$. We define $kind$ as follows:

$$\begin{aligned} kind(\text{nil}) = kind(\text{alt}) &= \{\text{send}, \text{get}\} \\ kind(\text{send}) = kind(\text{get}) &= \{\text{nil}, \text{alt}\}. \end{aligned}$$

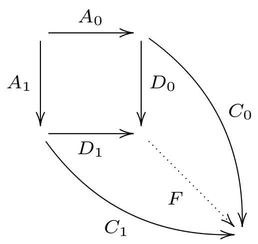
Interfaces in the related homomorphic s-category then have the form $I = \langle m, (C_0 \ldots C_{m-1}), X\rangle$ where each C_i, $i \in m$ is equal to either $\{\text{nil}, \text{alt}\}$ or $\{\text{send}, \text{get}\}$.

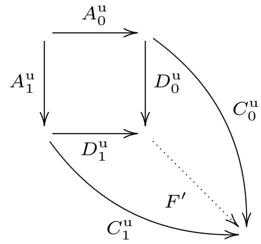
We will see below that the property of reflecting pushouts is important for a forgetful functor into pure bigraphs. In general, the forgetful functor from kind bigraphs to pure bigraphs does not reflect pushouts. However, the functor $\mathcal{U} : \mathcal{K}\text{BG}_\text{h}(\Sigma) \to \mathcal{B}\text{IG}_\text{h}(\mathcal{K})$ from a homomorphic s-category does, as we show below. This ensures that in our description of homomorphic sortings using kind signatures, we retain the important properties of RPO-creation[7] and pushout reflection. In the following proof, $\mathcal{U}$ is the forgetful functor $\mathcal{U} : \mathcal{K}\text{BG}_\text{h}(\Sigma) \to \mathcal{B}\text{IG}_\text{h}(\mathcal{K})$.

Proposition 4.14 ($\mathcal{U}$ reflects pushouts). *Whenever $\vec{D}$ bounds $\vec{A}$ in* $\mathcal{K}\text{BG}_\text{h}(\Sigma)$

[7] We omit the proof for RPO-creation here. It should fall out from the RPO construction for pure bigraphs (as stated in [14]). All one would need to do to define the sorted interface for the RPO is to identify the sort of the nodes under a root – the root then has that sort. A case analysis similar to all those in this report should then yield the proof.

and $\mathcal{U}(\vec{D})$ *is a pushout for* $\mathcal{U}(\vec{A})$*, then* $\vec{D}$ *is a pushout for* $\vec{A}$*.*





Proof. To prove that $\vec{D}$ is a pushout for $\vec{A}$, we first show that for any other bound $\vec{C}$ for $\vec{A}$, there exists a sorted mediator F such that $F \circ D_i = C_i$ and then prove the uniqueness of F.

Assume that $\vec{C}$ is another bound for $\vec{A}$. Let F' be the unique (pure) mediating arrow from $\vec{D}^{\mathrm{u}}$ to $\vec{C}^{\mathrm{u}}$. Let F have the same *prnt* map as F'. We will break the proof into cases on the parent map of F and show in each case that homomorphic sorting is obeyed. In the following, v and v' are nodes, t is a root of F, r is a site in F and s is a site in D_i.

- Let $v = F(v')$. Then $v = C_i(v')$ and since C_i is sorted, F obeys the sorting.
- Let $t = F(v)$. Then $t = C_i(v)$ and since C_i is sorted, F obeys the sorting.
- Let $v = F(r)$ where r has sort θ and v has sort θ'. We must show that $\theta' = prnt(\theta)$. There are two cases.
  - Let $r = D_i(s)$. Since D_i is sorted, s has sort θ. Now, C_i is sorted and $v = C_i(s)$. Therefore, the sort of v is $prnt(\theta)$ *i.e.* $\theta' = prnt(\theta)$.
  - Let $r = D_i(v')$. Since D_i is sorted, v' has sort θ. Now, C_i is sorted and $v = C_i(v')$. Therefore, the sort of v is $prnt(\theta)$ *i.e.* $\theta' = prnt(\theta)$.
- Let $t = F(r)$ where r has sort θ and t has sort θ'. We must show that $\theta' = \theta$. There are two cases.
  - Let $r = D_i(s)$. Since D_i is sorted, s has sort θ. Now, C_i is sorted and $t = C_i(s)$. Therefore, the sort of t is equal to θ *i.e.* $\theta' = \theta$.
  - Let $r = D_i(v)$. Since D_i is sorted, v has sort θ. Now, C_i is sorted and $t = C_i(v)$. Therefore, the sort of t is equal to θ *i.e.* $\theta' = \theta$.

We have now shown that F is sorted and clearly $F \circ D_i = C_i$. The uniqueness of F follows from the fact that $\mathcal{U}$ is faithful and that F' is a unique mediator. □

The reader may note that the notion of fitting is not that important for homomorphic-sorted bigraphs. In fact, fitness is now a redundant notion. Since we have reduced the set of interfaces to sorted interfaces, a root has exactly one sort ascribed to it. Since we are assuming all bigraphs are hard, the root also contains at least a site of the same sort or a node of the same sort. In a sense, every bigraph is then *fitting with respect to sorts* in that a root with sort θ need only be a parent of one place with sort θ — one child of sort θ suffices, regardless of how many controls are in $\mathcal{K}_\theta$.

We will finish this section by discussing two alternatives to homomorphic sorting which are very similar in their set of nesting rules but will undoubtedly yield different Brss.

First alternative One may visualise the *prnt* function of a homomorphic sorting as a directed graph with one vertex per sort and exactly one arc leaving each vertex. Arcs represent '(source) can be contained in (target)'.One may also think of the *kind* function in a kind signature as a simple, directed graph with one vertex per control and between one and $|\mathcal{K}|$ arcs leaving each vertex, with an arc from K to K' whenever $K' \in kind(K)$ *i.e.* arcs represent '(source) can contain (target)'. If the arcs are reversed, we have a graph where the arcs represent 'can be contained in' and the graph is comparable with the 'homomorpic graph'.

Given a kind signature which represents a homomorphic sorting (aside from the extra condition on the roots), we may also consider slightly reducing the set of controls that some control may contain *i.e.* removing some arcs from the 'kind graph' described above. We will not of course have a strictly homomorphic sorting anymore but we will have something similar. We can then restrict our attention to the fitting sub-s-category over this signature (for reasons given in the following sections.)

Second alternative Another alternative way to 'almost' describe homomorphic sortings as kind bigraphs is as follows. Given the description of a homomorphic sorting as a kind signature, instead of restricting our attention to the homomorphic sub-s-category we could instead use the fitting sub-s-category. What we end up with now is something similar to a homomorphic sorting, where the *prnt* map of the sorting is respected but where roots may be able to contain controls of different sorts and where sites may be able to contain subsets of $\mathcal{K}_\theta$, when the parent of the site is a node of sort $prnt(\theta)$. What makes this alternative interesting is that it admits a prime product as defined in the pure theory, something which is not true in homomorphic sortings.

We can not claim that these alternatives are useful without any applications! They may make interesting asides though — we will show below that the forgetful functor from the sub-s-category of fitting bigraphs over a kind signature still creates RPOs and reflects pushouts. The transition theories for the resulting alterative Brss may well be different to that of the original homomorphic sorting, but they will still have some nice properties.

4.2.5 Kind bigraphs as place sortings

We can describe many place sortings by "assigning a sort $\theta \in \Theta$ to every control, and then imposing constraints upon a bigraph in terms of the sorts thereby associated with nodes" [14]. Examples of such place-sortings include homomorphic sortings. Kind bigraphs and fitting bigraphs are also examples of such place-sortings as will now be shown.

In the definitions below, the signature $\mathcal{K}$ is a pure signature, not a kind signature. The sort of a root r will be denoted by $sort(r)$ and the sort of the control of a node v will be denoted by $sort(v)$.

Definition 4.15 (kind sorting). *In a* kind *sorting* $\Sigma_K = (\mathcal{K}, \mathcal{P}(\mathcal{K}), \Phi)$ *the condition* Φ *assigns a sort* $\theta \in \mathcal{P}(\mathcal{K})$ *to each control in* $\mathcal{K}$*. Then a bigraph* G *is admissible iff it satisfies*

- *if* $r = G(v)$ *then* $ctrl(v) \in sort(r)$
- *if* $r = G(s)$ *then* $sort(s) \subseteq sort(r)$
- *if* $v = G(v')$ *then* $ctrl(v') \in sort(v)$
- *if* $v = G(s)$ *then* $sort(s) \subseteq sort(v)$
- *if* $sort(v) = \emptyset$*,* v *has no children*

where r *is a root,* s *a site,* v *and* v' *are nodes.*

Given this alternate formulation of kindness, we can simplify its definition[8].

Definition 4.16 (kind sorting (revised)). *In a* kind *sorting* $\Sigma_K = (\mathcal{K}, \mathcal{P}(\mathcal{K}), \Phi)$ *the condition* Φ *assigns a sort* $\theta \in \mathcal{P}(\mathcal{K})$ *to each control in* $\mathcal{K}$*. Then a bigraph* G *is admissible iff it satisfies:*

K1 *if* $p = G(v)$ *then* $ctrl(v) \in sort(p)$

K2 *if* $p = G(s)$ *then* $sort(s) \subseteq sort(p)$

K3 *if* $sort(v) = \emptyset$*,* v *has no children*

where p *is a root or node,* s *a site, and* v *is a node.*

We have shown in previous sections that identities respect these rules and that the rules are preserved by composition and tensor product and so our encoding of kindness as a place sorting is well-defined. We can then state that kind bigraphs are a special case of place-sorted bigraphs. We write $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma_K)$ to denote an s-category which is sorted over a kind sorting Σ_K. Kind bigraphs also have RPOs and the sorting functor, the forgetful, faithful functor from kind bigraphs to pure bigraphs, is akin to the functor $\mathcal{U}$ described in Section 3.

We would also like to apply the transition theory of pure bigraphs to kind bigraphs. We have shown that the forgetful functor $\mathcal{U}$ associated with kind bigraphs creates RPOs. We immediately infer the following as a corollary of Theorem 10.6.1 [14], where $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma_K, \acute{\mathcal{R}})$ is a kind-sorted Brs of hard bigraphs (a mono context is a context which is a monomorphism.)

Corollary 4.17. *In* $\acute{\mathrm{B}}\mathrm{IG}_{\mathrm{h}}(\Sigma_K, \acute{\mathcal{R}})$*,* $\sim_{\mathrm{ST}}$ *is a congruence, and* $\dot{\sim}_{\mathrm{ST}}$ *is a congruence for mono contexts.*

We have not shown that $\mathcal{U}$ reflects pushouts and in fact, $\mathcal{U}$ *does not* reflect pushouts. A counter-example is easily found where $\vec{D}$ bounds $\vec{A}$ but the outer interface of $\vec{D}$ is not fitting for $\vec{D}$. We therefore can not directly infer the very useful result of Theorem 10.6.2 [14] which would allows us to reduce the size of the standard Lts without affecting bisimilarity.

This fact may turn out to be crucial in the dynamic theory of kind bigraphs but we will try to address that problem in further work. For now, note that kind

[8] The author wishes to acknowledge Søren Debois for his suggested simplification of kindness in a private communication. His suggestion has similarities with the following definition.

bigraphs do not reflect pushouts unless the bound in question is fitting. Therefore, an avenue of escape exists if we restrict our attention to the s-categories of fitting bigraphs over a kind signature. We show in the next section that this s-category both creates RPOs and reflects pushouts. The downside is that using fitting bigraphs limits the reaction rules we can define, as both the reactum and redex must both be fitting while having the same outer interface. Section 4.4 contains some discussion on how we may overcome this limitation to some extent.

4.2.6 Fitting bigraphs as well-behaved place sortings

The description of fitting bigraphs as a place-sorting is similar to that of kind bigraphs. In the definition below, the signature $\mathcal{K}$ is again a pure signature, not a kind signature.

Definition 4.18 (fitting sorting). *In a* fitting *sorting* $\Sigma_F = (\mathcal{K}, \mathcal{P}(\mathcal{K}), \Phi)$ *the condition* Φ *assigns a sort* $\theta \in \mathcal{P}(\mathcal{K})$ *to each control in* $\mathcal{K}$*. Then a bigraph is admissible iff it satisfies:*

F1 *if* $p = G(v)$ *then* $ctrl(v) \in sort(p)$

F2 *if* $p = G(s)$ *then* $sort(s) \subseteq sort(p)$

F3 *if* $sort(v) = \emptyset$*,* v *has no children*

where p *is a root or node,* s *a site,* v *is a node, and the sort of every root is the least subset of* $\mathcal{K}$ *such that* **F1** *and* **F2** *are satisfied in the bigraph.*

We have shown in previous sections that identities respect these rules and that the rules are preserved by composition and tensor product and so our encoding of fitness as a place sorting is well-defined. We write $\text{'B}\text{IG}_{\text{h}}(\Sigma_{\text{F}})$ to denote an s-category which is sorted over a fitting sorting Σ_{K} and $\text{'B}\text{IG}_{\text{h}}(\Sigma_{\text{K}}, \text{'}\mathcal{R})$ to denote a fitting-sorted Brs of hard bigraphs. However, for the moment we will continue to describe fitting bigraphs as special cases of kind bigraphs and not as place-sortings as it clarifies our proofs.

$\text{'F}\text{KB}(\mathcal{K})$ is a sub-s-category of $\text{'K}\text{BG}(\mathcal{K})$ and so there is an inclusion functor $\mathcal{F} : \text{'F}\text{KB}(\mathcal{K}) \to \text{'K}\text{BG}(\mathcal{K})$. In the rest of this section, we will refer to a forgetful functor from the s-category $\text{'F}\text{KB}(\mathcal{K})$ to $\text{'B}\text{IG}(\mathcal{K})$. This forgetful functor is the composition $\mathcal{U} \circ \mathcal{F}$, or $\mathcal{UF}$. We will abuse notation and denote the pure, underlying interface of an interface in $\text{'F}\text{KB}(\mathcal{K})$ as I^{u}, and similarly write G^{u} as the pure underlying bigraph of a fitting bigraph G.

The following proposition states that when a kind RPO is constructed for a bounded pair, where all four bigraphs are fitting, then the RPO triple consists of three fitting bigraphs. The remainder of the proposition then sets up the corollary that $\mathcal{UF}$ creates RPOs, given that $\mathcal{U}$ creates RPOs and $\mathcal{F}$ is faithful.

Proposition 4.19 (fitting RPOs). *Given a bound* $\vec{D}$ *for* $\vec{A}$ *in* $\text{'K}\text{BG}$ *where all four kind bigraphs are fitting, let* $(\vec{B}, B)$ *be the kind RPO triple for* $\vec{A}$ *relative to* $\vec{D}$*. Then* B_0, B_1 *and* B *are fitting bigraphs.*

Further, for any RPO candidate $(\vec{B}, B)$ *where* $\vec{C}$ *and* C *are fitting bigraphs, the unique kind mediator* $\hat{F}$ *from* $(\vec{B}, B)$ *to* $(\vec{C}, C)$ *is fitting.*

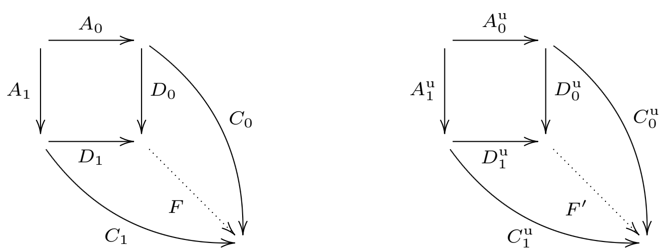
Proof. By Construction 2.18, the outer interface of $\vec{B}$ is fitting for the pair. Also, $\vec{B}$ is a bound for $\vec{A}$. By Lemma 3.4.4, B_0 and B_1 are fitting. As $D = B \circ B_0$ and both D and B_0 are fitting, by Lemma 3.4.2 above, B is fitting.

For the second half of the proof, note that by definition $\hat{F} \circ B_i = C_i$. B_i and C_i are fitting and so by Lemma 3.4.2 again, $\hat{F}$ is fitting. □

Corollary 4.20 (fitting bigraphs have RPOs). *The functor $\mathcal{UF}$ creates RPOs.* ´BIG *has RPOs and so* ´FKB *also has RPOs.*

It turns out that the next proposition, while not true for a forgetful functor from an s-category of kind bigraphs in general, is true for fitting bigraphs.

Proposition 4.21 ($\mathcal{UF}$ reflects pushouts). *Whenever $\vec{D}$ bounds $\vec{A}$ in* ´FKB *and $\mathcal{UF}(\vec{D})$ is a pushout for $\mathcal{UF}(\vec{A})$, then $\vec{D}$ is a pushout for $\vec{A}$.*



Proof. To prove that $\vec{D}$ is a pushout for $\vec{A}$, we show that for any other bound $\vec{C}$ for $\vec{A}$, there is a unique mediating arrow F such that $F \circ D_i = C_i$.

C_i and D_i have a common domain and both are fitting. As $\vec{C}^{u}$ is a bound for $\vec{A}^{u}$ and $\vec{D}^{u}$ is the pushout, there exists a unique F' such that $F' \circ D_i^{u} = C_i^{u}$. By Lemma 3.4.4, there exists a unique F in ´FKB such that $F \circ D_i = C_i$. Thus, $\vec{D}$ is a pushout for $\vec{A}$ and so $\mathcal{UF}$ reflects pushouts. □

We have shown that fitting sortings are well-behaved in that they create RPOs and reflect pushouts. We now present the following as a corollary of Theorem 10.6 [14].

Corollary 4.22 (fitting bigraphs are well behaved). *In* $´\mathrm{BIG}_{\mathrm{h}}(\Sigma_F, ´\mathcal{R})$*:*

1. *$\sim_{\mathrm{ST}}$ is a congruence, and $\dot{\sim}_{\mathrm{ST}}$ is a congruence for mono contexts.*
2. *If $´\mathcal{R}$ is prime simple, then* PE *is adequate for* ST *and* ˙PE *is adequate for* ˙ST.

This is an encouraging result.

That '$´\mathcal{R}$ is prime simple' means that each redex of the Brs is prime and simple. That means specifically that in each redex,

– the outer interface is of width 1,

– the parent of any site is a node,

– no two sites have the same node as parent *i.e.* they are not siblings,

– there are no inner names, no edges, and no idle names.

Knowing that we have a satisfactory transition theory, we now give some examples of different fitting Brss.

4.3 Examples of fitting Brss

It has been shown that the notion of place sorting is far more general than that of kind bigraphs. However, in this report we have succesfully identified a particular useful place-sorting – fitting bigraphs – which we believe may have interesting applications. This section consists of some examples of the types of rules which fitting bigraphs can encompass. We will perhaps labour the point that fitting bigraphs have some added expressiveness over pure bigraphs but this is merely for demonstrative purposes. Indeed, there are some pure Brss which can not be described as fitting Brss. We return to this problem in 4.4 with some conjectured solutions but the point should be made.

Notation When writing the terms representing redexes and reactums of the reaction rules in this section, we will omit some of the information regarding linkage. This is done purely to present the examples as simply as possible since our emphasis in these rules is on the kind place graph structures. As such, many of the terms in the figures will appear to be written incorrectly with respect to the links but the intended term should be clear from the diagram.

We can extend the term language of bigraphs to describe the set of controls that sites may contain. This will make it possible to describe reaction rules using only the term language. In the pure theory, a site is represented in a term by a square, $\square$, which may be indexed if necessary. For fitting bigraphs, sites in terms are denoted in one of three ways. Let $\mathcal{K} = \{\mathbf{room}, \mathbf{prsn}, \mathbf{pc}\}$. We will write $\boxed{\mathbf{room\,prsn}}$ to describe a site which can contain controls **room** and **prsn**. We may also write $\boxed{\neg\mathbf{pc}}$, or 'not **pc**' to describe the same site. If a site can contain all elements of $\mathcal{K}$, we denote it simply by $\square$. To avoid any ambiguity, we will index sites when necessary. This notation may prove unwieldy for some applications but will be useful when $\mathcal{K}$ contains few elements.

Rule convention In a fitting Brs, the set of controls that a site of a redex can contain is specified by the inner interface of the redex. This will usually lead us to define a *set* of parametric reaction rules for each 'pure' rule. For example, say we have a reaction rule where the redex has exactly one site which should be able to contain any control in $\mathcal{K}$. As the parameters of any parametric reaction rule are fitting, we will need to define a set of $2^{|\mathcal{K}|}$ fitting reaction rules – one for each subset of $\mathcal{K}$ – to capture the intended rule. Listing all such rules may be tedious and so we now introduce a convention where only one parametric reaction rule need be defined for each desired dynamic behaviour. We will typically use this convention in the examples below.

Given a parametric reaction rule (R, R'), we may generate another reaction rule $(R \circ \downarrow\vec{V}_I, R'')$ where the inner interface of the redex R is deflated. For our convention, we assume that any specified rule (R, R') generates a 'deflated rule' $(R \circ \downarrow\vec{V}_I, R'')$ for any inner deflation of the redex. We call this the 'deflated rule convention'.

4.3.1 Prioritised reaction-firing

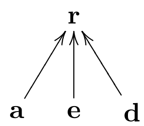
We now return to one of our previous examples to explain the added expressiveness of kind bigraphs over pure bigraphs. We wish to describe the rules of a simple game with a nesting structure, dynamics and constraints on how the game changes. The entities in the game are agents, enemies, and dots and the game takes place in a series of square rooms connected together. We therefore assume without further comment that we are employing our square-sorting. Agents move around these rooms collecting dots and enemies move around killing agents. The constraints of the game are as follows: Agents can not leave a room containing a dot without consuming it and can not leave a room containing an enemy. Enemies can not leave a room with agents in it and can not kill an agent while the agent is near to a dot. We will present reaction rules for this game and then show how using kindness allows us to capture the constraints.

We assume the deflated rule convention for the following *i.e.* any explicitly defined rule (R, R') generates a set of rules where each site i in the redex (resp. reactum) of an element of the set can contain a subset of the controls that the site i can contain in R (resp. R'.)

We capture the nesting structure of the game with a kind signature. The set of controls, $\mathcal{K}$, is $\{\mathbf{r}, \mathbf{a}, \mathbf{e}, \mathbf{d}\}$, representing rooms, agents, enemies and dots respectively. $ar(\mathbf{r}) = 4$ and $ar(\mathbf{a}) = ar(\mathbf{e}) = ar(\mathbf{d}) = 0$. The nesting structure, *kind*, is defined by

$$\mathbf{r} \mapsto \{\mathbf{a}, \mathbf{e}, \mathbf{d}\}, \quad \mathbf{a} \mapsto \emptyset, \quad \mathbf{e} \mapsto \emptyset, \quad \mathbf{d} \mapsto \emptyset.$$

The nesting structure can also be represented as a graph where the arcs represent 'can be contained in':



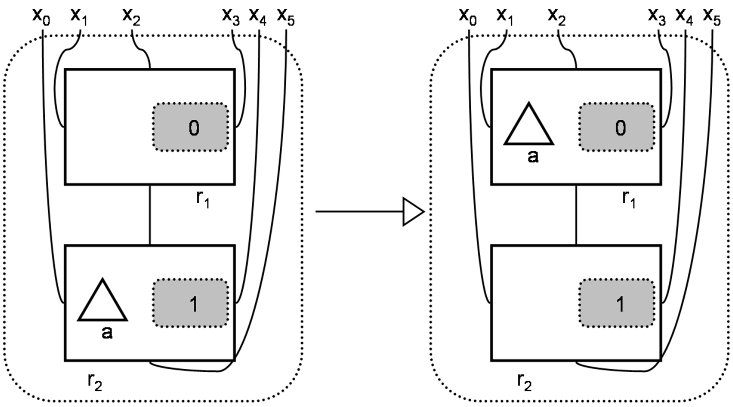
The rules for the game are encoded as reaction rules. The 'move north' rules are shown in Figures 7 and 8. The rules for moving in other directions are similar. For the agent movement rule, we have specified that the room the agent starts in has no dots or enemies. For the enemy movement rule, we stated that the starting room has no agents. Figure 9 depicts the rule where an enemy kills an agent in the same room where the room does not contain any dots and Figure 10 shows the rule where an agent collects a nearby dot. $\acute{\mathcal{R}}$ is not prime simple in this case as we have edges in some of the rules. In these rules, the sites state what controls are guaranteed not to be the various rooms. The constraints of the game are thus modelled by the *guaranteed absence* of certain entities.

An interesting aspect of this reactive system is that the dynamic behaviour of an agent can be described as:

```
if dot in room, collect dot
else if ghost in room, die
else move.
```

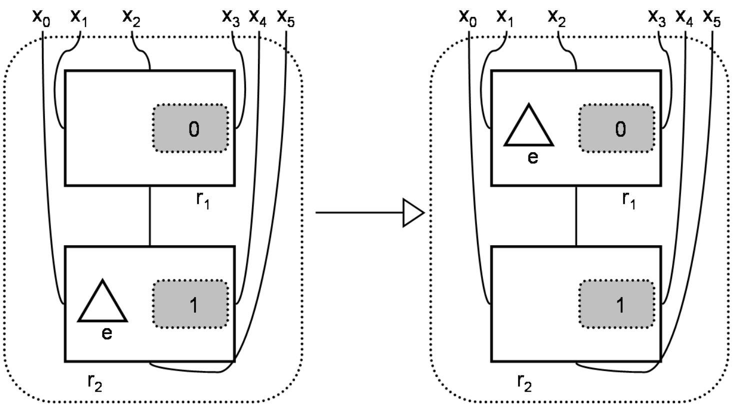
In short, the restrictions we have placed on what the sites can contain have prioritised the firing of reactive rules. The reason we get this clean separation (if, else if, else) is actually by using logical propositions in the construction of the reaction rules as we will now explain.

By our convention, each explicit reaction rule gives rise to a set of reaction



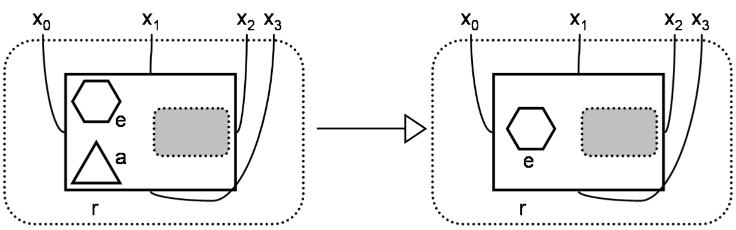
$$/l(\mathbf{r}_l\Box|\mathbf{r}_l(\mathbf{a}|\boxed{\neg\mathbf{e}\neg\mathbf{d}})) \longrightarrow\!\!\!\triangleright\ /l(\mathbf{r}_l(\mathbf{a}|\Box)|\mathbf{r}_l\,\boxed{\neg\mathbf{e}\neg\mathbf{d}})$$

Figure 7: An agent moves between two rooms



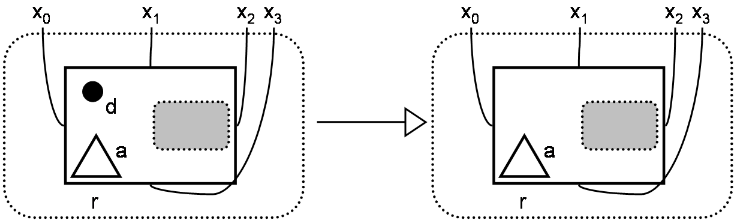
$$/l(\mathbf{r}_l\Box|\mathbf{r}_l(\mathbf{e}|\boxed{\neg\mathbf{a}})) \longrightarrow\!\!\!\triangleright\ /l(\mathbf{r}_l(\mathbf{e}|\Box)|\mathbf{r}_l\,\boxed{\neg\mathbf{a}})$$

Figure 8: An enemy moves between two rooms



$$\mathbf{r}(\mathbf{a}|\mathbf{e}|\boxed{\neg\mathbf{d}}) \longrightarrow\!\!\!\triangleright\ \mathbf{r}(\mathbf{e}|\boxed{\neg\mathbf{d}})$$

Figure 9: An enemy eliminates an agent



$$\mathbf{r}(\mathbf{a}|\mathbf{d}|\Box) \longrightarrow \mathbf{r}(\mathbf{a}|\Box)$$

Figure 10: An agent collects a dot

rules[9]. The facts we may state about a set of rules based on an explicit rule are i) which controls occur under some place of the redex and reactum and ii) which controls will not, for any element in the set of rules, occur under some parent of a site through composition. In the following, we will represent the proposition that some control K occurs under some place of the redex simply by K and the proposition that some control K will never occur under some specific parent of a site as $\neg K$. This representation will suffice for our purposes here – a better system would refine propositions with respect to the nesting structure of the redex *i.e.* $\neg_v K$ might mean that a node of control K will never occur under node v in the redex.

The propositions regarding agent behaviour for our game rules will talk about which controls occur or never occur in the room that the agent is in. The propositions are as follows: Let **a** mean 'there is at least one agent in the room' and the proposition $\neg\mathbf{a}$ mean 'there are no agents in the room'. Define propositions **d** and **e** and their negations similarly. The firing conditions for the rules for an agent moving, an agent dying and an agent collecting dots can then be represented as conjunctions of these propositions. For example, in the redex of the 'collect' rule, there is definitely one dot and one agent in the agent's room. The firing condition is then $\mathbf{a} \wedge \mathbf{d}$. In the redex of the 'move' rule, there is definitely one agent and no dots or enemies in the agent's room. The firing condition is then $\mathbf{a} \wedge \neg\mathbf{d} \wedge \neg\mathbf{e}$. The 'collect', 'die' and 'move' rules have been designed so that their firing conditions are mutually exclusive. This fact is shown in Figure 11 where we have omitted the 'False' entries for the last three columns. The conjunction of the three firing conditions is false – there is no overlap between the rules.

Given this example, we claim that kind bigraphs can enforce prioritised reaction-firing based on logical propositions on the nesting structure of redexes. The remainder of this section will demonstrate other possible rules which take advantage of this notion. We do not claim that the following models in themselves are theoretically interesting but we include them as examples of the use of fitting Brss.

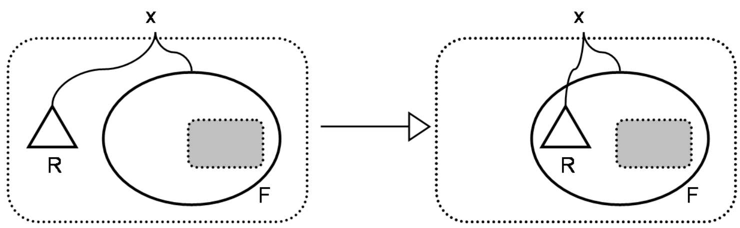
[9]From the pure theory, these rules also give rise to ground reaction rules and the set of all rules is closed under support equivalence.

a	**d**	**e**	**a** ∧ **d** (collect)	**a** ∧ ¬**d** ∧ **e** (die)	**a** ∧ ¬**d** ∧ ¬**e** (move)
T	T	T	T		
T	T	F	T		
T	F	T		T	
T	F	F			T
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Figure 11: Truth table representing mutually exclusive rules

4.3.2 Multiple readers

Let $\mathcal{K} = \{\mathbf{F}, \mathbf{R}, \mathbf{W}\}$, where the elements represent files, readers and writers respectively. Let readers and writers be atomic controls and let files be able to contain readers and writers. A pure reaction rule based on Figure 12 represents a reader being granted access to file, represented by moving the reader inside the file. Using kind bigraphs, we can specify the interfaces of the rule as R, R' :



$$\mathbf{R}_x|\mathbf{F}_x \boxed{\mathbf{R}} \rightarrowtriangle \mathbf{F}_x(\mathbf{R}_x|\boxed{\mathbf{R}})$$

Figure 12: Multiple readers accessing a file

$\langle 1, (\{\mathbf{R}\}), \emptyset\rangle \rightarrow \langle 1, (\{\mathbf{W}, \mathbf{R}\}), \{x\}\rangle$. The rule now states that a reader may gain access to a file if the reader has a link to the file and only readers (not writers) have access to the file.

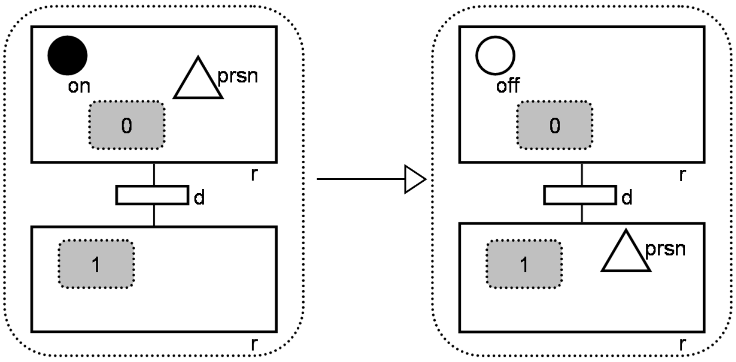
This rule is indeed prime simple but we cannot define it in a fitting Brs as the outer interface of R is not fitting for R'. We suggest using a kind Brs with fitting redexes for such a case or better still, a strong kind fitting Brs (to be discussed later). We discuss such cases of non-fitting rules in Section 4.4.

4.3.3 Smart buildings

This example was inspired by conversations with Simon Dobson, University College Dublin and by the DELCA example from the Bigraphical Programming Languages group page [5]. The structure of the system is taken from the DELCA example.

Let $\mathcal{K} = \{\mathbf{r}, \mathbf{d}, \mathbf{prsn}, \mathbf{g}, \mathbf{pc}, \mathbf{on}, \mathbf{off}\}$, where the elements represent rooms, doors, people, Ghosts, computers, lit lights, and unlit lights respectively. Let people, lights and doors be atomic controls and let rooms be able to contain people, Ghosts, computers, and lights. A pure reaction rule based on Figure

12 represents a person moving from one room to another room and the light in the first room turns off on exit. Using kind bigraphs, we can specify the



$$/l/r(\mathbf{r}_l(\mathbf{prsn}|\mathbf{on}|\,\boxed{\mathbf{g\;pc}}\,)|\mathbf{d}_{lr}|\mathbf{r}_r\square) \longrightarrow\!\!\triangleright\; /l/r(\mathbf{r}_l(\mathbf{off}|\,\boxed{\mathbf{g\;pc}}\,)|\mathbf{d}_{lr}|\mathbf{r}_r(\mathbf{prsn}|\square))$$

Figure 13: Lights that turn off when the last person exits the room

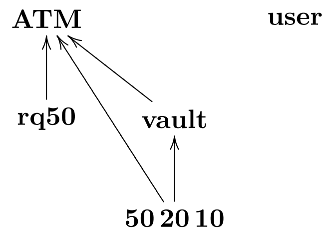
interfaces of the rule as $R, R' : \langle 2, (\{\mathbf{g}, \mathbf{pc}\}S), \emptyset\rangle \rightarrow \langle 2, (\{\mathbf{g}, \mathbf{pc}\}S), \{x\}\rangle$, where $S = \{\mathbf{prsn}, \mathbf{g}, \mathbf{pc}, \mathbf{on}, \mathbf{off}\}$. Again, we assume the deflated rule convention. The rule now states that if the last person in a room moves from the room then the light switches off — a behaviour which is perhaps desirable.

This behaviour can seemingly also be captured by two pure rules, one for people moving between rooms and another for turning the light off in an empty room. There are two problems with this model though. First, say rooms could contain any number of items. Then a separate rule would be needed for each unique instance of a room and its contents. This is perhaps a minor matter. Second, we may wish to represent such behaviour as an atomic action. In a non-deterministic system, there can be no guarantee as to how many reactions may take place between the last person leaving a room and the light turning off. Indeed, it is possible that a person may leave the room empty, perform some tasks, and return to the room and the light may never have switched off.

Finally, note that the redex of the rule is not prime simple as we again use edges.

4.3.4 Automatic teller machine

This example will show how kind bigraphs are sometimes able to refine rules by using an abstract model of a user's interaction with an automatic teller machine (ATM). Again, the example is purely for motivation. Let the controls **user**, **ATM**, **vault** represent a user, an ATM and an ATM's vault respectively. Let the controls **rq50**, **50**, **20**, **10** represent an internal ATM request for fifty dollars, a \$50 bank note, a \$20 bank note and a \$10 bank note respectively. Let $\mathcal{K}$ be a signature containing all of these controls and where the containment structure is:



We can imagine a scenario where a user has performed some interactions with the ATM. The ATM has an internal request for $50 to be made which is attached to some $50 note. The ATM then dispenses the note. This interaction is captured in Figure 14 where we assume we may combine binding bigraphs [7] with kind bigraphs (as such, this rule is not formally defined at the time of writing.)

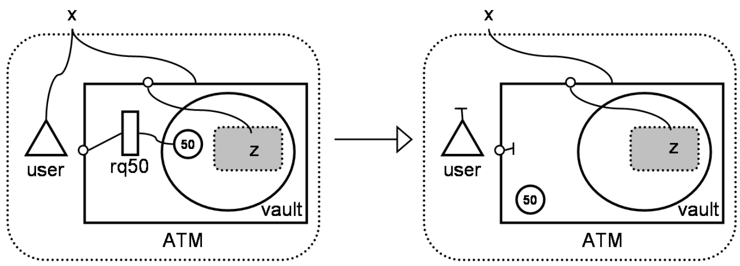
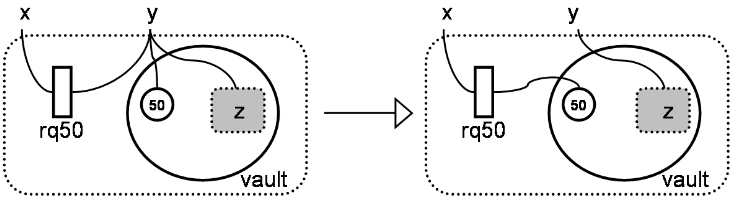


Figure 14: An ATM returns $50

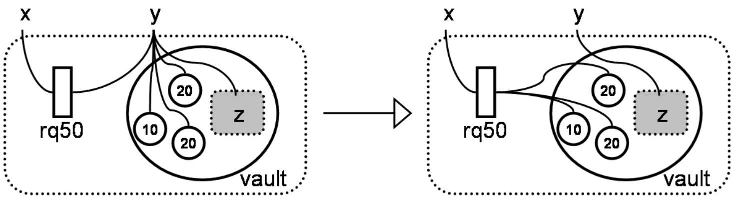
In some Brs containing the above reaction rule, we may also have rules stating that if the machine has an internal request for a sum of money then this request is linked with some notes, whose total value equals the requested sum. Two such rules are shown in Figure 15 where we omit the linking information in the second term. Again, we are supposing that we are able to use name binding. Using these rules in pure bigraphs, a user who has requested $50 may receive either one $50 note or else two $20 notes and a $10 note — there is a non-deterministic choice. With the kind version of these rules, our model would specify that if the vault contains a $50 note, then dispense that note *otherwise* dispense the smaller notes. We could add a third rule dispensing one $20 note and three $10 notes when only one $20 note exists and so on.

4.3.5 Draughts

We can use the static structure of kind bigraphs to model simple games and their rules. The game of draughts can be modelled using a signature $\mathcal{K} = \{\mathbf{B}, \mathbf{W}, \mathbf{KB}, \mathbf{KW}, \mathbf{S}, \mathbf{Trn}, \mathbf{Hop}\}$ where the elements represent black and white pieces and kings, squares, a 'turn' control, and a 'hop' control. Squares can contain pieces and kings and pieces and kings can contain turns and hops. The



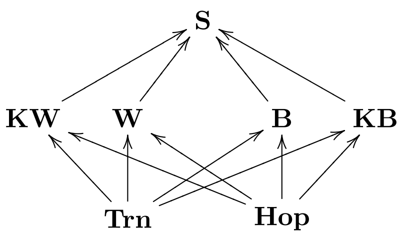
(a) $\mathbf{rq50}_{xy}|\mathbf{vault}(\mathbf{50}_y|\,{}^{y}\!/_{(z)}\,\square) \longrightarrow\!\!\triangleright\ /w(\mathbf{rq50}_{xw}|\mathbf{vault}(\mathbf{50}_w|\,{}^{y}\!/_{(z)}\,\square))$



(b) $\mathbf{rq50}|\mathbf{vault}(\mathbf{20}|\mathbf{20}|\mathbf{10}|\,{}^{y}\!/_{(z)}\,\boxed{\neg\mathbf{50}}) \longrightarrow\!\!\triangleright\ /w(\mathbf{rq50}|\mathbf{vault}(\mathbf{20}|\mathbf{20}|\mathbf{10}|\,{}^{y}\!/_{(z)}\,\boxed{\neg\mathbf{50}}))$

Figure 15: A request for \$50 is linked with notes of that value

containment structure is:



Squares have arity four and all other controls have zero arity. We assume a square-sorting on the link structure with five sorts (directions NE, NW, SE, SW and undirected a).

We present some of the rules of the game in Figures 16-20. We show the linkage in Figure 16 but omit it for simplicity in the remaining figures. In those figures, assume that any corner of a square not linked to another corner is uniquely linked to an outer name. A valid game of draughts is one in which the squares are linked in a n times n board pattern, where each square contains exactly one piece and there exists exactly one turn or hopper control (which by the signature is under some piece on the board.)

Figure 16 represents a white piece moving forwards. The white piece must contain a turn control, indicating that it is the white player's turn to move. Upon moving forwards, the turn control is then transferred to a black piece. Figure 17 represents a player changing their mind as to which piece on the board they wish to move. Figure 18 represents a black player taking a white piece. Upon taking the piece, the turn control is changed to a hop control. This indicates that the player is in the middle of a series of hops. In Figure 19, a player who was hopping has reached a point where there are no more hops available. The hop ends and the other player gets their turn. Figure 20 represents a black piece having reached the bottom of the board, whereupon the piece is crowned.

Kindness is not used a lot in this example but its contributions are there. For the most part, it merely induces a structure to the nesting. It is used in Figure 19 to specify that the sites may only contain playing pieces. In particular, we let this figure represent a family of rules where sites 0 and 3 are specified to contain at least a black or a white piece. As we are using fitting parameters, the rules assures that the player's hop ends if and only if there are no more hops available.

Again, the set of reaction rules is not prime simple as we use edges in redexes.

4.3.6 Boolean algebra

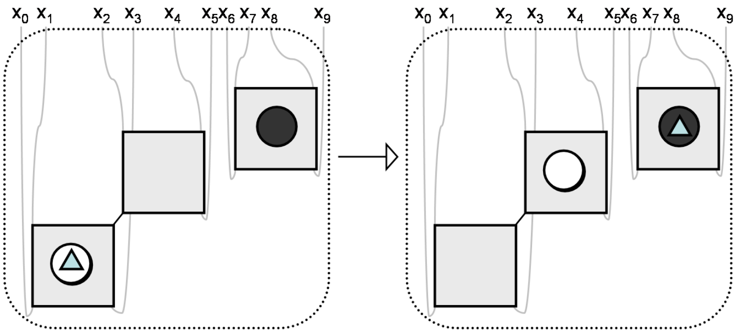
We now briefly introduce an encoding of boolean algebra as a kind Brs. Define the signature $\mathcal{K} = \{\mathbf{T}, \mathbf{F}, \wedge, \vee, \Rightarrow, \equiv, \neg\}$ where $\wedge$ and $\vee$ are the only non-atomic controls and they can contain all controls in $\mathcal{K}$. Let $ar(\mathbf{T}) = ar(\mathbf{F}) = ar(\wedge) = ar(\vee) = 1$, $ar(\neg) = 2$, and $ar(\Rightarrow) = ar(\equiv) = 3$. Call the first port of any control its 'tag-port'[10] and any other port a 'link-port'. We will gloss over some details here; we assume a link-sorting with a tag and a link sort where a link contains at most one tag and one link.

Figures 21-25 show some reaction rules representing the reduction of boolean terms to either true or false over this signature. We do not show all the reaction rules for implication, equivalence or negation but the missing rules are similar to those presented. The encoding is trivial and we will not dwell on it except to make two remarks.

The first remark is that the conjunction ($\wedge$) and disjunction ($\vee$) operators are represented by non-atomic controls in contrast to the implication, equivalence, and negation operators which are represented by 'binary' and 'unary' controls as one might expect. We could represent conjunction and disjunction with binary operators in the obvious way and then have rules like '$\mathbf{T}\wedge\mathbf{F} \longrightarrow\!\!\triangleright\ \mathbf{F}$'. With this design choice we could then define a link reactive system (informally, a Brs with no place graphs) to capture the reaction rules. However, we have represented conjunction and disjunction as non-atomic controls for two reasons. The first is that $\wedge$ and $\vee$ are associative and commutative – they are not 'directed' like implication. This means that we can represent $p_1 \wedge p_2 \wedge \ldots \wedge p_n$, where $p_i \in \{\mathbf{T}, \mathbf{F}\}$, as $\wedge(p_1, p_2, \ldots, p_n)$ as in the model. The second reason is that their behaviour can be explained with quantifiers *i.e.* the conjunction of n boolean values is false if at least one value is false and is true otherwise. This is the intuition behind our reaction rules in which we exploit kindness to encode the quantification. This exploitation leads us to efficient reaction rules where a conjunction of n variables may reduce to false as soon as any of its conjuncts reduces to false.

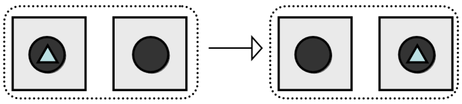
The second remark is of a similar vein. Note that the first reaction rule for implication covers both cases where the precedent is false. Any term that is connected to the 'antecedent port' of the implication via composition has its link severed after reaction — its tag port is then connected to an edge and it is essentially a useless term. The garbage collection 'rule' is a schema for a rule that can be described as "when a term is closed off, remove the term".

[10] This terminology is taken from the encoding of the λ-calculus in bigraphs [13].



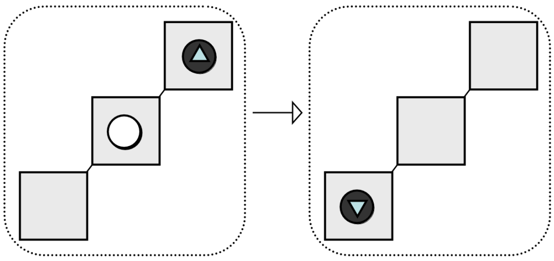
$$/x(\mathbf{S}_x(\mathbf{W}(\mathbf{Trn}))|\mathbf{S}'_x|\mathbf{S}''(\mathbf{B})) \longrightarrow /x(\mathbf{S}_x|\mathbf{S}'_x(\mathbf{W})|\mathbf{S}''(\mathbf{B}(\mathbf{Trn})))$$

Figure 16: A piece advances



$$\mathbf{S}(\mathbf{B}(\mathbf{Trn}))|\mathbf{S}'(\mathbf{B}') \longrightarrow \mathbf{S}(\mathbf{B})|\mathbf{S}'(\mathbf{B}'(\mathbf{Trn}))$$

Figure 17: A player considers another move



$$/x/y(\mathbf{S}_x|\mathbf{S}'_{xy}(\mathbf{W})|\mathbf{S}''_y(\mathbf{B}(\mathbf{Trn}))) \longrightarrow /x/y(\mathbf{S}_x(\mathbf{B}(\mathbf{Hop})))|\mathbf{S}'_{xy}|\mathbf{S}''_y)$$

Figure 18: A player takes a piece

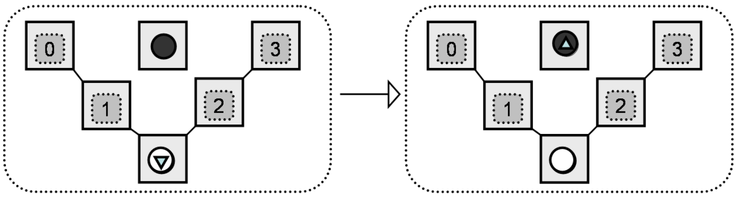
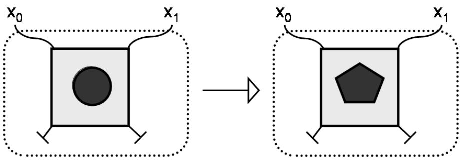


Figure 19: A hop ends (all sites contain playing pieces)



$$/y/z(\mathbf{S}_{x_0x_1yz}(\mathbf{B})) \longrightarrow\!\!\triangleright\ /y/z(\mathbf{S}_{x_0x_1yz}(\mathbf{KB}))$$

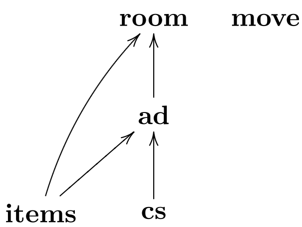
Figure 20: A piece is crowned

In summary, this encoding of boolean algebra may be trivial but the design choices of the modelling process may be of interest.

4.3.7 No tea

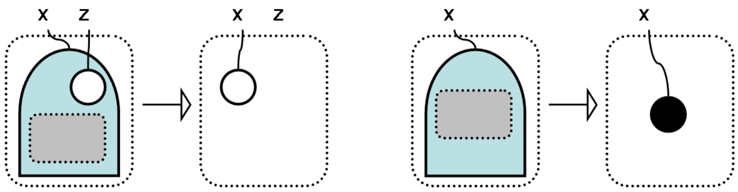
Our last example is purely for fun! There are many computer games where the player must navigate through a map, collecting and dropping items on his quest. We will represent the basic structure of such a game as follows.

Let $\mathcal{K}$ consist of a control **room** with arity four representing rooms, n controls **item1**, **item2**, … , **itemn** of arity one representing items, a control **ad** of arity one representing players, a control **cs** representing common sense and a control **move** of arity two representing the actions take/drop. Let the graph representing the *kind* map be



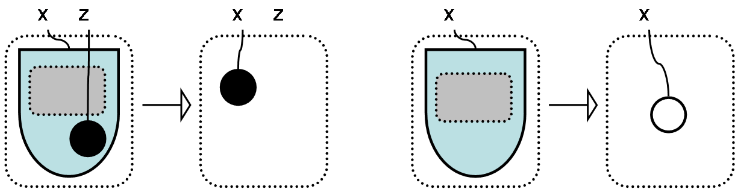
where **items** represents all items *i.e.* the graph states that a room or a player may contain any item. The *kind* map allows any item node to be inside a player node. We will refer to the set of item nodes that a player node contains as the player's inventory. This signature would also use an obvious square-sorting for creating a game map.

The two rules in Figure 26 represent the actions 'take' and 'drop' (the rules are mirror images of each other so we have unconventially represented them both using one figure) which move items in and out of a player's inventory. We assume in these rules and in the next rule that the four ports of a room are linked to four names in the redex and reactum as in the example of Section 4.3.1. This would allow similar rules for moving north, south, east and west which is common in such games. Now comes the fun part! Let **itemn** represent tea. We will refer to that item specifically as **tea** for convenience. Figure 27 represents the action 'drop no tea'. Note that in the redex, the player's inventory is specified to contain no tea and no common sense by stating that the site does not contain **tea** or **cs**. In the reactum, however, the player's inventory does indeed contain tea! Our rule then describes that if a player has no common sense and no tea then they may drop no tea and end up with tea!



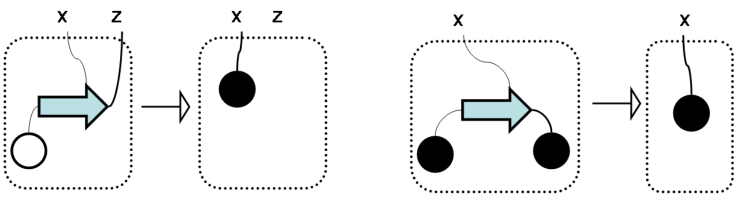
$\wedge_x(\mathbf{F}_z|\square) \longrightarrow \mathbf{F}_x|z$ $\qquad$ $\wedge_x(\boxed{\mathbf{T}}) \longrightarrow \mathbf{T}_x$

Figure 21: Boolean AND



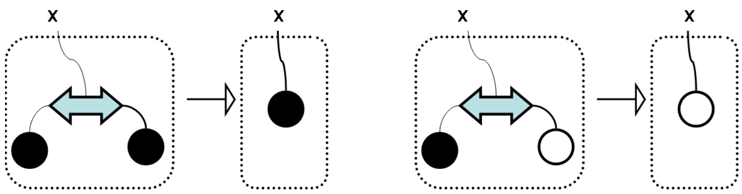
$\vee_x(\mathbf{T}_z|\square) \longrightarrow \mathbf{T}_x|z$ $\qquad$ $\vee_x(\boxed{\mathbf{F}}) \longrightarrow \mathbf{F}_x$

Figure 22: Boolean OR



$/w(\mathbf{F}_w|\Rightarrow_{xwz}) \longrightarrow \mathbf{T}_x|z$ $\qquad$ $/w/z(\mathbf{T}_w|\Rightarrow_{xwz}|\mathbf{T}_z) \longrightarrow \mathbf{T}_x$

Figure 23: Boolean IMPLIES

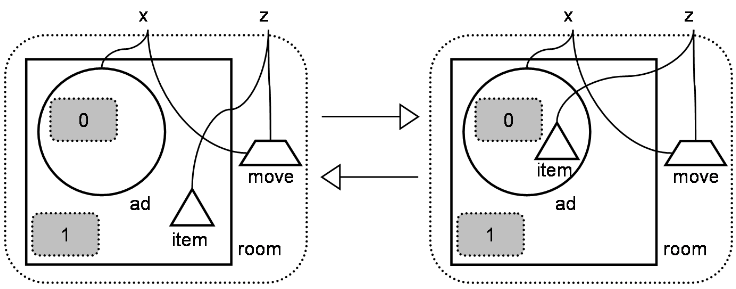


$/w/z(\mathbf{T}_w|\equiv_{xwz}|\mathbf{T}_z) \longrightarrow \mathbf{T}_x$ $\qquad$ $/w/z(\mathbf{T}_w|\equiv_{xwz}|\mathbf{F}_z) \longrightarrow \mathbf{F}_x$

Figure 24: Boolean EQUIV

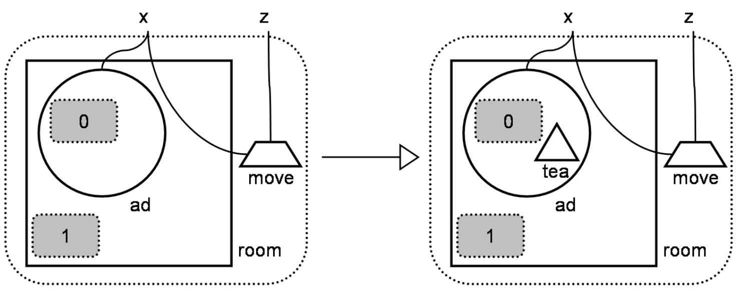
$$/w(\neg_{xw}|\mathbf{F}_w) \;\longrightarrow\!\!\triangleright\; \mathbf{T}_x$$

Figure 25: Boolean NOT and a garbage collection ‘rule’



$$\mathbf{room}(\mathbf{ad}_x\Box_0|\mathbf{item}_z|\Box_1)|\mathbf{move}_{zx} \;\longrightarrow\!\!\triangleright\; \mathbf{room}(\mathbf{ad}_x(\mathbf{item}_z|\Box_0)|\Box_1)|\mathbf{move}_{zx}$$

Figure 26: Taking and dropping items

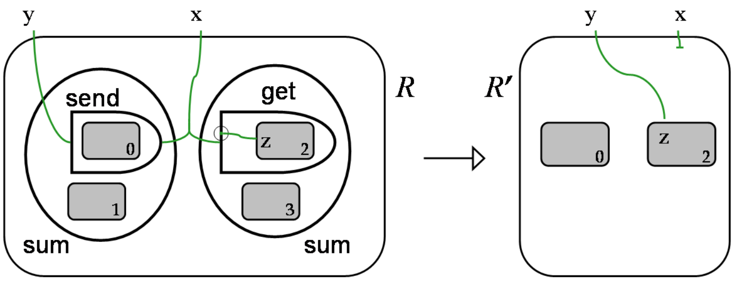


$$\mathbf{room}(\mathbf{ad}_x\,\boxed{\neg\mathbf{tea}\neg\mathbf{cs}}\,|\Box)|\mathbf{move}_{zx} \;\longrightarrow\!\!\triangleright\; \mathbf{room}(\mathbf{ad}_x(\mathbf{tea}|\,\boxed{\neg\mathbf{tea}\neg\mathbf{cs}}\,)|\Box)|\mathbf{move}_{zx}$$

Figure 27: Drop ‘no tea’

4.4 Restrictiveness of fitting Brss

Kind bigraphs generalise the notion of atomicity in pure bigraphs and the static theory of fitting bigraphs seems to be a conservative extension of pure bigraphs. However, while the dynamic theory of fitting bigraphs has some nice properties, it fails to conservatively extend the pure dynamic theory. Figure 28 shows a proposed reaction rule given in [7] for the π-calculus with summation. The rule uses binding bigraphs but we will just concentrate on the place graph structure of the rule. We can define this reaction rule using kind bigraphs, with



$$\mathrm{sum}(\mathrm{send}_{xy}\square_0|\square_1)|\mathrm{sum}(\mathrm{get}_{x(z)}\square_2|\square_3) \longrightarrow x|\square_0|\, {}^{y}\!/_{(z)}\,\square_2$$

Figure 28: Reaction rule for the π-calculus with summation [7]

$\langle 1, (\{\mathrm{send}, \mathrm{get}, \mathrm{sum}\}), \{x, y\}\rangle$ as the outer interface of the redex and reactum. However, we can not define this reaction rule using *fitting* bigraphs as this outer interface is not fitting for the redex. The redex and reactum of a reaction rule must have the same outer face in any Brs and so without changing the central theory, we can not encode this rule as such in a fitting Brs. We return briefly to this problem in Section 4.4.3.

Fitting Brss can be used when all reaction rules can have the same fitting outer interface. This is not the general case though, as we have seen above. We identify all cases of Brss below and give a conjectured (as we have not developed the dynamic theory of kind bigraphs yet) solution which uses a kind signature, or variant of, for each case. This will hopefully allow us to apply our theory to a greater range of systems where kindness is appropriate[11].

For a root r in the redex (resp. reactum), denote the union of the set of controls of the children of r and the sets C_s, where s is a site under r, as R_r (resp. R'_r.) Call any node under a root r an *ancestor* of r. For a root r in the redex (resp. reactum), denote the union of the set of controls of the ancestors of r and the sets C_s, where s is a site under r, as $R_{<r}$ (resp. $R'_{<r}$.)

The set of all possible Brss, with respect to the place graph structure, can be split into five (sometimes overlapping) cases as follows:

For all rules (R, R') in the Brs, and all roots r of the rule,

1. $R_r = R'_r$.

2. $R_r > R'_r$.

[11]It is not appropriate to use kindness for an encoding of the π-calculus with summation but we may wish to alter such a calculus and add kindness.

3. $R_{<r} = R'_{<r}$.

4. $R_{<r} > R'_{<r}$.

For at least one rule (R, R') in the Brs, and at least one root r of the rule,

5. Either $R_r < R'_r$ or $R_{<r} < R'_{<r}$ or both.

This section concludes with some discussion on how to use a kind Brs, or some variant of, for each of these cases. We think that we have an acceptable solution for the first four. In all the cases, we will assume that parameters of parametric reaction rules are fitting.

4.4.1 First case

In this case, it would be sensible to use a fitting Brs as it has the property of being a well-behaved place-sorting.

4.4.2 Second case

We can define any such rule (R, R') with a pair of kind bigraphs where the redex is fitting. A corollary of Proposition 3.9 is that if we ensure that all parametric redexes are fitting and all parameters are fitting, then a kind Brs will have fitting labels in the standard transition systems. As the forgetful functor from kind bigraphs to pure bigraphs creates RPOs, $\sim_{\mathrm{ST}}$ is a congruence by Theorem 4.13 and so we propose that this case can be covered by using a kind Brs with fitting redexes. Note that the reaction rule of the asynchronous π-calculus can be encoded in such a kind Brs.

Proposition 3.9 holds for any fitting bound. We thus propose a new type of transition system for kind bigraphs, similar but smaller to the full transition system and called a *fitting* transition system. The fitting transition system for a kind Brs consists of all interfaces, together with all transitions where the bound is fitting. By Proposition 3.9, when redexes (and parameters) are fitting, all labels of this transition system are fitting. This notion will be explored in future work.

4.4.3 Third case

The problem with encoding the summation rule for the asynchronous π-calculus is that the kind rules only care about the children of a place. We can change the kind rules as follows:

Definition 4.23 (strong kind sorting). *In a* strong kind *sorting* $\Sigma_{\mathrm{K}} = (\mathcal{K}, \mathcal{P}(\mathcal{K}), \Phi)$ *the condition* Φ *assigns a sort* $\theta \in \mathcal{P}(\mathcal{K})$ *to each control in* $\mathcal{K}$. *Then a bigraph* G *is admissible iff it satisfies:*

K1 *if* $p = G(v)$ *then* $\mathit{ctrl}(v) \cup \mathit{sort}(v) \subseteq \mathit{sort}(p)$

K2 *if* $p = G(s)$ *then* $\mathit{sort}(s) \subseteq \mathit{sort}(p)$

K3 *if* $\mathit{sort}(v) = \emptyset$, v *has no children*

where p *is a root or node,* s *a site, and* v *is a node.*

Now, a bigraph is only admissible if the control of all ancestors of a node or root are elements of the sort of the node or root. If we return to the example of the summation rule, the sort of the root in the outer interface of the redex must now contain $\mathrm{send} \cup (\mathrm{get} \cup \mathrm{send} \cup \mathrm{sum}) = (\mathrm{get} \cup \mathrm{send} \cup \mathrm{sum})$, assuming that any control may contain any other control (*i.e.* all controls are non-atomic as in the original encoding.) The sort of the root in the outer interface of the reactum must also be $(\mathrm{get} \cup \mathrm{send} \cup \mathrm{sum})$. So using this stronger (or perhaps 'deeper' or transitive is a more intuitive description) notion of kind, we can define this rule as a 'strong fitting' rule.

It remains to formally show that this strong kindness is a place sorting. Informally, it is obvious that it is satisfied by the identities and preserved by tensor product. The proof of preservation by composition should be trivial and similar to Proposition 2.4. Intuitively, it would seem that the associated forgetful functor into pure bigraphs would create RPOs and that the forgetful functor from strong fitting bigraphs into pure bigraphs would reflect pushouts but we will not prove these properties here.

As a motivation for strong kindness, we make the conjecture that any pure Brs where all prime reaction rules have at least one site or non-atomic node in both their redex and their reactum may be also encoded as a strong fitting Brs. We base this conjecture on the fact that a pure site or non-atomic node may contain any control. We can then define a strong kind rule based on the kind rule, where the sort of any site is $\mathcal{K}$. The sort of the root in both the redex and the reactum must then be $\mathcal{K}$.

Strong kindness is discussed informally again in Appendix B.

4.4.4 Fourth case

We propose that this case may be covered by a strong kind Brs with fitting redexes. We would hope to find that a proposition similar to Proposition 3.9 holds for strong kind bigraphs. This would then reduce our set of labels to the strong fitting labels. Again, the theory of strong kindness would need to be developed to cover this case.

4.4.5 Fifth case

An example of such a Brs is one in which the reactum of a rule may introduce a control under some root r which is neither an ancestor of r in the redex nor a member of C_s for any $s <_R r$.

We have two suggestions for such Brss. The first is to use a homomorphic sorting with more than one sort, if possible, to resolve the issue. This will only work in certain cases.

The second suggestion is to use either a kind Brs where a rule has either a fitting redex or a fitting reactum, or else a strong kind Brs with the same condition. Any rule in a kind Brs with a fitting redex will produce transitions with fitting labels for the full and standard transition systems. That is the best we can guarantee without developing the dynamic theory of kind bigraphs.

5 Future work

This report details the static theory of kind bigraphs. The dynamic theory will be explored in a future report. The aim will be to present a dynamic theory where the kind rules are respected by reactions in the system.

In this report we have generalised the notion of atomicity in place graphs. An alternative idea is to generalise the notion of activity with respect to controls. In the pure theory, a non-atomic control may be passive or active. Nodes with an active control allow reactions inside themselves. We could combine kind bigraphs with a generalisation of this activity by defining a set of activity functions

$$actv_K : kind(K) \to \{active, passive\}$$

for each $K \in \mathcal{K}$, where *kind* refers to the *function* $kind : \mathcal{K} \to \mathcal{P}(\mathcal{K})$ and where $actv_K(K') = active$ means that a K node allows reactions involving any children which are K' nodes. This may allow us to model, say, that a public system may contain honey pots and resources but that reaction may only happen inside resources.

Another possible generalisation of kind bigraphs is adding the notion of the *capacity* of a control. This generalisation is also motivated by the model of the Confederation of the Islands [10] which is being investigated by the author. This model has the notion of the capacity of an entity *e.g.* 'a town may contain n citizens'. A formulation of kind bigraphs with capacity would describe these bigraphs using a place-sorting. The interfaces would be enriched to state the number of nodes of a control K that a root may contain. We will give an informal example of how 'strong capacity' can be described (we say 'strong' as we consider ancestors as with strong kindness.) Say a place of an interface r can contain n nodes of control K. Represent this as $\mathrm{Cpc}_r = n$. Let G be a bigraph with r as a root and $m =$ the number of nodes of control K under r in G. Then G is admissible as a kind bigraph with capacity if

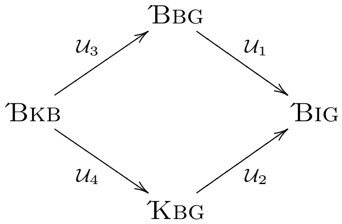
$$n \geq m + \sum_{s, s <_G r} \mathrm{Cpc}_s.$$

We have shown that kind bigraphs are a particular place-sorting. It may also be possible to define kind link graphs in an analogous way to kind place graphs so that kind link graphs are a particular link-sorting. An analogous notion of fitting could also be defined whereas the notion of strong kindness would not be an issue. If such a link-sorting turned out to be well-behaved, it could be very expressive. It could also be immediately combined with kind place graphs, as the structures would be orthogonal.

There is another direction to follow. Binding bigraphs [7] are a generalisation of pure bigraphs[12], as are kind bigraphs. It seems feasible that a combination of both approaches should yield a theory with all the properties of RPO creation and correspondences of IPOs. Such a hybrid bigraph form would be a further generalisation. If the details worked out then we would have a diagram of

[12]Local bigraphs [13] are a further generalisation of binding bigraphs

forgetful functors:



A proof of RPO creation in BKB could be simplified by proving that either $\mathcal{U}_3$ or $\mathcal{U}_4$ created RPOs, instead of forgetting all structure to the level of pure bigraphs. Proving that $\mathcal{U}_3$ created RPOs would seem easier to the author and such a proof could follow the same approach given in this report.

Spatial logic for graphs of the form in [2, 3] has recently been studied in the bigraphical setting [4]. We have shown here that one may make logical statements about kind bigraphs as to what controls a root may contain and what controls the parent of a site may contain after composition. It would be interesting to explore the role of kind bigraphs within this spatial logic for bigraphs [4].

6 Acknowledgements

The inspiration to explore the remark made in [7] about assigning a 'kind' to a node, upon which this work is based, occurred towards the end of the author's visit at the IT University (ITU), Copenhagen. This research and that visit was partially supported by funding from the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan. The visit was also made possible by the support of my supervisor, Mícheál Mac an Airchinnigh, and Lars Birkedal, ITU. My time in Copenhagen was also made very enjoyable and beneficial thanks to all the members of the Bigraphical Programming Languages (BPL) group and the Theory department that I encountered. It was in fact suggested to me during a BPL meeting that I would somehow need to extend the pure bigraph interfaces to encode the notion of 'kind' in kind bigraphs.

The author also wishes to thank Mícheál Mac an Airchinnigh and Søren Debois for their comments and suggestions on this report.

A Pure bigraphs

Here we present the notation and some of the original definitions from the pure bigraph theory of Jensen and Milner [7], updating some definitions with the notation and terminology of Milner [14]. All definitions in this appendix are taken either verbatim or with minor modifications from the former source with the exception of a few definitions which we reproduce in the revised form of the latter source. We number the definitions with their numbering in the former and if the definition has been significantly revised, we mark the title with '(R)'. We only reproduce the definitions and propositions here and so refer the reader to [7] and [14] for detailed explanations. Any abuses of notation are typically mentioned in the order that they appear in the original text and so are also contained within Section A.2.

A.1 Notation

The names of s-categories will always be accented, as in $\acute{\mathbf{C}}$. '$\circ$', 'id' and '$\otimes$' are used for composition, identity and tensor product respectively. Juxtaposition will also be used for composition. The domain I and codomain J of an arrow $f : I \to J$ are denoted by $\mathrm{dom}(f)$ and $\mathrm{cod}(f)$; the set of arrows from I to J, called a *homset*, is denoted by $\acute{\mathbf{C}}(I \to J)$.

Id_S will denote the identity function on a set S, and $\emptyset_S$ the empty function from $\emptyset$ to S. $S \uplus T$ denotes the union of sets S and T known or assumed to be disjoint and $f \uplus g$ for union of functions whose domains are known or assumed to be disjoint. This use of $\uplus$ on sets should not be confused with the disjoint sum '+', which disjoins sets *before* taking their union. A fixed representation of disjoint sums is assumed; for example, $X + P + Y$ means $(\{0\} \times X) \cup (\{1\} \times P) \cup (\{2\} \times Y)$, and $\sum_{v \in V} P_v$ means $\bigcup_{v \in V} (\{v\} \times P_v)$.

$f \restriction S$ denotes the restriction of a function f to the domain S, and $R \restriction S$ denotes the restricted relation $R \cap S^2$.

A natural number m is often interpreted as a finite ordinal $m = \{0, 1, \ldots, m-1\}$. i is often used to range over m; when $m = 2$, $\bar{\imath}$ is used for the complement $1-i$ of i. $\vec{x}$ denotes a finite sequence $\{x_i \mid i \in m\}$; when $m = 2$ this is an ordered pair.

In general, $I, J, K, \ldots$ are used to stand for objects of a (pre)category and $f, g, h, \ldots$ for arrows.

When discussing bounds, RPOs and IPOs (defined below in Definitions 3.7, 3.8 and 3.9 respectively) $\vec{f}$ 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 $\vec{f} : H \to \vec{I}$.

A.2 Pure definitions

We have split the definitions into sections corresponding to those in which they originally appeared. The exception is Definition 4.1 (wide s-category, wide functor) which is presented with the categorical definitions as it is the only definition of Section 4 required for reading this report.

A.2.1 S-categories and relative pushouts

We assume that the reader is familiar with the basic categorical notions of category, subcategory and functor.

[Definition 3.1 (precategory)] A *precategory* $\acute{\mathbf{C}}$ is defined exactly as a category, except that the composition of arrows is not always defined. Composition with the identities is always defined, and $\mathrm{id}f = f = f\mathrm{id}$. In the equation $h(gf) = (hg)f$, the two sides are either equal or both undefined.

[Definition 3.2 (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) $\gamma_{I,\epsilon} = \mathrm{id}_I$

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

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

[Definition 3.3 (s-category)] (R) 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 $\rho \cdot f : I \to J$ called a *support translation* of f such that

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

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

(3) $\rho \cdot (f \otimes g) = \rho \cdot f \otimes \rho \cdot g$.

(4) $\mathrm{Id}_{|f|} \cdot f = f$

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

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

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

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

[Definition 3.4 (support equivalence, supported functor)] (R) Two arrows $f, g : I \to J$ in an s-category $\acute{\mathbf{A}}$ are *support-equivalent*, written $f \bumpeq g$, if $\rho{\bullet}f = g$ for some support translation ρ. By Definition 3.3 this is an equivalence relation. If $\acute{\mathbf{B}}$ is another s-category, then a *supported functor* $\mathcal{F} : \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ is a function on objects and arrows that preserves identities, composition, tensor product and support equivalence. If $\mathcal{F}$ is an inclusion function then $\acute{\mathbf{A}}$ is a *sub-s-category* of $\acute{\mathbf{B}}$.

[Definition 3.5 (congruence)] Let $\equiv$ be an equivalence defined on every homset of a s-category $\acute{\mathbf{C}}$. It is said that $\equiv$ is *preserved* by an operator $*$ if $f \equiv f'$ and $g \equiv g'$ imply $f * g \equiv f' * g'$ whenever the latter are defined. Then $\equiv$ is a *congruence on* $\acute{\mathbf{C}}$ whenever it is preserved by composition and tensor product.

[Definition 3.6 (quotient categories)] Let $\acute{\mathbf{C}}$ be an s-category, and let $\equiv$ be a congruence on $\acute{\mathbf{C}}$ that includes support equivalence, *i.e.* $\bumpeq \subseteq \equiv$. Then the *quotient* of $\acute{\mathbf{C}}$ by $\equiv$ is a category $\mathbf{C} \stackrel{\text{def}}{=} \acute{\mathbf{C}}/\equiv$, whose objects are the objects of $\acute{\mathbf{C}}$ and whose arrows are equivalence classes of arrows in $\acute{\mathbf{C}}$:

$$\mathbf{C}(I, J) \stackrel{\text{def}}{=} \{[f]_\equiv \mid f \in \acute{\mathbf{C}}(I, J)\}.$$

In $\mathbf{C}$, the identities, composition and tensor product are given by

$$\begin{aligned} \mathrm{id}_m &\stackrel{\text{def}}{=} [\mathrm{id}_m]_\equiv \\ [g]_\equiv \circ [f]_\equiv &\stackrel{\text{def}}{=} [gf]_\equiv \\ [f]_\equiv \otimes [g]_\equiv &\stackrel{\text{def}}{=} [f \otimes g]_\equiv \end{aligned}$$

By assigning empty support to every arrow, $\mathbf{C}$ may also be regarded as an s-category and $[\![\cdot]\!]_\equiv : \acute{\mathbf{C}} \to \mathbf{C}$ is called the $\equiv$-*quotient functor* for $\acute{\mathbf{C}}$.

In the following definition, **Ord** is the s-category of finite ordinals and functions between them.

[Definition 4.1 (wide s-category)] (R) An s-category $\acute{\mathbf{A}}$ is *wide* if equipped with a functor width : $\acute{\mathbf{A}} \to \mathbf{Ord}$ with $\mathrm{width}(\epsilon) = 0$ such that, for each bijection π on the ordinal $\mathrm{width}(I)$, there is an isomorphism $\pi_I : I \to I$ in $\acute{\mathbf{A}}$ with $\mathrm{width}(\pi_I) = \pi$.

The objects $I, J, \ldots$ of $\acute{\mathbf{A}}$ are called *interfaces*, and its arrows $A, B, C, \ldots$ are called *contexts*. The domain and codomain of a context will be called its *inner* and *outer faces*. Arrows in a homset $\acute{\mathbf{A}}(\epsilon \to I)$ — which is often abbreviated to $\acute{\mathbf{A}}(I)$ — are called *ground* arrows; lower case letters $a, b, \ldots$ range over these, and $a : \epsilon \to I$ can be abbreviated to $a : I$.

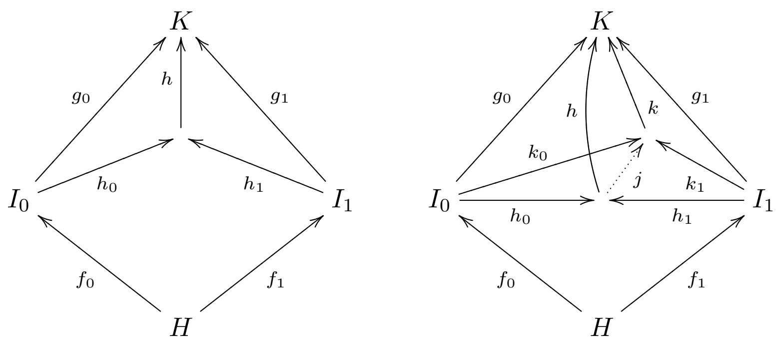
Note that in kind bigraphs, the isomorphisms π_I typically have domain I and codomain I' with $I \neq I'$. This is fine, however, as both $\mathrm{width}(I) = \mathrm{width}(I')$ and $\mathrm{width}(\pi_I) = \pi$ hold.

I

[Definition 3.7 (bound, consistent)] If two arrows $\vec{f} : H \to \vec{I}$ share domain H, the pair $\vec{g} : \vec{I} \to K$ share codomain K and $g_0 f_0 = g_1 f_1$, 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 3.8 (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 $hh_i = g_i (i = 0, 1)$. 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 $jh_i = k_i (i = 0, 1)$ and $kj = h$.



[Definition 3.9 (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 $\vec{f}$ to $\vec{h}$.

A.2.2 Pure bigraphs — structure

[Definition 6.1 (pure signature)] A *(pure) signature* $\mathcal{K}$ is a set whose elements are called *controls*. For each control K it provides a finite ordinal $ar(K)$, an *arity*; it also determines which controls are *atomic*, and which of the non-atomic controls are *active*. Controls which are not active (including the atomic controls) are called *passive*.

[Definition 6.2 (concrete pure bigraph)] A *(concrete) pure bigraph* over the signature $\mathcal{K}$ takes the form $G = (V, E, ctrl, G^{\mathrm{P}}, G^{\mathrm{L}}) : I \to J$ where $I = \langle m, X \rangle$ and $J = \langle n, Y \rangle$ are its *inner* and *outer faces*, each combining a *width* (a finite ordinal) with a finite set of global names drawn from a denumerable set $\mathcal{X}$[13]. Its first two components V and E are finite sets of *nodes* and *edges* respectively. The third component $ctrl : V \to \mathcal{K}$, a *control map*, assigns a control to each node. The remaining two are:

$$G^{\mathrm{P}} = (V, ctrl, prnt) : m \to n \quad \text{a } \textit{place graph}$$
$$G^{\mathrm{L}} = (V, E, ctrl, link) : X \to Y \quad \text{a } \textit{link graph.}$$

Place graphs and link graphs are defined in the next two sections. G is called the *combination* of its *constituents* G^{P} and G^{L}; it is written as $G = \langle G^{\mathrm{P}}, G^{\mathrm{L}} \rangle$.

[13]This denumerable set $\mathcal{X}$ of *global names* is presupposed in the theory.

A.2.3 Pure place graphs

[Definition 7.1 (place graph)] (R) A *place graph* $A = (V, ctrl, prnt) : m \to n$ has an *inner width* m and an *outer width* n, both finite ordinals; a finite set V of nodes with a control map $ctrl : V \to \mathcal{K}$; and a *parent map* $prnt : m \uplus V \to V \uplus n$. $w >_A w'$, or just $w > w'$, means that $w = prnt^k(w')$ for some $k > 0$. The parent map is *acyclic*, *i.e.* $prnt^k(v) \neq v$ for all $k > 0$ and $v \in V$ (equivalently, $>_A$ is a partial order.) An *atom*, *i.e.* a node with atomic control, may not be a parent.

The widths m and n index the *sites* and *roots* of A respectively. The sites and nodes — *i.e.* the domain of *prnt* — are called *places*. A place graph is *hard* if every root, and every node except an atom, has a child.

[Definition 7.2 (s-category of place graphs)] (R) The s-category ´PLG has finite ordinals as objects and place graphs as arrows. The support of a place graph is its node set. The composition $A_1A_0 : m_0 \to m_2$ of two place graphs $A_i = (V_i, ctrl_i, prnt_i) : m_i \to m_{i+1} (i = 0, 1)$ with disjoint supports is $A_1A_0 \stackrel{\text{def}}{=} (V, ctrl, prnt)$ 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 m is $\mathrm{id}_m \stackrel{\text{def}}{=} (\emptyset, \emptyset_{\mathcal{K}}, \mathrm{Id}_m) : m \to m$.

The *tensor product* $\otimes$ in ´PLG is defined as follows: On objects, $m \otimes n \stackrel{\text{def}}{=} m + n$. For the product $A_0 \otimes A_1$ of two place graphs with disjoint support, the support of the product is the union of their node set; for the parent map, if $A_0 : m_0 \to n_0$, first offset the sites and roots of A_1 by m_0 and n_0 respectively, then take the union of the two parent maps.

For an injective map ρ on nodes, the support translation $\rho \bullet A$ is defined by systematic replacement of each node v by $\rho(v)$, preserving all structure.

When dealing with many place graphs $A, B, \ldots$, instead of indexing their parent maps as $prnt_A, prnt_B$ etc. it will be more convenient to abuse notation and denote the parent map of a place graph A again by A.

When considering a pair $\vec{A} : h \to \vec{m}$ of place graphs with common sites h, a convention is adopted for naming their nodes. The node set of A_i $(i = 0, 1)$ is denoted by V_i, and $V_0 \cap V_1$ is denoted by V_2. Recall that $\bar{\imath}$ means $1 - i$ for $i \in 2$. $v_i, v_i', \ldots$ will be used to range over V_i $(i = 0, 1, 2)$, and r_i, r_i' will be used to range over the roots m_i $(i = 0, 1)$. $w_2, w_2', \ldots$ will also be used to range over $h \uplus V_2$.

[Definition 7.3 (barren, sibling, active, passive)] A node or root is *barren* if it has no children. Two sites are *siblings* if they have the same parent. A site s of A is *active* if $ctrl(v)$ is active whenever $v > s$; otherwise s is *passive*. If s is active (resp. passive) in A, it will also be said that A is *active* (resp. *passive*) *at* s.

[Propositions 7.4, 7.6 (isomorphisms, epis and monos in place graphs)] *An arrow* $\iota : m \to m$ *in* ´PLG *is an isomorphism iff it has no nodes, and its parent map is a bijection. In* ´PLG, *a place graph is an epi iff no root is barren; it is mono iff no two sites are siblings.*

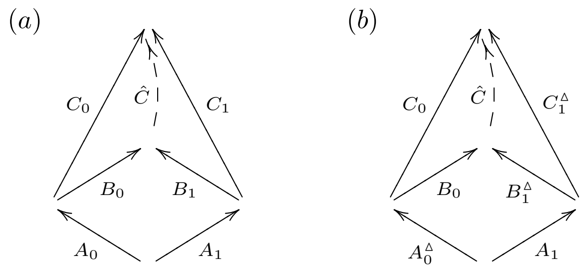
If in a place graph no two sites are siblings, it is also said that the place graph is *inner-injective*.

[Definition 7.9 (consistency conditions for place graphs)] Three *consistency* conditions are defined on a pair $\vec{A} : h \to \vec{m}$ of place graphs. Let i range over $\{0,1\}$; also let w_2, w_2' range over $h \uplus V_2$, the shared places.

CP0 $ctrl_0(v_2) = ctrl_1(v_2)$
CP1 If $A_i(w) \in V_2$ then $w \in h \uplus V_2$ and $A_{\bar{\imath}}(w) = A_i(w)$
CP2 If $A_i(w_2) \in V_i - V_2$ then $A_{\bar{\imath}}(w_2) \in m_{\bar{\imath}}$, and if also $A_{\bar{\imath}}(w) = A_{\bar{\imath}}(w_2)$ then $w \in h \uplus V_2$ and $A_i(w) = A_i(w_2)$.

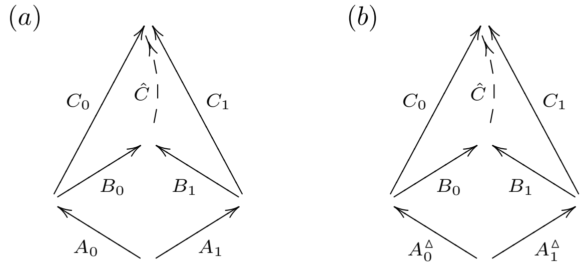
[Propositions 7.14 (pushouts for hard place graphs)] *If $\vec{A}$ is a consistent pair of hard kind place graphs, then the pushout $\vec{B}$ built in ´PLG by [7, Construction 7.11] is also hard, and is indeed a pushout in ´PLG$_\mathrm{h}$.*

[Propositions 7.15 (first pushout variation)] *Let $\vec{B}$ be a bound for $\vec{A}$ in ´PLG$_\mathrm{h}(\mathcal{K}^\triangle)$. Add a new place node $\triangle$ to both A_0 and B_1, yielding $A_0^\triangle$ and $B_1^\triangle$ such that $B_0 \circ A_0^\triangle = B_1^\triangle \circ A_1$. Then $\vec{B}$ is a pushout for $\vec{A}$ iff $(B_0, B_1^\triangle)$ is a pushout for $(A_0^\triangle, A_1)$.*



[Propositions 7.16 (second pushout variation)] *Let $\vec{B}$ be a bound for $\vec{A}$ in ´PLG$_\mathrm{h}(\mathcal{K}^\triangle)$. Let a fresh place node $\triangle$ be added to both members of $\vec{A}$, yielding $\vec{A}^\triangle$ such that $\vec{B}$ is also a bound for $\vec{A}^\triangle$, and with $A_0^\triangle(\triangle)$ a node (not a root). Then*

1. *If $\vec{B}$ is a pushout for $\vec{A}$, it is also a pushout for $\vec{A}^\triangle$.*
2. *Let $\triangle$ have a sibling w in both $A_0^\triangle$ and $A_1^\triangle$. Then if $\vec{B}$ is a pushout for $\vec{A}^\triangle$, it is also a pushout for $\vec{A}$.*



A.2.4 Link graphs

Since this report does not directly concern link graphs, we only reproduce those definitions which arise in definitions in the following section.

[Definition 8.1 (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.

The inner names X and ports P are called the *points* of A, and the edges E and outer names Y are called the *links* of A.

[Definition 8.2 (idle, open, closed, peer, lean)] A link is *idle* if it has no preimage under the *link* map. An (outer) name is an *open* link, an edge is a *closed* link. A point (*i.e.* an inner name or port) is *open* if its link is open, otherwise *closed*. Two distinct points are *peers* if they are in the same link. A link graph is *lean* if it has no idle edges.

[Definition 8.3 (s-category of link graphs)] The s-category $\text{'L}\textsc{ig}$ 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 $\mathrm{id}_X \stackrel{\text{def}}{=} (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, \mathrm{Id}_X) : X \to X$.

[Definition 8.10 (consistency conditions for link graphs)] Three *consistency* conditions are defined on a pair $\vec{A} : W \to \vec{X}$ of link graphs. Let p range over arbitrary points, $p_i, p'_i, \ldots$ to range over P_i, and $p_2, p'_2, \ldots$ to range over $W \uplus P_2$, the shared points.

- CL0 $ctrl_0(v_2) = ctrl_1(v_2)$
- CL1 If $A_i(p) \in E_2$ then $p \in W \uplus P_2$ and $A_{\bar{\imath}}(p) = A_i(p)$
- CL2 If $A_i(p_2) \in E_i - E_2$ then $A_{\bar{\imath}}(p_2) \in X_{\bar{\imath}}$, and if also $A_{\bar{\imath}}(p) = A_{\bar{\imath}}(p_2)$ then $p \in W \uplus P_2$ and $A_i(p) = A_i(p_2)$.

A.2.5 Pure bigraphs — theory

[Definition 9.1 (s-category of pure concrete bigraphs)] (R) The s-category $\text{'B}\textsc{ig}(\mathcal{K})$ of pure concrete bigraphs over a signature $\mathcal{K}$ has interfaces $I = \langle m, X \rangle$ as objects, with origin $\epsilon = \langle 0, \emptyset \rangle$, and bigraphs $G : I \to J$ as arrows. I is called the *inner face* of G, and J the *outer face*. The support set, $|G|$, of G is the disjoint union of the sets of nodes and edges of G. If $F : J \to K$ is another bigraph with $|F| \cap |G| = \emptyset$, then their composition is defined directly in terms of the compositions of the constituents as follows:

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

The identities are $\langle \mathrm{id}_m, \mathrm{id}_X \rangle : I \to I$, where $I = \langle m, X \rangle$. The sub-s-category $\text{'B}\textsc{ig}_{\mathrm{h}}$ consists of *hard* bigraphs, those with place graphs in $\text{'P}\textsc{lg}_{\mathrm{h}}$.

We leave the definition of the tensor product separate as we will reference it directly.

[Definition 9.3 (tensor product)] The *tensor product* of two bigraph interfaces is defined by $\langle m, X\rangle \otimes \langle n, Y\rangle \stackrel{\text{def}}{=} \langle m+n, X \uplus Y\rangle$ when X and Y are disjoint. The *tensor product* of two bigraphs $G_i : I_i \to J_i (i = 0, 1)$ is defined by

$$G_0 \otimes G_1 \stackrel{\text{def}}{=} \langle G_0^{\mathrm{P}} \otimes G_1^{\mathrm{P}}, G_0^{\mathrm{L}} \otimes G_1^{\mathrm{L}}\rangle : I_0 \otimes I_1 \to J_0 \otimes J_1$$

when the interfaces exist and the node and edge sets are disjoint. This combination is well-formed, since its constituents share the same node set.

[Theorem 9.4 (bigraphs are wide)] (R) $\acute{\text{B}}\text{IG}(\mathcal{K})$ *and* $\acute{\text{B}}\text{IG}_{\text{h}}(\mathcal{K})$ *are wide s-categories. The interface* $I = \langle n, X\rangle$ *has* $\text{width}(I) = n$*, and for* $G : \langle m, X\rangle \to \langle n, Y\rangle$ *the width map* $\text{width}(G)$ *sends each site* $i \in m$ *to the unique root* $j \in n$ *such that* $i <_G j$.

Lower case letters $a, b, \ldots$ are used for *ground* bigraphs, those with inner face ϵ and $a : \epsilon \to I$ can be written as $a : I$.

[Corollary 9.6 (RPOs for bigraphs)] *In both* $\acute{\text{B}}\text{IG}$ *and* $\acute{\text{B}}\text{IG}_{\text{h}}$ *an RPO for* $\vec{A}$ *to* $\vec{D}$ *is provided by the triple*

$$(\langle B_0^{\mathrm{P}}, B_0^{\mathrm{L}}\rangle, \langle B_1^{\mathrm{P}}, B_1^{\mathrm{L}}\rangle, \langle B^{\mathrm{P}}, B^{\mathrm{L}}\rangle)$$

where $(\vec{B}^{\mathrm{P}}, B^{\mathrm{P}})$ *is a place graph RPO for* $\vec{A}^{\mathrm{P}}$ *to* $\vec{D}^{\mathrm{P}}$ *and* $(\vec{B}^{\mathrm{L}}, B^{\mathrm{L}})$ *is a link graph RPO for* $\vec{A}^{\mathrm{L}}$ *to* $\vec{D}^{\mathrm{L}}$.

A bigraph is called *lean* if its link graph is lean *i.e.* has no idle edges. A^E denotes the result of adding a set E of fresh idle edges to A.

[Definition 9.12 (Abstract pure bigraphs and their category)] Two concrete bigraphs A and B are *lean-support equivalent*, written $A \mathrel{\widehat{\simeq}} B$, if after discarding any idle edges they are support equivalent. The category $\text{BIG}(\mathcal{K})$ of *abstract pure bigraphs* has the same objects as $\acute{\text{B}}\text{IG}(\mathcal{K})$, and its arrows are lean-support equivalence classes of concrete bigraphs. Lean-support equivalence is clearly a congruence (Definition 3.5). The associated quotient functor, assured by Definition 3.6, is

$$[\![\cdot]\!] : \acute{\text{B}}\text{IG}(\mathcal{K}) \to \text{BIG}(\mathcal{K}).$$

The definition of BIG_{h} is analogous, with the restriction of $[\![\cdot]\!]$ to $\acute{\text{B}}\text{IG}_{\text{h}}$ as quotient functor.

[Terminology (wirings)] A bigraph with interfaces of zero width (and hence having no nodes) is called a *wiring.*

[Terminology (discreteness)] A bigraph is *discrete* if it has no edges, and its link map is bijective. The link map is then defined as $link : X \uplus P \to Y$ and $|X| + |P| = |Y|$. This means that every point is open (*i.e.* is mapped to a name), no two points are peers, and no name is idle.

[Notation] We often omit $\ldots \otimes \mathrm{id}_I$ in compositions, when there is no ambiguity; for example, given $G : \langle m, \emptyset \rangle \to \langle n, X \rangle$ and $\mathit{merge} : \langle m, \emptyset \rangle \to \langle 1, \emptyset \rangle$, we write $\mathit{merge} \circ G$ for $(\mathit{merge} \otimes \mathrm{id}_X) \circ G$.

Given a wiring $\omega : \langle 0, Y \rangle \to \langle 0, Z \rangle$ we may restrict its link map to any subset $X \subseteq Y$, yielding the *restricted* wiring $\omega \restriction X : \langle 0, X \rangle \to \langle 0, Z \rangle$. Then, if the outer face of G is $\langle m, X \rangle$ we may write simply ωG for $\omega \restriction X \otimes \mathrm{id}_m) \circ G$.

[Definition 9.13 (parallel product)] The *parallel product* of two bigraphs is defined on interfaces by $\langle m, X \rangle \| \langle n, Y \rangle \stackrel{\text{def}}{=} \langle m+n, X \cup Y \rangle$, and on bigraphs by

$$G_0 \parallel G_1 \stackrel{\text{def}}{=} \langle G_0^{\mathrm{P}} \otimes G_1^{\mathrm{P}}, G_0^{\mathrm{L}} | G_1^{\mathrm{L}} \rangle : I_0 \otimes I_1 \to J_0 \parallel J_1$$

when the interfaces exist and the node sets are disjoint[14].

[Definition 9.14 (parallel product)] *Let $G_0 \parallel G_1$ be defined. Then*

$$G_0 \parallel G_1 = \sigma(G_0 \otimes \tau G_1),$$

where the substitutions σ and τ are defined as follows: If z_i $(i \in n)$ are the names 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 9.15 (prime product)] The *prime product* of two interfaces is given by

$$\langle m, X \rangle \| \langle n, Y \rangle \stackrel{\text{def}}{=} \langle 1, X \cup Y \rangle.$$

For two bigraphs $\vec{P} : \vec{I} \to \vec{J}$, if $I_0 \otimes I_1$ defined and n is the sum of the widths of J_0 and J_1, we define their *prime product* by

$$P_0 \,|\, P_1 \stackrel{\text{def}}{=} \mathit{merge}_n \circ (P_0 \parallel P_1) : I_0 \otimes I_1 \to J_0 \otimes J_1.$$

[Proposition 9.16 (underlying discrete bigraph)] *Every bigraph G in* ´Big *or* ´Big$_{\mathrm{h}}$ *can be expressed uniquely (up to iso) as $G = (\omega \otimes \mathrm{id}_n) \circ D$, where ω is a wiring and D is discrete.*

The proof of Corollary 3.6 is identical to the proof below and so we reproduce the original here.

[Proposition 11.6 (preserving RPOs)] *The forgetful functor $\mathcal{U}$ preserves RPOs[15]; this is, if $(\vec{C}, C)$ is a binding RPO for $\vec{A}$ to $\vec{D}$ then $(\vec{C}, C)^{\mathrm{u}}$ is a pure RPO for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$.*

[14] Note that the edge sets are not required to be disjoint. This is because the parallel product on link graphs unions the link maps of both link graphs, not requiring the edge sets to be disjoint.

[15] This functor $\mathcal{U}$: ´Bbg($\mathcal{K}$) → ´Big($\mathcal{K}$) was from an s-category of binding bigraphs to an s-category of pure bigraphs. Replace the word 'binding' with 'kind' for the required proof.

Proof Assume the binding RPO $(\vec{C}, C)$ with interface J. Let $(\vec{B}, B)$, with interface I, be the binding RPO built for $\vec{A}$ to $\vec{D}$ by [7, Construction 11.4]. Then, since RPOs are unique up to isomorphism, there is a mediating iso $\iota : I \to J$ between these two RPOs. Also from the construction we know that $(\vec{B}, B)^{\mathrm{u}}$ is a pure RPO for $\vec{A}^{\mathrm{u}}$ to $\vec{D}^{\mathrm{u}}$, and we have a mediating iso $\iota^{\mathrm{u}} : I^{\mathrm{u}} \to J^{\mathrm{u}}$ between this RPO and the relative bound $(\vec{C}, C)^{\mathrm{u}}$. But isomorphism preserves the RPO property, so $(\vec{C}, C)^{\mathrm{u}}$ is also a pure RPO.

B Stronger kind rules

We now present an alternative definition for the notion of 'kind', using a stronger set of conditions on what kinds of nodes a place can be an ancestor of. It is more restrictive than the definition used throughout this report.

STRONGER KIND RULES:

SK1 if $r = \mathit{prnt}(v)$ then $\mathit{ctrl}(v) \cup \mathit{kind}(\mathit{ctrl}(v)) \subseteq C_r$

SK2 if $r = \mathit{prnt}(s)$ then $C_s \subseteq C_r$

SK3 if $v = \mathit{prnt}(v')$ then $\mathit{ctrl}(v') \cup \mathit{kind}(\mathit{ctrl}(v')) \subseteq \mathit{kind}(\mathit{ctrl}(v))$

SK4 if $v = \mathit{prnt}(s)$ then $C_s \subseteq \mathit{kind}(\mathit{ctrl}(v))$

where r is a root, s a site, and v and v' are nodes.

These stronger conditions can be described as follows, where p is a root or node and w is a node or site. The stronger kind rules state:

> If a place p is the parent of a place w then it must be able to contain any control that w can contain. If w is a node then p must be able to contain the control of w.

Another way to picture the structure of strong kind place graphs is as follows. Let $P_G = V_G \uplus \{s_0, \ldots, s_{m-1}\} \uplus \{r_0, \ldots, r_{n-1}\}$ be the set of nodes, sites and roots of a bigraph G. P_G is the set of places of G. We can provide a mapping $\mathit{contains} : P_G \to \mathcal{P}(\mathcal{K})$ from the places of G to the subsets of $\mathcal{K}$ that the places can contain. The mapping is defined by

$$\mathit{contains}(i) = C_i \text{ where } i \text{ is a root or site}$$

$$\mathit{contains}(v) = \mathit{ctrl}(v) \cup \mathit{kind}(v) \text{ where } v \text{ is a node.}$$

The mapping has the property that if $p <_G p'$ then $\mathit{contains}(p) \subseteq \mathit{contains}(p')$ and so is a monotonic function between the partially ordered sets (P_G, prnt_G)[16] and $(\mathcal{P}(\mathcal{K}), \subseteq)$. We illustrate this in the example below.

Example B.1. *Let the signature $\mathcal{K} = \{K_1, K_2, K_3, K_4\}$. Let kind be defined as:*

$$\begin{array}{lcl} K_1 & \mapsto & \{K_1, K_2, K_3\} \\ K_2 & \mapsto & \emptyset \\ K_3 & \mapsto & \{K_2, K_4\} \\ K_4 & \mapsto & \emptyset \end{array}$$

We will not specify the function ar. Let

$$G : \langle 2, (\{K_1, K_2\}\{K_2\}) \rangle \to \langle 1, (\{K_1, K_2, K_3, K_4\}) \rangle$$

[16] We abuse notation here as prnt_G is a function not a relation but the intended meaning should be clear.

be a strong kind place graph in $\acute{\mathrm{K}}\mathrm{PG}(\mathcal{K})$*, depicted in Figure 29. Let* $ctrl_G$ *be defined as:*

$$v_0 \mapsto K_1, v_1 \mapsto K_3, v_2 \mapsto K_2, v_3 \mapsto K_2, v_4 \mapsto K_2.$$

G and $contains(G)$ *are shown below. We depict the prnt graph using directed arcs instead of edges to emphasise the ordering. We omit the obvious mapping* $G \to contains(G)$ *for clarity.*

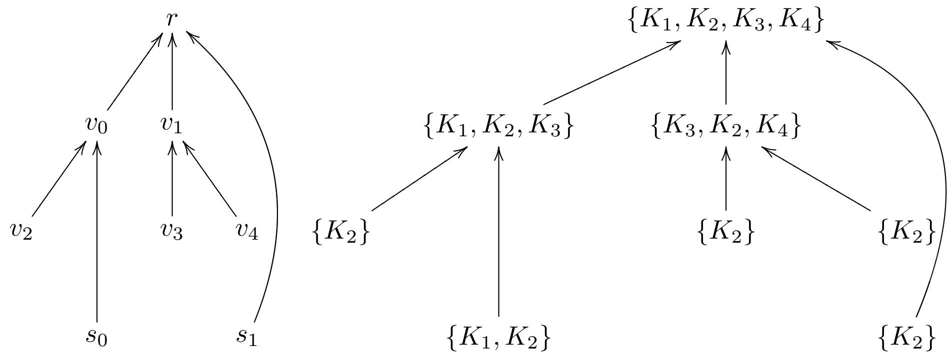


Figure 29: A strong kind bigraph $G : \langle 2, (\{K_1, K_2\}\{K_2\})\rangle \to \langle 1, (\{K_1, K_2, K_3, K_4\})\rangle$ and the poset $contains(G)$.

It should be noted that if such rules were used then the definition of signature should be altered. Consider this situation:

> We have a kind signature with a set $\mathcal{K}$ of controls. K and K' are elements of $\mathcal{K}$, $K' \in kind(K), K \in kind(K')$ and $kind(K) \subset kind(K')$.

Even though the signature specifies that a node of control K can contain a node of control K', any place graph where a node of control K is a parent of a node of control K' breaks SK3 as $kind(K')$ is a strict superset of $kind(K)$. In other words, a sensible definition of such a 'stronger' kind signature would state that if K can contain K' and K' can contain K'' then K can contain K'' *i.e.* that the containment relation is transitive. As an aside, the author finds that it helps to imagine the containment relation as a directed graph with $\mathcal{K}$ as the set of vertices and if there is an arc from K to K' then we say that K can contain K'. The set $kind(K)$ is then the set of reachable vertices from K.

Stronger kind bigraphs would be of interest as any parametric reaction rules based on them would be able to specify which controls could never be an ancestor of a site in the redex or reactum of the rules. It is conceivable that the above set of stronger kind rules satisfies all of the theorems in this report but this has not been investigated.

Finally, we speculate on a possible use of strong kindness. Let G be a strong kind bigraph with a site s. Let S be the set of controls that s may contain. Say we have a prime reaction rule (R, R') where the root r of R can contain the set T of controls. If $T \nsubseteq S$ then in any composition $G \circ F$, no redex of the reaction rule will occur under the parent of site s. This information could be used to 'tag' the parent of s as not being able to contain a redex for certain rules. This may be useful in some program which tries to identify redexes in a bigraph and the scenario presented here could obviously be generalised to the case of wide reaction rules.

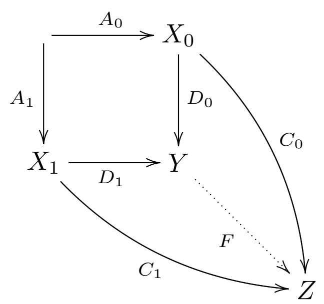
C Miscellaneous proofs

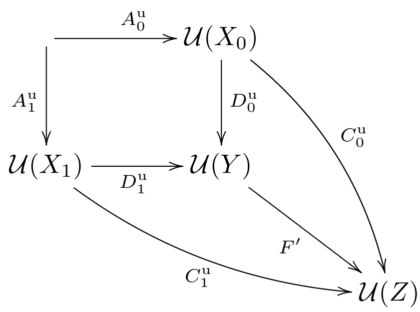
The following is a proof of Lemma 4.3 for tile-sorted link graphs. Less detailed arguments can be given for many of the cases but we present the original proof as similar arguments may be applicable to other link sortings.

Lemma C.1 (the Hammer lemma). *Let $\vec{D} : \vec{X} \to Y$ and $\vec{C} : \vec{X} \to Z$ be two bounds for $\vec{A} :\to \vec{X}$ in* $\textrm{'L}\text{IG}(\Sigma)$ *such that each $y \in Y$ is not idle in at least one of (B_0, B_1).*

If there exists a mediator $F' : \mathcal{U}(Y) \to \mathcal{U}(Z)$ such that the diagram on the right commutes, then there exists an $F : Y \to Z$ such that $\mathcal{U}(F) = F'$ and the diagram on the left commutes.

Further, if F' is unique as a mediating arrow between $\vec{D}$ and $\vec{C}$ then F is also a unique mediating arrow between $\vec{D}$ and $\vec{C}$.





Proof. F has the same *link* map as F'. We will break the proof into cases on the parent map of F and show in each case that tile sorting is obeyed.

We will not annotate points with directions or a until we need to assume that they are (un)directed. We use $\mathsf{ts}(G)$ to denote that a link graph G obeys tile sorting.

The following assumption is crucial to our proof:

(I) Each $y \in Y$ is not idle in at least one of (D_0, D_1).

We first prove that the edges in F obey the tile sorting.

- Let $F^{-1}(e_1) = \{p_1, p_2, \ldots, p_n\}$. Then $C_i^{-1}(e_1) = \{p_1, p_2, \ldots, p_n\}$. $\mathsf{ts}(C_i)$ and so $\mathsf{ts}(F)$ for this case.
- Let $F^{-1}(e_1) = \{y_1\}$. Whether y is directed or not does not affect whether $\mathsf{ts}(F)$.
- Let $F^{-1}(e_1) = \{y_1, y_2\}$.
  - If y_1 and y_2 are undirected, then $\mathsf{ts}(F)$.
  - Let y_1 be directed *i.e.* y_1^{d}. Since $\mathsf{ts}(D_0)$, $D_0^{-1}(y_1^{\mathsf{d}}) = \{q_0^{\mathsf{d}}\}$. Similarly for D_1.
    * Let y_2 also be directed *i.e.* $y_1^{\mathsf{d}'}$. Since $\mathsf{ts}(D_0)$, $D_0^{-1}(y_2^{\mathsf{d}'}) = \{q_0^{\mathsf{d}'}\}$. We now have $C_0(q_0^{\mathsf{d}}) = e_1 = C_0(q_0^{\mathsf{d}'})$. Since $\mathsf{ts}(C_0)$, d' is the opposite of d and so $\mathsf{ts}(F)$ for this case.

* Let y_2 be undirected *i.e.* y_2^{a}. Given **(I)**, assume that y_2^{a} is not idle in D_0 w.l.o.g. Since $\mathsf{ts}(D_0)$, there exists some q_0^{a} such that $D_0(q_0^{\mathsf{a}}) = y_2^{\mathsf{a}}$. We then have $C_0(q_0^{\mathsf{d}}) = e_1 = C_0(q_0^{\mathsf{a}})$ but this breaks the sorting in C_0 and so this subcase never arises.

- Let $F^{-1}(e_1) = \{y_1, p_1\} \uplus Y' \uplus P'$ where $\{y_1\} \uplus Y' \subseteq Y$, $\{p_1\} \uplus P' \subseteq P_1$. For any $p \in \{p_1\} \uplus P'$, since $F(p) = e_1$, $C_0(p) = e_1 = C_1(p)$.

  – Let y_1 be directed *i.e.* y_1^{d}. From $\mathsf{ts}(D_i)$, there exists a q_i^{d} such that $D_i(q_i^{\mathsf{d}}) = y_1^{\mathsf{d}}$ and so $C_i(q_i^{\mathsf{d}}) = e_1$. Since $C_i(q_i^{\mathsf{d}}) = e_1 = C_i(p_1)$ and $\mathsf{ts}(C_i)$, p_1 must have opposite direction o *i.e.* p_1^{o}. If $Y' \uplus P' = \emptyset$ then $\mathsf{ts}(F)$ for this case.

  Let $p \in P'$. Then e_1 has at least three preimages in C_i, two of which are directed. This breaks the sorting in C_i and so this case never occurs.

  Let $y \in Y'$. If y is directed then e_1 has at least three preimages in C_i, all of which are directed. This breaks the sorting in C_i and so this case never occurs. Let y be undirected. Then by **(I)**, y has a preimage q in w.l.o.g. D_0 and $C_0(q) = e_1$. But then in C_0, e_1 has three preimages q, q_i^{d}, and p_1^{o}, two of which are directed. This breaks the sorting in C_i and so this case never occurs.

  – Let y_1 be undirected *i.e.* y_1^{a}. Given **(I)**, assume that y_1^{a} is not idle in D_0 w.l.o.g. It must then have a non-empty set of undirected preimages by $\mathsf{ts}(D_0)$. Let $D_0(q_0^{\mathsf{a}}) = y_1^{\mathsf{a}}$. Then $C_0(q_0^{\mathsf{a}}) = e_1 = C_0(p)$ for all $p \in \{p_1\} \uplus P'$ and so by $\mathsf{ts}(C_0)$, all points in $p \in \{p_1\} \uplus P'$ must be undirected. If $Y' = \emptyset$ then $\mathsf{ts}(F)$ for this case.

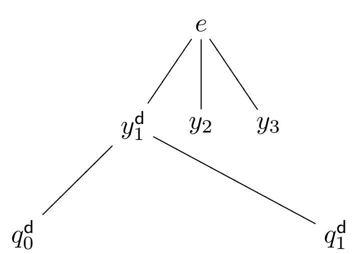
  Otherwise, given **(I)**assume that $y \in Y'$ is not idle in D_i. It must then have a non-empty set of preimages by $\mathsf{ts}(D_i)$. Let $D_i(q) = y$. Then $C_i(q) = e_1 = C_i(p_1^{\mathsf{a}})$ and so y must be undirected. Hence, all elements of $\{y_1, p_1\} \uplus Y' \uplus P'$ are undirected and $\mathsf{ts}(F)$ for this case.

- Let $F^{-1}(e_1) = \{y_1, y_2, y_3\} \uplus Y'$ where $\{y_1, y_2, y_3\} \uplus Y' \subseteq Y$.

  – If all elements in the preimage of e_1 under F are undirected, then Ξ holds for F.

  – Let y_1 have direction d, so y_1^{d}. We will show that this case never arises.

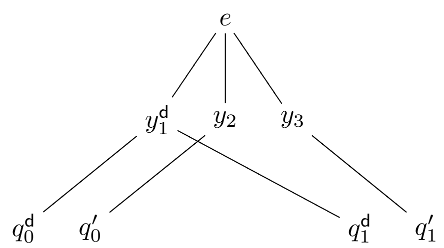
  Since $\mathsf{ts}(D_0)$, there exists exactly one point q_0^{d} such that $D_0(q_0^{\mathsf{d}}) = y_1^{\mathsf{d}}$. Similarly for D_1, so we have the diagram below.



  Now, since $(D_0^{\mathrm{u}}, D_1^{\mathrm{u}})$ is a pushout for $(A_0^{\mathrm{u}}, A_1^{\mathrm{u}})$, no name of Y is idle in *both* D_0 and D_1. If y_2 and y_3 are both non-idle in D_0, say, then

C_0 maps three points, one directed, to e breaking the sorting. Thus, w.l.o.g, y_2 must be idle in D_1 and y_3 idle in D_0. Now assume $Y' \supset \emptyset$ and $y' \in Y'$. Then y' must be non-idle in one of D_0 or D_1 but this would break the sorting in C_0 or C_1 for the reason above. Thus, $Y' = \emptyset$.

Say that y_2 has q'_0 in its preimage under D_0 and that y_3 has q'_1 in its preimage under D_1. We now have the following diagram.



As $C_0(q_0^{\mathsf{d}}) = e_1 = C_0(q'_0)$, q'_0 must have direction o and so must y_2 *i.e.* ${q'}_0^{\mathsf{o}}, y_2^{\mathsf{o}}$. Likewise, we have ${q'}_1^{\mathsf{o}}$ and y_3^{o}.

${q'}_0^{\mathsf{o}}$ is either a port or a name in X_0. Similarly for ${q'}_1^{\mathsf{o}}$.

Say ${q'}_0^{\mathsf{o}} \in X_0$. As ${q'}_0^{\mathsf{o}}$ is directed, it must have a preimage q^{o} under A_0. Then $(D_0 \circ A_0)(q^{\mathsf{o}}) = y_2^{\mathsf{o}}$. But as y_2^{o} is idle in D_1, y_2^{o} is not in the image of $(D_1 \circ A_1)$ which is equal to $(D_0 \circ A_0)$. Thus, ${q'}_0^{\mathsf{o}}$ is not directed and we have reached a contradiction. Similarly for ${q'}_1^{\mathsf{o}}$.

So, ${q'}_0^{\mathsf{o}}$ is a port in D_0 and ${q'}_1^{\mathsf{o}}$ is a port in D_1.

Now, by the pure link graph IPO construction, the ports of D_0 are the unshared ports of A_1. Further, $y_2 = (D_0 \circ A_0)({q'}_0^{\mathsf{o}}) = (D_1 \circ A_1)({q'}_0^{\mathsf{o}}) = D_1(A_1({q'}_0^{\mathsf{o}}))$. But y_2 is not in the image of $link_{D_1}$. We have reached a final contradiction.

Thus, this case can never occur.

We now prove that the names in F obey the tile sorting.

- Let $F^{-1}(z) = \{p_1, p_2, \ldots, p_n\}$. Then $C_i^{-1}(z) = \{p_1, p_2, \ldots, p_n\}$. $\mathsf{ts}(C_i)$ and so $\mathsf{ts}(F)$ for this case.

- Let $F^{-1}(z) = \{y\}$. If y is directed *i.e.* y^{d} then by $\mathsf{ts}(D_0)$ it has a preimage x_0^{d} in D_0. Hence, $C_0(x_0^{\mathsf{d}}) = z$ and so by $\mathsf{ts}(C_0)$, z has direction d. Hence, $\mathsf{ts}(F)$.

- Let $F^{-1}(z) = \{y_1, y_2, \ldots, y_n\}, n > 1$.
  - If $y_1, y_2, \ldots, y_n$ are all undirected, then $\mathsf{ts}(F)$.
  - Let y_1 be directed *i.e.* y_1^{d}. Since $\mathsf{ts}(D_0)$ and $\mathsf{ts}(D_1)$, $D_0^{-1}(y_1^{\mathsf{d}}) = \{q_0^{\mathsf{d}}\}$ and $D_1^{-1}(y_1^{\mathsf{d}}) = \{q_1^{\mathsf{d}}\}$.

    Given **(I)**, assume that y_2 is not idle in D_0 w.l.o.g. It must then have a non-empty set of preimages by $\mathsf{ts}(D_0)$. Let $D_0(q'_0) = y_2$. Then $C_0(q'_0) = z = C_0(q_0^{\mathsf{d}})$ but since q_0^{d} is directed, this breaks the sorting for C_0. Thus, this case can never occur.

- Let $F^{-1}(z) = \{y_1, p_1\} \uplus Y' \uplus P'$ where $\{y_1\} \uplus Y' \subseteq Y$, $\{p_1\} \uplus P' \subseteq P_1$. For any $p \in \{p_1\} \uplus P'$, since $F(p) = z$, $C_0(p) = z = C_1(p)$.

– If all the preimages of z are undirected then $\mathsf{ts}(F)$.

– Let p_1 be directed *i.e.* p_1^{d}. By **(I)**, assume that y_1 is not idle in D_0 w.l.o.g. It must then have a non-empty set of preimages by $\mathsf{ts}(D_0)$. Let $D_0(q_0') = y_1$. Then $C_0(q_0') = z = C_0(p_1^{\mathsf{d}})$ but since p_1^{d} is directed, this breaks the sorting for C_0. Thus, this case can never occur.

– Let y_1 be directed *i.e.* y_1^{d}. By $\mathsf{ts}(D_0)$, $D_0(q_0^{\mathsf{d}}) = y_1^{\mathsf{d}}$ for some x_0^{d}. Then $C_0(q_0^d) = z = C_0(p_1)$ but since q_0^{d} is directed, this breaks the sorting for C_0. Thus, this case can never occur.

Hence, F is tile sorted and $\mathcal{U}(F) = F'$.

If F' is unique as a mediating arrow then, since $\mathcal{U}$ is faithful, F' is also unique as a mediating arrow. □

The above proof is not very pleasant but we can gain some intuition from it. The proof rests on the fact that

(**I**) Each $y \in Y$ is not idle in at least one of (D_0, D_1).

This property is assured by the RPO and IPO constructions. But why is that? It is related to the intuition that a coequaliser *"is the least destructive identification necessary to force an equation to be true on the equivalence classes"* [1]. This notion of 'least' underlies colimits and in particular, it guarantees that a colimit has no 'spare bits'. For example, the coequaliser $h : B \to C$ of two functions $f, g : A \to B$ in the category of sets has the property that all elements of C are in the image of h *i.e.* there are no 'spare' elements.

Pushouts can be constructed from sums and coequalisers and again have this property of no spare bits. For example, the pushout $\vec{g} : \vec{B} \to C$ of a pair $\vec{f} : A \to \vec{B}$ in the category of sets has the property that all elements of C are in the image of one or both of g_0, g_1. This property is analogous to the property the RPOs and IPOs have in pure bigraphs. Indeed, the interface of an RPO is constructed by taking a disjoint sum and then forming the quotient over an equivalence relation. The link map restricted to the elements of the sum is then formed by mapping each element onto its equivalence class — this process is also how one may construct a pushout in the category of sets.

This intuition is not new — it was mentioned by Leifer [8] that RPOs and IPOs are related to pushouts via slice categories. However, the same intuition also guided the author towards realising that the RPO interface in a kind RPO is fitting — an intuition which directed many of the subsequent proofs.

References

[1] Michael Barr and Charles Wells. *Category Theory for Computing Science.* Les Publications CRM, Montréal, third edition, 1999.

[2] Luís Caires. *A Model for Declarative Programming and Specification with Concurrency and Mobility.* PhD thesis, University of Lisbon, 1999.

[3] 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.

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

[5] Lars Birkedal et al. Delca project model.

[6] 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.

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

[8] James J. Leifer. *Operational congruences for reactive systems.* PhD thesis, Computer Laboratory, University of Cambridge, 2001. Available in revised form as Technical Report 521,Computer Laboratory, University of Cambridge, 2001, from http://pauillac.inria.fr/∼leifer/research.html.

[9] 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.

[10] Mícheál Mac an Airchinnigh. The confederation of the isles, 2002. Ongoing example for course material and exam questions available from http://www.cs.tcd.ie/Micheal.MacanAirchinnigh/Web Pages/IRISH_VDM/WRITINGS/PROJECTS/isles.pdf.

[11] Robin Milner. Bigraphical reactive systems: basic theory. Technical Report UCAM-CL-TR-523, Computer Laboratory, University of Cambridge, 2001.

[12] Robin Milner. Bigraphs whose names have multiple locality. Technical Report UCAM-CL-TR-603, Computer Laboratory, University of Cambridge, September 2004.

[13] Robin Milner. Local bigraphs, confluence and λ-calculus. Technical Report to appear, Computer Laboratory, University of Cambridge, 2005.

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