Technical Report

UCAM-CL-TR-570
ISSN 1476-2986

Number 570

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Bigraphs and mobile processes

Ole Høgh Jensen, Robin Milner

July 2003

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

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

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

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

Series editor: Markus Kuhn

ISSN 1476-2986

Bigraphs and mobile processes

Ole Høgh Jensen
BRICS, Ålborg University
Department of Computer Science

Robin Milner
University of Cambridge
Computer Laboratory

July 2003

Abstract: A *bigraphical reactive system* (BRS) involves *bigraphs*, in which the nesting of nodes represents locality, independently of the edges connecting them; it also allows bigraphs to reconfigure themselves. BRSs aim to provide a uniform way to model spatially distributed systems that both compute and communicate. In this memorandum we develop their static and dynamic theory.

In Part I we illustrate bigraphs in action, and show how they correspond to to process calculi. We then develop the abstract (non-graphical) notion of *wide reactive system* (WRS), of which BRSs are an instance. Starting from reaction rules —often called rewriting rules— we use the RPO theory of Leifer and Milner to derive (labelled) transition systems for WRSs, in a way that leads automatically to behavioural congruences.

In Part II we develop bigraphs and BRSs formally. The theory is based directly on graphs, not on syntax. Key results in the static theory are that sufficient RPOs exist (enabling the results of Part I to be applied), that parallel combinators familiar from process calculi may be defined, and that a complete algebraic theory exists at least for pure bigraphs (those without binding). Key aspects in the dynamic theory —the BRSs— are the definition of parametric reaction rules that may replicate or discard parameters, and the full application of the behavioural theory of Part I.

In Part III we introduce a special class: the *simple* BRSs. These admit encodings of many process calculi, including the π-calculus and the ambient calculus. A still narrower class, the *basic* BRSs, admits an easy characterisation of our derived transition systems. We exploit this in a case study for an asynchronous π-calculus. We show that structural congruence of process terms corresponds to equality of the representing bigraphs, and that classical strong bisimilarity corresponds to bisimilarity of bigraphs. At the end, we explore several directions for further work.

CONTENTS

PAGE

Part I : Illustrations and Mathematical Framework 5

Part II : Bigraphical Reactive Systems 37

Part III : Specialisation and Application 81

Part I
Illustrations and Mathematical Framework

The introduction to Part I provides a rationale for bigraphs, an account of the work that leads up to bigraphs, and a synopsis of the whole memorandum. We continue with illustrations of bigraphs themselves, how they may reconfigure, and how they correspond to process calculi.

We then present the categorical framework in which the theory of bigraphs will be developed; this includes the notion of a well-supported precategory and the properties of *relative pushouts* (RPOs). We introduce an abstract notion of dynamic system called a *wide reactive system* (WRS); it is not graphical, but gives prominence to spatial extension, or *width* as we shall call it. This allows us to develop important aspects of structure and behaviour, which we shall apply to BRSs in Part II. In particular a WRS has parametric reaction (rewriting) rules; in terms of these, we define labelled transition systems for a wide range of WRSs and prove behavioural congruence theorems for them.

Thus Part I provides a mathematical frame within which both BRSs (in Part II) and their applications (in Part III) can be developed.

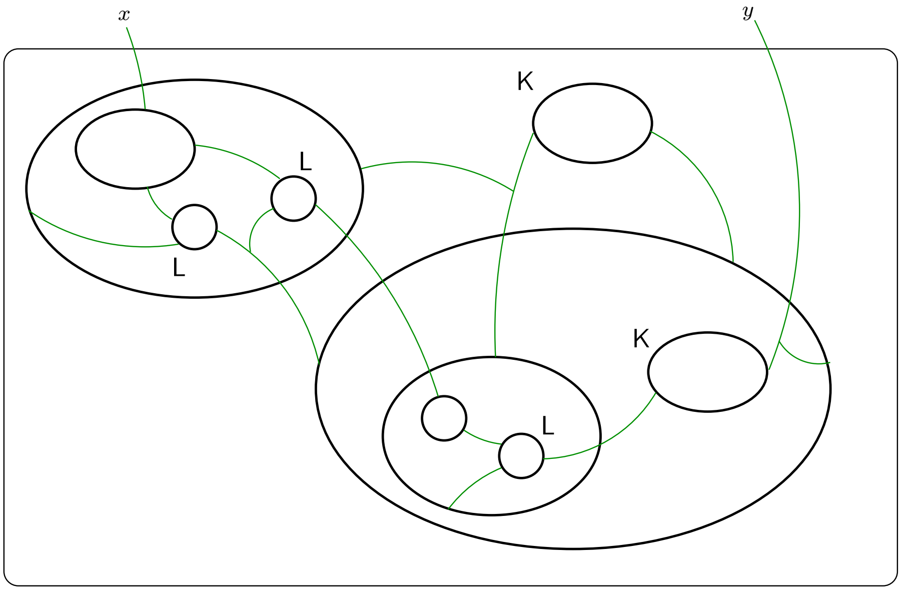


Figure 1: An example of a bigraph

1 Introduction

Bigraphical reactive systems (BRSs) [27, 28, 29, 20] are a graphical model of computation in which both *locality* and *connectivity* are prominent. Recognising the increasingly topographical quality of global computing, they take up the challenge to base all distributed computation on graphical structure. A typical bigraph is shown in Figure 1. Such a graph is reconfigurable, and its nodes (the ovals and circles) may represent a great variety of computational objects: a physical location, an administrative region, a data constructor, a π-calculus input guard, an ambient, a cryptographic key, a message, a replicator, and so on.

Bigraphs are a development of action calculi [25], but simpler. They use ideas from many sources: the Chemical Abstract machine (Cham) of Berry and Boudol [2], the π-calculus of Milner, Parrow and Walker [30], the interaction nets of Lafont [21], the mobile ambients of Cardelli and Gordon [7], the explicit fusions of Gardner and Wischik [16] developed from the fusion calculus of Parrow and Victor [32], Nomadic Pict by Wojciechowski and Sewell [40], and the uniform approach to a behavioural theory for reactive systems of Leifer and Milner [23]. This memorandum is self-contained; it builds on preliminary definitions and results put forward by Milner [28], but the approach here is a lot simpler and developed more fully.

The theory of BRSs responds to twin challenges: from application, and from existing process theory. The former demands greater breadth of concepts, while the latter demands continuity of ideas. We now discuss these challenges separately.

The challenge from applications

The long-term aim of this work is to provide a model of computation on a global scale, as represented by the Internet and the Worldwide Web. The aim is not just to build a mathematical model in which we can analyse systems that already exist. Beyond that, we seek a theory to guide the specification, design and programming of these systems, to guide future adaptations of them, and not to deteriorate when these adaptations are implemented. There is much talk of the vanishing ubiquitous computer of the future, which will obtrude less and less visibly in our lives, but will pervade them more and more. Technology will enable us to create this. To speak crudely, we must make sure that we understand it before it vanishes.

This will only be achieved if we can reverse the typical order of events, in which design and implementation come first, modelling later (or never). For example, a programming language is rarely based thoroughly upon a theoretical model. This has inevitably meant that our initial understanding of designed systems is brittle, and deteriorates seriously as they are adapted. We believe that the only acceptable solution, in the long run, is for system designs to be expressed with the concepts and notations of a theory rich enough to admit all that the designers wish.

The arrival of ubiquitous mobile computing provides an opportunity for this, simply because it is new enough for its languages and implementation techniques not to be entrenched. Another reason is that concurrency theorists have anticipated mobility and have some structures to offer for new languages. Thus designers and analysts may come to speak the same tongue. For example the π-calculus model is beginning to be adopted by business process management to provide languages and analytical tools for business processes with mobile structure [39].

Whatever our optimism, we cannot expect to arrive immediately at the right model. Initially we have to strike a compromise between incremental development of existing ideas on the one hand, and making too large a leap on the other hand. For if a model is to be seriously used in design, then it must be somewhat complex; it must grasp enough of the complexity of real systems to allow us to assess whether we are on the right track. If we tackle each aspect of global computation in isolation from the others we may develop an elegant theory, but it may not survive when other aspects are taken into account. Yet to tackle all aspects at once will defeat us.

So our strategy here is to tackle just two aspects —*mobile connectivity* and *mobile locality*— simultaneously. In fact this combination contains a novel challenge: to what extent in a model should connectivity and locality be interdependent? In plain words, does *where you are* affect *whom you can talk to*? To a user of the Internet there is total independence, and we want to model the Internet at a high level, in the way it appears to users. But to the engineer these remote communications are not atomic, but represented by chains of interactions between neighbours, and we should also provide a low-level model which reflects this reality. So we want to have it both ways; furthermore, we want to be able to describe rigorously how the high-level model is *realised* by the low-level one.

Of these two models, the low-level is the less novel. Indeed, von Neumann's cellular automata are the original paradigm for it; his agents were arranged on a fixed rectangular grid and interaction could only occur between neighbours. But in such a model we can *realise* a higher-level one in which a single agent is represented by different cells at different moments, and may send messages to other distant agents. So the challenge we address here is to provide the means to make locality and connectivity as dependent —or independent— as you wish. This seems to require new mathematical structures, and bigraphs represent our attempt to provide them.

In defining the bigraph model we are concerned not to ignore familiar calculi of mobile processes, which deal with interaction and mobility in a variety of different ways. Instead, we want a theory that can be specialised to each of these calculi, and therefore unifies them. This leads naturally to the second of our twin challenges.

The challenge from process theory

Existing process calculi have made great progress with interconnected concurrent processes [4, 1, 18, 26], with processes having mobile connectivity [30, 13] and with processes having notions of spatial location and mobility [2, 7]. There is some agreement among all these approaches, both in their basic notions and in their theories; perhaps the strongest feature is a good understanding of behavioural specification and equivalence. At the same time the space of possible calculi is large, we lack a uniform development of their theories, and there is no settled way to combine their various kinds of mobility. In particular, as shown by Castellani's [8] comprehensive survey, widely varying notions of locality have been explored.

The bigraphical model aims at further generality both in the treatment of mobility and in behavioural theory. As far as mobility is concerned, its notion of spatial region is akin to the ambients of Cardelli and Gordon [7]; we find that it supports both mobility in physical space (or in analogous organisational spaces, which may be virtual), and the dynamical control structures found in more traditional process calculi. The way this works is explained at some length, with examples, in Section 2. Here, we turn to the quest for a uniform behavioural theory.

It is common to present the *dynamics* of processes by means of *reactions* (typically known as rewriting rules) of the form $a \longrightarrow\!\!\triangleright\, a'$, where a and a' are agents. In the context of process calculi this treatment is typically refined somehow into *labelled transitions* of the form $a \xrightarrow{\ell}\!\!\triangleright\, a'$, where the label ℓ is drawn from some vocabulary expressing the possible interactions between an agent and its environment. These transitions have the great advantage that they support the definition of behavioural preorders and equivalences, such as traces, failures and bisimilarity. But the extension to labelled transitions is tailored for each calculus.

We therefore ask whether these labels can be *derived* uniformly from any given set of reaction rules of the form $r \longrightarrow\!\!\triangleright\, r'$, where r is an agent that may change its state to r'. A natural approach is to take the labels to be a certain class of (environmental) *contexts*; if L is such a context, the transition $a \xrightarrow{L}\!\!\triangleright\, a'$ implies that a reaction can occur in $L \circ a$ leading to a new state a'. (As we shall see, bigraphical agents and contexts live in a category, or more generally a precategory, where the composition $L \circ a$ represents the insertion of agent a in context L.) Moreover, we would like to be sure that the behavioural relations —such as bisimilarity— that are determined by the transitions are indeed congruential, i.e. preserved by insertion into any surrounding environment.

But we don't want *all* contexts as labels; as Sewell [37] points out, the behavioural equivalences that result from this choice are unsatisfactory. How to find a satisfactory —and suitably minimal— set of labels, and to do it uniformly, remained an open problem for many years. As a first step, Sewell [37] was able uniformly to derive satisfactory context-labelled transitions for parametric term-rewriting systems with parallel composition and blocking, and showed bisimilarity to be a congruence. It remained a problem to do it for reactive systems dealing with connectivity, which presents extra difficulty. Recently Leifer and Milner [23] were able to define minimal labels in terms of the categorical notion of *relative pushout* (RPO), and moreover to ensure that behavioural equivalence is a congruence for a wide class of reactive systems. These results were extended and refined in Leifer's PhD Dissertation [22], and Cattani et al [9] applied this theory to action graphs with rich connectivity. Meanwhile, Milner developed the bigraph model from action graphs, with inspiration from the mobile ambients of Cardelli and Gordon. The development was driven by the simplicity that comes from treating locality and connectivity independently, and was also inspired by Gardner's development [15] of *symmetric* action graphs (i.e. undirected edges).

In this memorandum, the technical thrust is towards a theory of bigraphs in which behaviour is uniformly represented by RPO-based transitions. The reader will soon see that, with this purpose in mind, several different paths could have been taken. We could therefore have proceeded more slowly, analysing each different option and its implications. Instead, we chose to follow one path far enough to provide evidence that a bigraphical approach will work; we have aimed at a point at which we can, within the new theory, recover some of the successes of existing theories such as the π-calculus. At the same time, we have tried to make it easier to explore different paths by dividing the theory —wherever possible— into independent topics. For example, *bigraphs* themselves are defined in terms of two independent structures, *place graphs* and *link graphs*, and each of these can be varied. Also, *bigraphical reactive systems* (BRSs) are defined as merely one instance of a general concept, *wide reactive systems* (WRSs), whose abstract theory we first develop; many other instances are possible.

Thus, making reasonable choices (which can be re-examined), we have taken the theory far enough to be able to set up within it a version of the π-calculus. Our main example is a finite asynchronous π-calculus, for which —as one member of a broad class— we are able to derive a transition system that corresponds closely to that defined in the classical approach; indeed, structural congruence in this π-calculus turns out to be represented by equality of bigraphs, and exactly the same bisimilarity congruence is achieved. In deriving this system we define general tools for the refinement of contextual transition systems, and comment on how we may tackle richer calculi, including ambient calculi, with the same approach.

Related work

We now turn to related work by other researchers, apart from those already mentioned.

The longest tradition in graph reconfiguration —often called graph-rewriting— is based upon the *double pushout* (DPO) construction originated by Ehrig [11]. Our use of (relative) pushouts to derive transitions is quite distinct from the DPO construction, whose purpose is to explain the anatomy of graph-rewriting rules (not labelled transitions) working in a category of graph embeddings with graphs as objects and embeddings as arrows. This contrasts with our contextual (pre)categories, where objects are interfaces and arrows are bigraphs. But there are links between these formulations, both via cospans [14] and via a categorical isomorphism between graph embeddings and a coslice over our contextual (pre)categories [9]. Ehrig [12] has recently investigated these links further, after discussion with the second author, and we believe that useful cross-fertilisation is possible.

In the paper just cited, Gadducci, Heckel and Llabrés Segura [14] represent graph-rewriting by 2-categories, whose 2-cells correspond to our reactions. Another use of 2-categories is by Sassone and Sobocinski [36]; they present an alternative way of deriving congruential bisimilarities in which 2-categories replace our precategories. This correspondence is under ongoing discussion; it appears to be very close. Thus the 2-categories will link our theory more closely to category-theoretic standards, while the corresponding precategories may continue to provide ease of manipulation.

Several other formulations of graph reconfiguration employ hypergraphs, for example Hirsch and Montanari [17]. In their model the hypergraphs are not nested, as bigraphs are; rewriting rules may replace a hyperedge by an arbitrary graph. Drewes, Hoffmann and Plump [10] deal with hierarchical graphs, but their links do not join graphs at different levels.

Synopsis

Part I In Section 2, as an illustration of bigraphs in action, it is shown how the dynamics of the π-calculus and (in less detail) the ambient calculus can be modelled in bigraphs. Then Section 3 sets up our category-theoretic framework, including the notion of relative pushout (RPO); in particular it introduces *supported* precategories, building upon work in Leifer's PhD thesis. Roughly speaking, in a precategory whose arrows are graphs or syntactic entities, *support* is a way of identifying each occurrence of a node or a subterm.

Supported precategories are then enriched to *wide* precategories, suitable for representing systems with distributed regions. On this basis, Section 4 defines the notion of *wide reactive systems* (WRS), equipped with parametric *reaction rules*; these are well illustrated by the examples of Section 2, even though we do not formulate bigraphs explicitly in Part I.

Section 5 shows how a (labelled) transition system (TS) can be uniformly derived in any WRS, using RPOs and their closely associated *idem pushouts* (IPOs), which are a kind of weak pushout. It also adapts the work of Leifer and Milner [23] to show that bisimilarity in such a TS with sufficient RPOs must be congruential. Using WRS functors (defined in Section 4), it is seen that these TSs and their congruential bisimilarities can be transmitted from one WRS to another, along a functor that is sufficiently well-behaved. One well-behaved functor is the *support quotient*, which forgets the identity of nodes and subterms.

The final topic in Part I is the notion of an *adequate* sub-TS; it allows a TS to be reduced while leaving bisimilarity unchanged. This will be important in Part III, to make certain applications tractable.

Part II In Section 6 the notion of a *pure bigraph* is formally defined in terms of its two constituents: a *place graph* and a *link graph*. These two notions, dealing respectively with locality and connectivity, are developed in Sections 7 and 8. In each of these two sections the crucial results are the theorem that RPOs always exist, and the characterisation of all the IPOs for a given pair of arrows.

In Section 9 the static theory of bigraphs is developed. A pure bigraph is an arrow in a supported precategory whose objects are interfaces; each interface consists of *places* for the place graphs and *points* for the link graphs. Several structural properties are introduced, especially RPOs and IPOs — whose characterisation is already provided by that for place graphs and link graphs respectively. The section provides a taxonomy of bigraphs, including the notions of *ion*, *atom* and *molecule* which are based upon a single control node. It also defines forms of parallel product close to the parallel composition operators of familiar process calculi.

The algebra of pure bigraphs is axiomatised and proved complete in Section 10. (This section is not required for any subsequent section.) In Section 11 pure bigraphs are enriched to *binding bigraphs* by adding binding names, along lines already set out in [28]. This relaxes the independency between placing and linking, by allowing certain names to have scope. The precategory of binding bigraphs has enriched interfaces, and its arrows are defined in terms of underlying pure bigraphs. Thus there is a forgetful functor from binding to pure bigraphs; the RPO theory for binding bigraphs is derived via this functor.

Finally in Section 11, with the addition of reaction rules, the central notion of a *bigraphical reactive system* (BRS) is defined. A BRS is seen to be a special case of WRS, as defined in Part I. Furthermore, because bigraphs have RPOs, the congruence results from Part I immediately apply. The work of Part I also tranfers these results to abstract BRSs, which are those most closely related to process calculi. It is shown that RPOs *do not exist* in abstract BRSs; that is why concrete BRSs —and indeed concrete WRSs and supported precategories— were introduced.

Part III In Section 13 the class of *simple* BRSs is introduced. These BRSs include models of both the π-calculus and the mobile ambient calculus. Their structural properties also ensure adequacy of a certain sub-TS, namely the *engaged* transitions; this eases the task of modelling the calculi mentioned.

Section 14 narrows the class of BRSs still further, to the *basic* ones. The purpose of this is to obtain a nice characterisation of the labels involved in modelling the asynchronous π-calculus; moreover we believe that a slight widening of the class of basic BRSs will embrace both the full π-calculus and mobile ambients.

In Section 15 this characterisation is specialised to a finite asynchronous π-calculus. It is then proved that the bisimilarity induced by this representation coincides with two standard congruences, strong bisimilarity and strong barbed bisimilarity. This provides the technical detail for work already presented by the authors at a conference [20]. It justifies the claim that bigraphical systems are consistent with previous work in process calculi, which has been one of the main purposes of the work reported here.

Finally, Section 16 explores several lines for further research.

2 Bigraphs in action

We introduce bigraphs informally, with examples showing the kinds of system that they represent, and the kind of mobility that they model. We also illustrate a simple term language for describing bigraphs. The examples allow us to explain how locality and connectivity co-operate; they also help to understand how bigraphs are naturally treated as arrows in a (pre)category whose objects are a simple kind of interface.

Figure 1 shows an uninterpreted example of a bigraph. It has *nodes* that support two kinds of structure; hence the term ‘bigraph’. First, nodes may occur inside other nodes, so a bigraph has depth; since a node represents locality we call this nesting structure of a bigraph its *place graph*. Second, nodes have *ports* that may be connected by *links*, represented here by thin lines which may fork; we call this linked structure of a bigraph, which is independent of locality, its *link graph*. To each node is assigned a *control*, such as K or L, which tells us what kind of node it is. Each control has an *arity*, a finite ordinal; for example, L has arity three, so each L-node has an ordered set of three ports, at each of which a link may impinge. It may impinge either from inside or from outside the node. The diagram also shows the use of *names* x and y; such names allow a bigraph to be linked into larger bigraph.

The place graph and the link graph share a node set, but are otherwise independent structures. The *dynamics* of bigraphs, i.e. the reconfigurations that may occur, depend upon both structural components; they are determined by one or more *reaction rules*. Each such rule has a *redex* and a *reactum*. The redex is a precondition for a reaction, represented by a pattern of nesting and linkage; the reactum is a postcondition indicating how the reaction will change that pattern. The places at which reactions may occur are determined by the controls. A control K may be specified as *atomic*, meaning that nothing may be nested within a K-node; if non-atomic it may also be specified as *active*, meaning that reactions may occur within a K-node. On the other hand if K is non-atomic but *passive*, then a K-node must be destroyed before its inhabitant nodes can react.

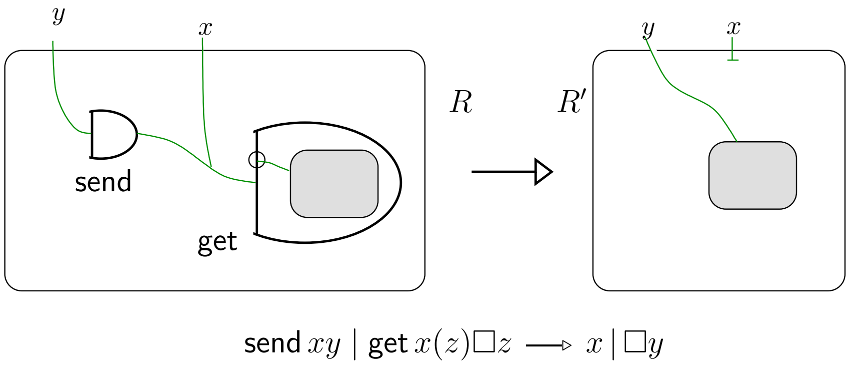
We now give some typical reaction rules. A reaction consists of the replacement of a redex occurring in a bigraph by the corresponding reactum; we shall see later how the notion of ‘occurrence’ is represented in a precategory of bigraphs.

Example 1 (reaction in the π-calculus) Our first example (Figure 2) represents the familiar reaction rule of the asynchronous π-calculus (without summation)

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

To present this reaction rule in terms of bigraphs we need two controls send and get, both with arity two. Recall that in the asynchronous π-calculus there are no output guards $\overline{x}y.(-)$ and reaction is forbidden inside the input guard $x(z).(-)$; to match this we declare send atomic, and get non-atomic but inactive.

The redex R of Figure 2 illustrates a feature of bigraphs that is absent in Figure 1; the notion of a *hole* — the grey box. This is a place where another bigraph may be inserted. Arcs may impinge at points upon the hole, which we call *inner names* of R; when another bigraph is inserted, its (outer) names are fused with these inner names. The inner names and ports in a bigraph are collectively called its *points*. A link may



y
x
send
get
R
R'
y
x
send xy | get $x(z)\square z \longrightarrow x \mid \square y$

Figure 2: Reaction rule for the asynchronous π-calculus

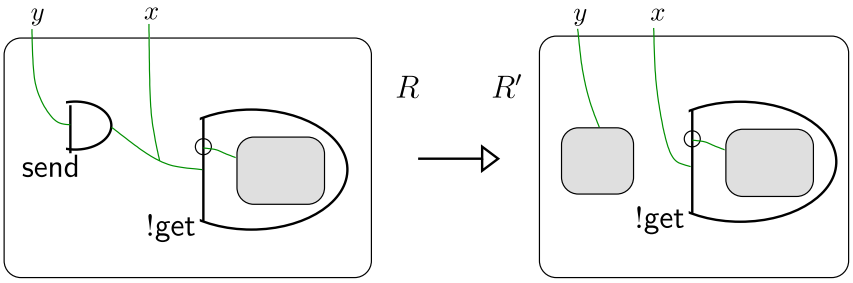
connect an arbitrary number of points, and may also (but need not) be given an outer name.

Here, the hole in R represents the parameter P of the π-calculus rule; the port on the hole represents the name z bound in P. In fact, every bigraph is parametric in general; it has both an *inner* interface I with its parameter(s), and an *outer* interface J indicating the kind of hole(s) in which it, in turn, may be placed. We shall call I and J respectively the *inner face* and *outer face* of R. The precategory of bigraphs will have interfaces as objects and parametric bigraphs such as $R : I \to J$ as arrows. Interfaces will be triples of the form $I = \langle m, \vec{k}, X \rangle$, where m is a *width* (i.e. the number of sites represented by I), $\vec{k}$ is a vector of length m indicating how many *local* names are associated with each site, and X is a set of *global* names.

In the present case R has an inner face $I = \langle 1, (1), \emptyset \rangle$; the width 1 tells us that there is only one hole in R, the singleton vector (1) tells us that this hole has one *local* name (corresponding to z in the π-calculus rule) associated with it, and the third component tells us that there are no global names involved at the inner face. Note that if I is an inner face of a bigraph R, as here, then its names (local or global) are the inner names of R. The outer face of R is $J = \langle 1, (0), \{xy\} \rangle$; again, the width is 1, telling us that R will occupy just one hole in some outer bigraph. The third component indicates that J has two global names x and y, also called global (outer) names of R.

In a bigraph, both (outer) names and ports can be *binding*. Here R has only free names; if it had a binding name then this would be a local name of the outer face J. Dually, a local inner name of the inner face I corresponds to a binding name of any parameter. (The parameter in this example really corresponds to the π-calculus abstraction $(z).P$.) We represent a binding occurrence of a point by a small circle; note that a port —such as the second port of get— may be binding, which means that it may only be connected to ports inside the node.

The reactum $R' : I \to J$ has the same inner face I as R, because the parameter persists through the reaction; it also has the same outer face J so that it may replace R in some outer context. The substitution $\{y/z\}$ in the π-calculus rule is represented just by an arc. The name x is unattached in R', because the two nodes have been discarded.



$$\mathsf{send}\,xy \mid \mathsf{!get}\,x(z)\square z \longrightarrow \square y \mid \mathsf{!get}\,x(z)\square z$$

Figure 3: Reaction rule for input replication in the asynchronous π-calculus

Turning to the term language, note how a local name (here z) is written in parentheses. Local names may be changed by alpha-conversion. Holes are squares. Note especially that the operation of juxtaposing two bigraphs, linking any edges with a name in common, is represented in a term by *parallel composition* '$\mid$'. The occurrence of x in the reactum $R' = x \mid \square y$ means that x, though unused, is part of the outer face of R'. Thus the correspondence between terms and bigraphs is quite accurate. ■

Example 2 (a π-calculus reaction rule for replication) In the previous rule, the parameter P of the π-calculus redex appears exactly once in the reactum; this is reflected in the bigraphical rule by the single occurrence of a hole in the reactum R', and by the fact that R and R' have the same inner face. But there is also a π-calculus rule called *replicated input*:

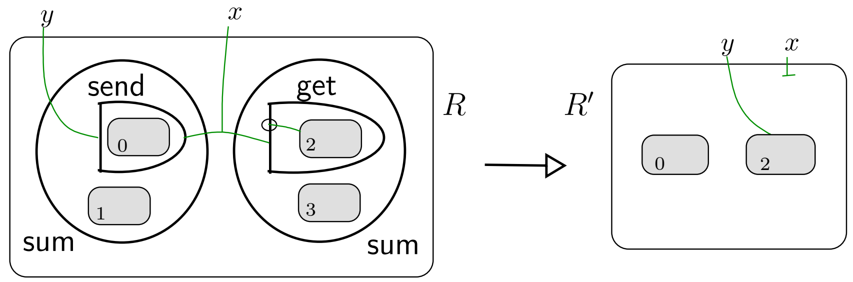
$$\overline{x}y \mid !x(z).P \longrightarrow \{y/z\}P \mid !x(z).P \ .$$

Here the parameter P is replicated; we can think of the input of y along x as triggering the creation of a copy of P to handle it. Figure 3 represents the rule bigraphically; note that it uses a different control !get, which is preserved in the reactum. Thus the reactum in this case has a different inner face of width two, namely $R' : I' \to J$ with $I' = \langle 2, (1,1), \emptyset \rangle$. ■

Example 3 (a π-calculus reaction rule for summation) Figure 4 shows the communication rule for a π-calculus with summation,

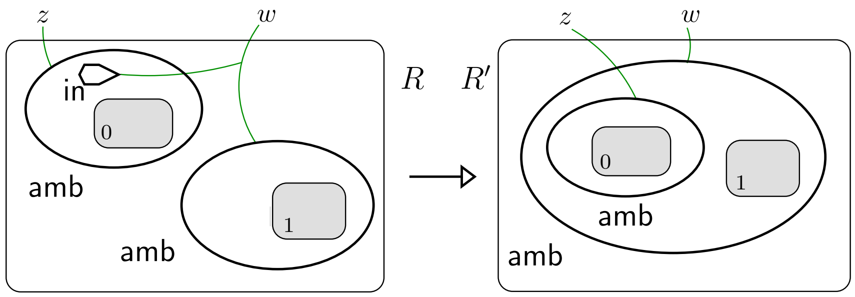
$$(M + \overline{x}y.P) \mid (N + x(z).Q) \to P \mid \{y/z\}Q \ ,$$

in which two of the parameters, M and N, are discarded. The controls send, get and sum are *passive*; this means that no reaction may occur inside nodes with these controls. Note that sum has arity 0; it serves to group together alternatives, only one of which will be enacted. ■



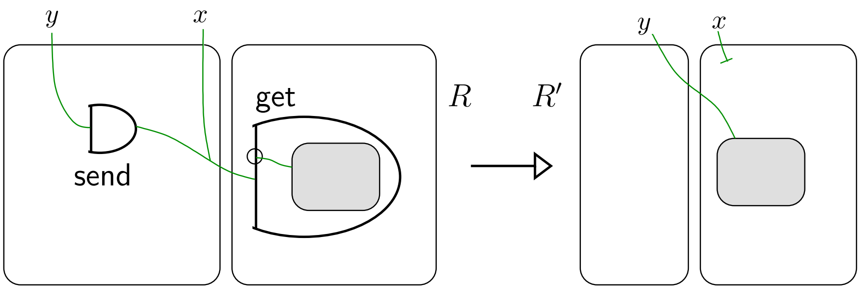
$$\mathsf{sum}(\mathsf{send}\,xy\,\square_0 \mid \square_1) \mid \mathsf{sum}(\mathsf{get}\,x(z)\,\square_2 z \mid \square_3) \longrightarrow x \mid \square_0 \mid \square_2 z$$

Figure 4: A reaction rule for the π-calculus with summation



$$\mathsf{amb}\,z(\mathsf{in}\,w \mid \square_0) \mid \mathsf{amb}\,w\square_1 \longrightarrow \mathsf{amb}\,w(\mathsf{amb}\,z\square_0 \mid \square_1)$$

Figure 5: Reaction rule for the ambient calculus



$$\mathsf{send}\,xy \parallel \mathsf{get}\,x(z)\Box z \longrightarrow 1 \parallel x \parallel \Box y$$

Figure 6: Global reaction rule for the π-calculus

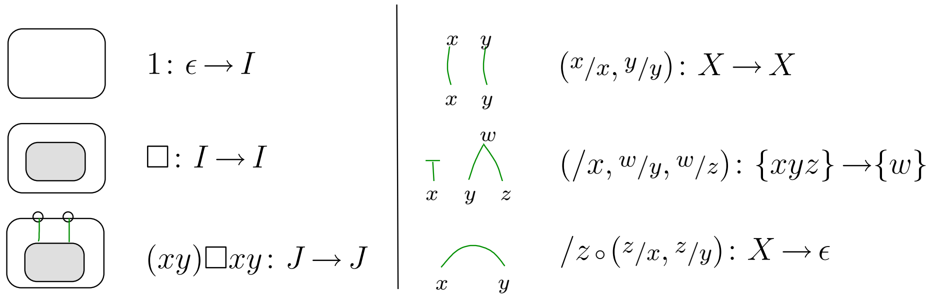
Example 4 (reaction in the ambient calculus) In the ambient calculus of Cardelli and Gordon [7], one of the primitive forms of reaction is the movement of one *ambient* into another. Figure 5 shows how bigraphs may represent such a rule. We use two controls, each with arity one: amb for an ambient, and in for a 'command' to move its parent ambient somewhere else. We declare in to be atomic; on the other hand we declare amb to be non-atomic and active, since ambients are intended only to localize activity, not to inhibit it.

The redex and reactum again have two global names z, w in their outer face, which has width 1; so this interface is $J = \langle 1, (0), \{zw\}\rangle$. The inner face now has two sites, so has width 2, and no names (local or global) associated with either site; so it is $I = \langle 2, (0, 0), \emptyset\rangle$.

The two parameters of this rule are, literally, passengers; their linkage (if any) plays no part in the reaction. However, as we shall see later, this does not prevent the two passengers —like passengers with mobile phones on a train— being linked to elsewhere, or even to each other. One can imagine interactions occurring between them independently of the occurrence of this ambient reaction. Our next example provides a possible instance of this. ■

In the preceding examples the reactions permitted are all *local*. For example, the ambient reaction rule will permit the ambient named x to enter the ambient named y only if these two ambients are neighbours — i.e. not separated by any control boundaries. Similarly, the first π-calculus rule requires the send and get nodes to be neighbours. But we may want to have in a more permissive rule which will allow action *at a distance*; in the case of the π-calculus this will mean that we can model the passing of a message in one step across arbitrarily many control boundaries. For this purpose the sender and receiver must be linked across region boundaries, as shown in the next example.

Example 5 (global reaction in the π-calculus) In the π-calculus reaction rule of Example 3 the redex has width 1; this means that the rule applies only when the send and get molecules are co-located. To allow a context to place them apart, we need only


$1\colon \epsilon \to I$
$\square\colon I \to I$
$(xy)\square xy\colon J \to J$
x
y
x
y
$({}^{x}/{x}, {}^{y}/{y})\colon X \to X$
w
x
y
z
$(/x, {}^{w}/{y}, {}^{w}/{z})\colon \{xyz\} \to \{w\}$
x
y
$/z \circ ({}^{z}/{x}, {}^{z}/{y})\colon X \to \epsilon$

(Interfaces: $I = \langle 1, (0), \emptyset \rangle$, $J = \langle 1, (2), \emptyset \rangle$, $X = \{xy\}$)

Figure 7: Some simple bigraphs

change the outer width of the redex and reactum to 2, shown in Figure 6; thus in this case we have $R, R' : \langle 1, (1), \emptyset \rangle \to \langle 2, (0, 0), \{xy\} \rangle$. Note that, in the term language, we have used ‘ ∥ ’ rather that ‘ | ’ for parallel composition; this combinator keeps regions separate but still merges links with a common global name.

Such ‘wide’ reaction rules are interesting in the presence of one or more active controls, because they can be used to separate the components of a distributed redex but still allow it to react. We have already introduced amb as an example of an active control. Indeed, our categorical representation will allow us to insert a bigraph with arbitrarily many global names in the double-width hole of the ambient rule’s redex. In particular, we might insert an instance of the redex of our remote π-calculus rule; by this means we would create two interwoven but independent redexes, such that neither reaction precludes the other. This is not an unlikely occurrence in the Internet, modelled at a suitable level of abstraction. ■

In our illustrations of reaction rules we chose to stay close to familiar calculi. Beyond these, the possibilities range widely. For example, using a combination of active and passive controls, various forms of *failure management* can be modelled. This may include the inactivation of processes due to failure, the reporting of failures, recovery procedures, and the subsequent re-activation of inactivated processes.

Our illustrations so far have emphasised dynamics. We should also realise that some bigraphs have no dynamic behaviour but are useful building blocks. Figure 7 shows six simple examples, together with the terms that denote them. On the left side, the first is just a region containing nothing. Its inner face is the so-called *origin* ϵ, the simplest possible interface where everything is null, while its outer face is the simplest interface I of width 1. The second is the categorical identity at I. The third is again an identity at an interface J of width 1, but here the site has two local names.

The three bigraphs on the right side of the figure will be called *wirings*; they have both interfaces of width 0, i.e. of the form $\langle 0, (), X \rangle$, which we abbreviate to X (a set of names). Their function is to link global inner and outer names in various patterns. The first wiring is just the identity on an interface $\{xy\}$; think of it as the identity *substitution* on these two names. The next involves a substitution of the name w for both the inner names y and z. This wiring also *closes* the inner name x; that is, when composed with another bigraph with name x, such as R in Example 3, it will remove x from the outer face. The last is an example of what Gardner and Wischik [16] call a *fusion*. It is like a substitution of z for x and y, but it also closes z.

This concludes our illustration of bigraphs. Our main purpose was to show how they can represent the dynamics of process calculi; we have also seen that even simple things like name closure and substitution are bigraphs. We wrote each bigraph as a term in a language that we shall not formalise here (this will be done in future work). These terms are a mildly sugared form of mathematical constructions that we shall introduce in later parts of the paper; we have shown them here to indicate that bigraphs are not far from a programming language — in which a programmer can define a wide variety of specialised reaction rules.

Discussion

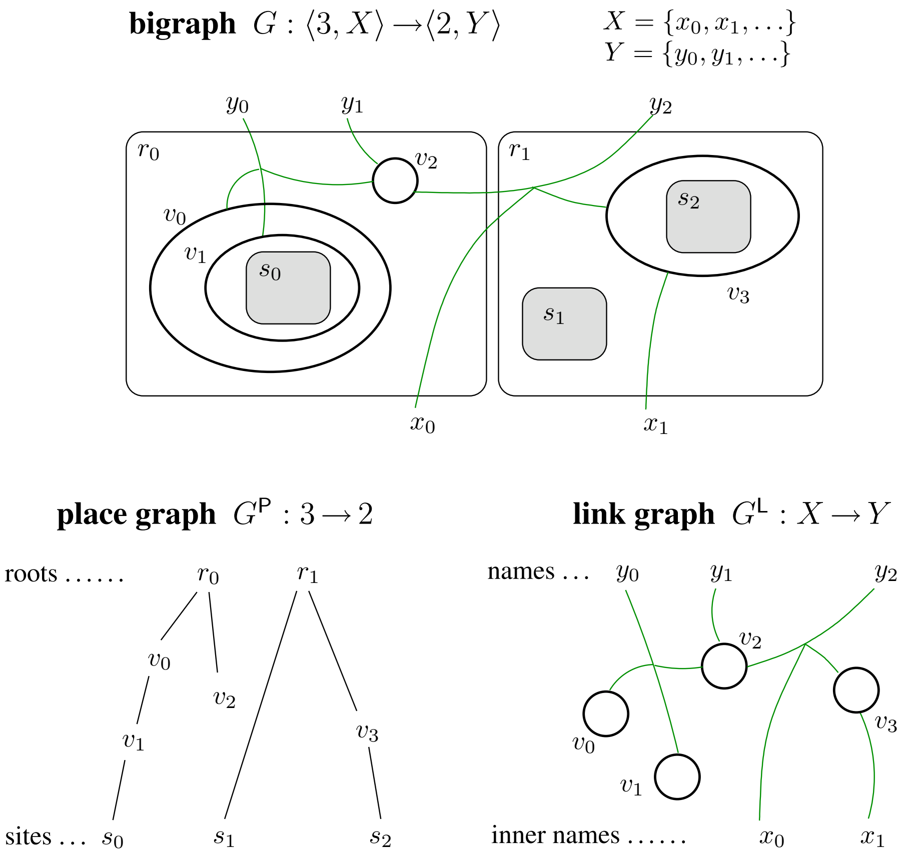
By means of several examples we have informally introduced what we shall call, in Part II, a *bigraphical reactive system* (BRS). Each BRS is based upon a precategory of agents and contexts built according to a *signature* that defines controls and their static properties, and a set of *reaction rules* that defines dynamics. The bigraphical theory of Part II will begin with a direct formulation of bigraphs, in the classical tradition of graph theory. As we have said, bigraphs will be the arrows of a precategory whose objects are interfaces.

The reader may ask why we go to the trouble of a graphical formulation, when —as we have illustrated in our examples— there is a rather pleasant algebraic formulation of them. Can we not develop this algebraic theory, and then consider bigraphs as just an alternative presentation of its elements?

There are two reasons for taking the graphs as primary. The first is that the space of mobile computing that we want to model has a strong topographical character — whether the topography is real or virtual— and it is reasonable to seek to model this directly.

The second reason is theoretically deeper. One of our main goals is to build a theory of dynamic systems embracing as much as possible of the behavioural theory embodied in process calculi. This is often based upon a (labelled) transition system, and we wish to apply the theory originated by Leifer and Milner [23], which defines such transition systems in terms of so-called *relative pushouts* (a weak form of pushout); this ensures that the resulting behavioural equivalences are congruential — provided that sufficient relative pushouts (RPOs) exist in the appropriate precategory of agents and contexts. But it turns out that neither the algebraic theory of bigraphs, nor their straightforward presentation as a category, possesses RPOs. This is because they do not cater for the notion of *occurrence* of one bigraph in another; they represent only *abstract* bigraphs, where the identity of nodes is absent. By moving to *concrete* bigraphs —formulated as a precategory rather than a category— we regain enough structure for the RPO theory to work, and can thereby gain a congruential behavioural theory.

Fortunately, when we quotient this theory to recover a category of abstract bigraphs, the quotiented behavioural theory is still congruential, and we thereby derive a uniform



bigraph $G : \langle 3, X\rangle \to \langle 2, Y\rangle$

$X = \{x_0, x_1, \ldots\}$
$Y = \{y_0, y_1, \ldots\}$

place graph $G^{\mathsf{P}} : 3 \to 2$

link graph $G^{\mathsf{L}} : X \to Y$

Figure 8: Resolving a pure bigraph into a place graph and a link graph

approach to behaviour that can be instantiated for different process calculi.

We shall formalise bigraphs directly in Part II. To manage their complexity we shall first consider *pure* bigraphs, those that have no local names; then in a later section we introduce local names and binding and define *binding* bigraphs. We represent a pure bigraph as a combination of two independent mathematical structures — a *place graph* and a *link graph*. Note that this *combination* is quite distinct from the categorical *composition* used to insert one bigraph into another (e.g. an agent into a context). But it is simply related to them; to compose two bigraphs categorically, we first resolve them into their respective place graphs and link graphs, then compose these, and finally combine the results into a new bigraph.

It is helpful to see an example in Figure 8 of how a pure bigraph G can be resolved into a place graph G^{P} representing locality, and a link graph G^{L} representing connectivity. (Controls are not shown in the diagram.) The nodes $v_0, \ldots, v_3$ are common to the two structures, which are otherwise independent. Note the bigraph's double interfaces $\langle 3, X\rangle \rightarrow \langle 2, Y\rangle$; there is no third component $\vec{k}$ here, because a pure bigraph has no local names. This interface combines the place graph interface $3 \rightarrow 2$ with the link graph interface $X \rightarrow Y$; nothing determines that the names y_0, y_1, y_2 'belong' to any particular region of the bigraph (= root of the place graph), nor that the inner names x_0, x_1 'belong' to any particular site.

Let us repeat: in a *pure* bigraph $G : \langle m, X\rangle \rightarrow \langle n, Y\rangle$ we admit no association between the names Y and the roots (regions) n, nor between the inner names X and the sites m. It is this dissociation that enables us to treat locality and connectivity independently, yielding a tractable theory. This theory can then be extended rather easily to binding bigraphs.

Part II ends with a section introducing the dynamic theory of bigraphs, and Part III goes on to specialise and apply this theory. But the foundation of this theory is laid in the abstract setting of *wide reactive systems* (WRSs), where the topographical element is reduced to a very simple categorical notion of *width*. The remainder of Part I is devoted entirely to these abstract dynamic systems.

3 Precategories and relative pushouts

In this section and the following one we develop a mathematical framework for the static and dynamic properties of bigraphs. There are several varieties of bigraph, and we wish to derive in an abstract setting as many definitions and properties as we can that will apply to all varieties. Sections 3 and 4 are an adaptation and extension of work started by Leifer and Milner [23], then further developed by Leifer in his PhD Dissertation [22] and by Milner [28]. These two sections closely follows Section 3 in the latter paper.

The reader can perfectly well study Part II and beyond, independently of Sections 3 and 4, provided he or she is willing to take their main results on trust and to refer back to important definitions from time to time.

The present section is concerned with the categorical framework and the important concepts, especially relative pushouts, that will underlie the treatment of dynamics in Section 4.

Notation We shall always accent the name of a precategory, as in $\acute{\mathbf{C}}$. We use '$\circ$', 'id' and '$\otimes$' for composition, identity and tensor product. We denote the domain I and codomain J of an arrow $f\colon I \to J$ by $\mathsf{dom}(f)$ and $\mathsf{cod}(f)$; the set of arrows from I to J, called a *homset*, is denoted by $\acute{\mathbf{C}}(I,J)$.

Id_S will denote the identity function on a set S, and $\emptyset_S$ the empty function from $\emptyset$ to S. We shall use $S \uplus T$ for union of sets S and T known or assumed to be disjoint, and $f \uplus g$ for union of functions 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. We assume a fixed representation of disjoint sums; 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 $\bigcup_{v\in V}(\{v\}\times P_v)$.

We write $f \restriction S$ or $R \restriction S$ for the restriction of a function f or relation R to the set S.

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

Definition 3.1 (precategory, functor) 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 $\mathsf{id} \circ f = f = f \circ \mathsf{id}$. In the equation $h\circ(g\circ f) = (h\circ g)\circ f$, the two sides are either equal or both undefined.

A *subprecategory* $\acute{\mathbf{D}}$ of $\acute{\mathbf{C}}$ is defined like a subcategory, with $g \circ f$ defined in $\acute{\mathbf{D}}$ exactly when defined in $\acute{\mathbf{C}}$. A *functor* $\mathcal{F} : \acute{\mathbf{D}} \to \acute{\mathbf{C}}$ between precategories is a total function on objects and on arrows that preserves identities and composition, in the sense that if $g\circ f$ is defined in $\acute{\mathbf{D}}$, then $\mathcal{F}(g)\circ\mathcal{F}(f) = \mathcal{F}(g\circ f)$ in $\acute{\mathbf{C}}$. ■

In general we shall use $I, J, K, \ldots$ to stand for objects and $f, g, h, \ldots$ for arrows. We shall extend category-theoretic concepts to precategories without comment when they are unambiguous. One concept which we now extend explicitly is that of a monoidal category:

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:

$$\begin{array}{ll}
(1) & f \otimes (g \otimes h) = (f \otimes g) \otimes h \\
(2) & (f_1 \otimes g_1) \circ (f_0 \otimes g_0) = (f_1 \circ f_0) \otimes (g_1 \circ g_0) \\
(3) & \gamma_{I,\epsilon} = \mathsf{id}_I \\
(4) & \gamma_{J,I} \circ \gamma_{I,J} = \mathsf{id}_{I \otimes J} \\
(5) & \gamma_{I,K} \circ (f \otimes g) = (g \otimes f) \circ \gamma_{H,J} \qquad (\text{for } f : H \to I,\ g : J \to K)\ . \qquad \blacksquare
\end{array}$$

'Strict' means that equation (1) holds exactly, as stated, not merely up to isomorphism; 'symmetric' refers to the symmetry isomorphisms satisfying equations (3)–(5). We shall omit 'strict' and 'symmetric' from now on, as we shall always assume these properties.

Why do we wish to work in precategories? In the Introduction we pointed out that the dynamic theory of bigraphs will require the existence of relative pushouts (RPOs). This means that we need to develop the theory first for *concrete* bigraphs, those in which nodes have identity; then we can transfer the results to *abstract* graphs by the quotient functor that forgets this identity. Precategories allow a direct presentation of concrete bigraphs; for we can stipulate that two bigraphs F and G may be composed to form $H = G \circ F$ only if their node sets are disjoint. We can think of this composition as as *keeping track* of nodes[1]; we can recover from H exactly which nodes come from F and which from G.

More generally, we are interested in monoidal precategories where the definedness of composition and of tensor product depends upon a *support* set associated with each arrow. In bigraphs the support of an arrow will be its node set. In general we assume support to be drawn from some unspecified infinite set. We now give a general definition of precategories $\acute{\mathbf{C}}$ with support; we continue to use this accented notation for them, dropping the accent only when we have a category.

Definition 3.3 (supported (monoidal) precategory) A precategory $\acute{\mathbf{C}}$ is *supported* if it has:

- for each arrow f, a finite set $|f|$ called its *support*, such that $|\mathsf{id}_I| = \emptyset$. The composition $g \circ f$ is defined iff $|g| \cap |f| = \emptyset$ and $\mathsf{dom}(g) = \mathsf{cod}(f)$; then $|g \circ f| = |g| \uplus |f|$.
- for any arrow $f : I \to J$ and any injective map ρ whose domain includes $|f|$, an

[1] Leifer's development [22] (see Chapter 7) made use of a special category $\mathsf{Track}(\acute{\mathbf{C}})$ to keep track of nodes in a precategory $\acute{\mathbf{C}}$. This allowed the theory of RPOs to be developed for categories rather than for precategories. However, it can be developed more succinctly if we stay with the latter.

arrow $\rho\bullet f : I \to J$ called a *support translation* of f such that

$$\begin{array}{ll}
(1) & \rho\bullet\mathsf{id}_I = \mathsf{id}_I \\
(2) & \rho\bullet(g \circ f) = \rho\bullet g \circ \rho\bullet f \\
(3) & \mathsf{Id}_{|f|}\bullet f = f \\
(4) & (\rho_1 \circ \rho_0)\bullet f = \rho_1\bullet(\rho_0\bullet f) \\
(5) & \rho\bullet f = (\rho \restriction |f|)\bullet f \\
(6) & |\rho\bullet f| = \rho(|f|)\ .
\end{array}$$

If $\acute{\mathbf{C}}$ is monoidal as a precategory, it is *monoidal* as a supported precategory if it also satisfies

$$(7)\ \ \rho\bullet(f \otimes g) = \rho\bullet f \otimes \rho\bullet g\ .$$

Each of these seven equations is required to hold only when both sides are defined. ■

Exercise Deduce condition (1) from conditions (5) and (3).

We now consider functors between supported precategories.

Definition 3.4 (support equivalence, supported functor) Let $\acute{\mathbf{A}}$ be a supported precategory. Two arrows $f, g : I \to J$ in $\acute{\mathbf{A}}$ are *support-equivalent*, written $f \doteq g$, if $\rho\bullet f = g$ for some support translation ρ. By Definition 3.3(5) and (6) this is an equivalence relation. If $\acute{\mathbf{B}}$ is another supported precategory, then a functor $\mathcal{F}\colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ is called *supported* if it preserves support equivalence, i.e. $f \doteq g \Rightarrow \mathcal{F}(f) \doteq \mathcal{F}(g)$. ■

When we no longer need to keep track of support we may use a quotient *category* (not just precategory). To define such quotients, we need the following notion:[2]

Definition 3.5 (static congruence) Let $\equiv$ be an equivalence defined on every homset of a supported precategory $\acute{\mathbf{C}}$. We say 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 *static* (*monoidal*) *congruence on* $\acute{\mathbf{C}}$ whenever it is preserved by (tensor product and) composition. ■

As an example of a simple static congruence on bigraphs, we may define $f \equiv f'$ to mean that f and f' are identical when all K-nodes are discarded, for some particular control K.

The most important example of a static congruence will be support equivalence ($\doteq$). But the following definition shows that any static congruence at least as coarse as support equivalence will yield a well-defined quotient category:

Definition 3.6 (quotient categories) Let $\acute{\mathbf{C}}$ be a supported precategory, and let $\equiv$ be a static congruence on $\acute{\mathbf{C}}$ that includes support equivalence, i.e. $\doteq\ \subseteq\ \equiv$. 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 $\acute{\mathbf{C}}$:

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

[2] We use the term *static* congruence to emphasize that these congruences depend only on static structure, in contrast with behavioural congruences —like bisimilarity— that depend upon the dynamics of a system.

In **C**, identities and composition (and tensor product when $\acute{\mathbf{C}}$ has it) are given by

$$\begin{aligned} \mathsf{id}_m &\overset{\text{def}}{=} [\mathsf{id}_m]_{\equiv} \\ [f]_{\equiv} \circ [g]_{\equiv} &\overset{\text{def}}{=} [f \circ g]_{\equiv} \\ [f]_{\equiv} \otimes [g]_{\equiv} &\overset{\text{def}}{=} [f \otimes g]_{\equiv} \ . \end{aligned}$$

By assigning empty support to every arrow we may also regard **C** as a supported precategory, so that $[\cdot]_{\equiv} : \acute{\mathbf{C}} \to \mathbf{C}$ is a special supported functor called the *$\equiv$-quotient functor* for $\acute{\mathbf{C}}$. ■

Note that the quotient does not affect objects; thus any tensor product on **C** may still be partial on objects. But composition is always well-defined in **C** because $f \equiv g$ implies $f \bumpeq g$, and so also is tensor product provided it is defined on the domains and codomains.

We often abbreviate $[\cdot]_{\bumpeq}$ to $[\cdot]$; we call it the *support quotient functor*. From the definition, clearly a coarser quotient $[\cdot]_{\equiv}$ exists whenever $\equiv$ is the least equivalence that includes an arbitrary static congruence $\equiv'$ as well as support equivalence. In Parts II and III we shall define two coarser quotient functors on bigraphs by this means.

We now turn to the notion of relative pushout (RPO), which is of crucial importance in defining labelled transitions in the following section.

Notation In what follows we shall frequently use $\vec{f}$ to denote a pair f_0, f_1 of arrows in a precategory. If, for example, the two arrows share a domain H and have codomains I_0, I_1 we write $\vec{f} : H \to \vec{I}$.

Definition 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 $g_0 \circ f_0 = g_1 \circ f_1$, then we say 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) In a precategory, let $\vec{g} : \vec{I} \to K$ be a bound for $\vec{f} : H \to \vec{I}$. A *bound for $\vec{f}$ relative to $\vec{g}$* is a triple $(\vec{h}, h)$ of arrows such that $\vec{h}$ is a bound for $\vec{f}$ and $h \circ h_i = g_i$ $(i = 0, 1)$. We may call the triple a *relative bound* when $\vec{g}$ is understood.

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

$$\begin{tikzcd} & {} & \\ & {} \arrow[u, "h"] & \\ {} \arrow[uur, "g_0"] \arrow[ur, "h_0"'] & & {} \arrow[uul, "g_1"'] \arrow[ul, "h_1"] \\ & {} \arrow[ul, "f_0"] \arrow[ur, "f_1"'] & \end{tikzcd}
\qquad
\begin{tikzcd} & {} & \\ & {} \arrow[u, "k"'] & \\ {} \arrow[uur, "g_0"] \arrow[ur, "k_0"] \arrow[r, "h_0"'] & {} \arrow[u, dashed, "j"'] \arrow[uu, bend left, "h"] & {} \arrow[uul, "g_1"'] \arrow[ul, "k_1"'] \arrow[l, "h_1"] \\ & {} \arrow[ul, "f_0"] \arrow[ur, "f_1"'] & \end{tikzcd}$$

We shall often omit the word ‘relative’; for example we may call $(\vec{h}, h)$ a bound (or RPO) for $\vec{f}$ to $\vec{g}$.

The more familiar notion, a pushout, is a bound $\vec{h}$ for $\vec{f}$ such that *for any* bound $\vec{g}$ there exists an h which makes the left-hand diagram commute. Thus a pushout is a *least* bound, while an RPO provides a *minimal* bound relative to a given bound $\vec{g}$. In bigraphs we shall find that RPOs exist in cases where there is no pushout; see the discussions following Constructions 7.11 and 8.12.

Now suppose that we can create an RPO $(\vec{h}, h)$ for $\vec{f}$ to $\vec{g}$; what happens if we try to iterate the construction? More precisely, is there an RPO for $\vec{f}$ to $\vec{h}$? The answer lies in the following important concept:

Definition 3.9 (idem pushout) In a precategory, if $\vec{f} : H \to \vec{I}$ is a pair of arrows with common domain, then a pair $\vec{h} : \vec{I} \to J$ is an *idem pushout* (*IPO*) for $\vec{f}$ if $(\vec{h}, \mathsf{id}_J)$ is an RPO for $\vec{f}$ to $\vec{h}$. ■

Then it turns out that the attempt to iterate the RPO construction will yield the *same* bound (up to isomorphism); intuitively, the minimal bound for $\vec{f}$ to any bound $\vec{g}$ is reached in just one step. This is a consequence of the first two parts of the following proposition, which summarises the essential properties of RPOs and IPOs on which our work relies. They are proved for categories in Leifer’s Dissertation [22] (see also Leifer and Milner [23]), and the proofs can be routinely adapted for precategories.[3]

Proposition 3.10 (properties of RPOs) *In any precategory* $\acute{\mathbf{A}}$*:*

(1) If an RPO for $\vec{f}$ to $\vec{g}$ exists, then it is unique up to isomorphism.

(2) If $(\vec{h}, h)$ is an RPO for $\vec{f}$ to $\vec{g}$, then $\vec{h}$ is an IPO for $\vec{f}$.

(3) If $\vec{h}$ is an IPO for $\vec{f}$, and an RPO exists for $\vec{f}$ to $h \circ h_0, h \circ h_1$, then the triple $(\vec{h}, h)$ is such an RPO.

(4) (IPO pasting) Suppose that the diagram below commutes, and that f_0, g_0 has an RPO to the pair $h_1 \circ h_0, f_2 \circ g_1$. Then

- *if the two squares are IPOs, so is the big rectangle;*
- *if the big rectangle and the left square are IPOs, so is the right square.*

$$\begin{tikzcd}
{} \arrow[r, "h_0"] & {} \arrow[r, "h_1"] & {} \\
{} \arrow[u, "f_0"] \arrow[r, "g_0"'] & {} \arrow[u, "f_1"] \arrow[r, "g_1"'] & {} \arrow[u, "f_2"']
\end{tikzcd}$$

(5) (IPO sliding) If $\acute{\mathbf{A}}$ is supported then IPOs are preserved by support translation; that is, if $\vec{g}$ is an IPO for $\vec{f}$ and ρ is any injective map whose domain includes the supports of $\vec{f}$ and $\vec{g}$, then $\rho \bullet \vec{g}$ is an IPO for $\rho \bullet \vec{f}$.

[3] This adaptation works for the definition of precategory in Definition 3.1, which is satisfied by our supported precategories.

4 Wide reactive systems

We now introduce a kind of dynamical system, of which bigraphs will be an instance. This section adapts and extends the work of Section 3.3 in [28].

In previous work [23, 22] a notion of reactive system was defined. In our present terms, this consisted of a supported precategory whose arrows are called *contexts*, including *agents* whose domain is the origin ϵ, together with a set of agent-pairs (r, r') called *reaction rules*, and a subprecategory of so-called *active* contexts. The reaction relation $\longrightarrow\!\!\!\triangleright$ between agents was taken to be the smallest such that $D \circ r \longrightarrow\!\!\!\triangleright D \circ r'$ for every active context D and reaction rule (r, r').

For such systems we uniformly derived labelled transitions based upon IPOs. Several behavioural preorders and equivalences based upon these transitions —including bisimilarity— were shown to be congruences, subject to two conditions: first, that sufficient RPOs exist in the precategory; second, that decomposition preserves activity — i.e. $D \circ C$ active implies both C and D active.

In subsequent work, sufficient RPOs were found in interesting cases (Leifer [22], Cattani et al [9]). In each of these cases the condition that decomposition preserves activity is also met, if we limit attention to contexts with a single hole. However, certain derived transition systems are unsatisfactory under this limitation, as Sewell [37] has pointed out with examples. Also, as we saw in Section 1, we wish to consider multi-hole bigraphical contexts — not only to represent parametric reaction rules but also to accommodate multiple or 'wide' agents, as in the remote π-calculus reaction rule in Example 5. There are other reasons for treating wide agents; for example, we would like to understand reactions that may occur between agents located arbitrarily far apart.

This gives rise to the possibility of contexts in which some sites may be active, i.e. may permit reaction to occur, but not others. The following definitions allow this. They lead to *wide* reactive systems, which refine the above notion of reactive system as little as necessary for that purpose. We shall also see that, if we specialise this new treatment to *narrow* contexts (those with unit width), we recover the original notion of reactive system.

In what follows we shall use **Nat**, the strict symmetric monoidal category whose objects are finite ordinals, and whose arrows are functions between them.

Definition 4.1 (wide precategory, wide functor) A *wide precategory* $\acute{\mathbf{A}}$ is a supported precategory equipped with a functor $\mathsf{width} : \acute{\mathbf{A}} \to \mathbf{Nat}$ invariant under support translation, and a distinguished object ϵ called the *origin* such that $\mathsf{width}(\epsilon) = 0$. Moreover, for each permutation π on the ordinal $\mathsf{width}(I)$ there is an isomorphism $\pi_I : I \to I$ in $\acute{\mathbf{A}}$ with $\mathsf{width}(\pi_I) = \pi$.

If $\acute{\mathbf{A}}$ —as a precategory— is monoidal with unit ϵ, and width preserves tensor product, then $\acute{\mathbf{A}}$ equipped with ϵ and width is a *wide monoidal precategory*.

The objects $I, J, \ldots$ of $\acute{\mathbf{A}}$ are called *interfaces*, and its arrows $A, B, \ldots$ are called *contexts*. The set of contexts from I to J, called as usual the *homset* of the pair (I, J), will be denoted by $\acute{\mathbf{A}}(I, J)$ or $I \to J$. We call I and J the *inner* and *outer faces* of this homset. Arrows in a homset $\mathsf{Gr}(I) \overset{\text{def}}{=} \epsilon \to I$ are called *ground* arrows; we let lower case letters $a, b, \ldots$ range over these, and abbreviate $a \colon \epsilon \to I$ to $a \colon I$.

A supported functor $\mathcal{F}\colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ is called *wide* if it preserves origin and width, i.e. (distinguishing elements of $\acute{\mathbf{B}}$ by a prime) $\epsilon' = \mathcal{F}(\epsilon)$ and $\mathsf{width}' \circ \mathcal{F} = \mathsf{width}$. ■

We shall define bigraphs as a wide precategory in the next section. Meanwhile, from our discussion in Section 1 it is easy to see that, in bigraphs, 'width' is concerned only with locality, not with connectivity; the width of a bigraphical interface $I = \langle m, \vec{k}, X\rangle$ is just its multiplicity m, and the width of a bigraph $G : \langle m, \vec{k}, X\rangle \to \langle n, \vec{\ell}, Y\rangle$ is the function mapping each site $s \in m$ to the unique region $r \in n$ that contains s. We here define width at the abstract level of wide precategories, but when specialised to bigraphs it will allow us to define exactly which sites of a bigraph permit reaction. The notion of *location* will help us to formulate this:

Definition 4.2 (location) A *location* of an interface I with width m is a subset $\lambda \subseteq m$. We denote by $\mathsf{loc}(I)$ the set of locations of I, i.e. the powerset of $\mathsf{width}(I)$.

The width function of a context $C : I \to J$ is extended to $\mathsf{loc}(I)$ by $\mathsf{width}(C)(\lambda) \stackrel{\text{def}}{=} \{\mathsf{width}(C)(i) \mid i \in \lambda\}$. The *offset* of λ by n is given by $n \dotplus \lambda \stackrel{\text{def}}{=} \{n + i \mid i \in \lambda\}$. ■

We are now ready to consider how to add dynamics to wide precategories. We shall define a reaction relation over ground arrows.

At the start of this section we spoke of reaction rules of the form (r, r'), a pair of agents (redex and reactum) in the same homset. This does not reflect our examples of reaction rules in the introduction, which were all *parametric*; they were pairs of the form

$$(R\colon I \to J,\ R'\colon I' \to J)$$

with, in general, different inner faces I and I'. The parameter for such a rule will be a ground arrow in $\mathsf{Gr}(I)$. In general the reactum R' will be composed with a *transform* of this parameter, in $\mathsf{Gr}(I')$. This is illustrated by Example 2, the replication rule which duplicates its parameter, and by Example 3 which discards parts of a parameter (which may of course have arbitrary width). So the following definition allows rules that arbitrarily transform their parameters. It also allows for the possibility that parameters are constrained to lie in some subset of $\mathsf{Gr}(I)$. Remarkably, the congruence theorem of this section holds without any constraint upon either the parameter set or the nature of parameter transformations.

Finally, since reactions are supposed to occur only in contexts that are active, the following definition introduces an *activity map* to determine the sites at which each context is active, and how this activity is treated by composition; this map is further explained in the discussion that follows the definition.

Definition 4.3 (wide reactive system) A *wide reactive system (WRS)* is a wide precategory $\acute{\mathbf{A}}$ equipped with a triple $(\mathsf{Par}, \mathsf{Reacts}, \mathsf{act})$, where

- For each I, the set $\mathsf{Par}(I) \subseteq \mathsf{Gr}(I)$ represents the *parameters* of reaction rules.
- Reacts is a set of *reaction rules* of the form (R, R', trans), with *redex* $R : I \to J$, *reactum* $R' : I' \to J$ and *transform* map $\mathsf{trans} : \mathsf{Par}(I) \to \mathsf{Gr}(I')$.

- For each I, J the *activity map* $\mathsf{act} : \acute{\mathbf{A}}(I, J) \to \mathsf{loc}(I)$ satisfies two properties:

$$\begin{array}{ll} (1) & \mathsf{act}(\mathsf{id}_I) = \mathsf{width}(I) \\ (2) & \mathsf{act}(D \circ C) = \mathsf{act}(C) \cap \mathsf{width}(C)^{-1}(\mathsf{act}(D)) \ . \end{array}$$

We require that $\mathsf{Par}(I)$ and Reacts are closed under support translation, that each trans preserves it, and that act respects it. We say C is *active at* i if $i \in \mathsf{act}(C)$; similarly C is *active at* λ if $\lambda \subseteq \mathsf{act}(C)$, and C is *active* if $\mathsf{act}(C) = \mathsf{width}(\mathsf{dom}(C))$.

The *reaction relation* $\longrightarrow\!\!\!\!\triangleright$ over ground arrows is defined as follows: $g \longrightarrow\!\!\!\!\triangleright g'$ iff there exist a reaction rule (R, R', trans), an active[4] context D and a parameter $d \in \mathsf{Par}(I)$ such that $g = D \circ R \circ d$ and $g' = D' \circ R' \circ d'$, where $D' \bumpeq D$ and $d' \bumpeq \mathsf{trans}(d)$.

For a *monoidal* WRS we require a third condition on act:

$$(3) \quad \mathsf{act}(C \otimes D) = \mathsf{act}(C) \mathbin{\dot{\cup}} (\mathsf{width}(\mathsf{dom}(C)) \dot{+} \mathsf{act}(D)) \ . \qquad \blacksquare$$

We shall usually denote this WRS just by $\acute{\mathbf{A}}$. Let us explain the activity conditions more fully. Condition (2) asserts that $D \circ C$ is active at i iff C is active at i and D is active at $\mathsf{width}(C)(i)$. If $\mathsf{width}(\mathsf{dom}(C)) = m$ then condition (3) asserts that $C \otimes D$ is active at i iff either $i < m$ and C is active at i or $i \geq m$ and D is active at $i - m$. We leave it to the reader to check that these conditions make sense — i.e. that they are consistent with the equations governing composition and tensor product.

In passing, suppose that we are only concerned with reaction in contexts D that have interfaces of unit width $1 = \{0\}$, so that $\mathsf{width}(D)(0) = 0$. Then D is *active* iff it is active at 0. Conditions (1) and (2) then amount to requiring that the active contexts form a subprecategory closed under decomposition. Thus, as promised, we have a proper generalisation of the conditions under which the original congruence theorems [22, 23] were proved.

Definition 4.3 ensures that all its ingredients are closed under support equivalence, allowing us in Definition 4.7 to divide $\acute{\mathbf{A}}$ by $\bumpeq$, forming a quotient WRS. The following is immediate:

Proposition 4.4 (support translation of reactions) *Reaction in a WRS is closed under support equivalence, i.e. if $g \bumpeq h$, $g' \bumpeq h'$ and $g \longrightarrow\!\!\!\!\triangleright g'$ then $h \longrightarrow\!\!\!\!\triangleright h'$.*

A natural notion of functor $\mathcal{F} \colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ between WRSs is one that preserves reaction. What this means is that all the ingredients that constitute a reaction in $\acute{\mathbf{B}}$ must be at least as generous as in $\acute{\mathbf{A}}$. The definition is as follows:

Definition 4.5 (WRS functor, sub-WRS) Let $\acute{\mathbf{A}}$ and $\acute{\mathbf{B}}$ be wide reactive systems. A wide functor $\mathcal{F} \colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ of wide precategories is a *WRS functor* from $\acute{\mathbf{A}}$ to $\acute{\mathbf{B}}$ if it

[4] Our definition requires D to be active at the whole width n of the codomain of the redex R. An alternative, more refined, approach is to equip a reaction rule with a fourth component λ, a location in n; then we can require only that D be active at λ, not at the whole of n. One can imagine reaction rules, like the one in Example 5 of width two, where we might wish only one part of the redex to lie at an active site. Everything that follows can be adapted to this refinement; we avoid it here only for the sake of simplicity.

preserves the extra components of a WRS, i.e. (distinguishing the components of $\acute{\mathbf{B}}$ by a prime):

$$\begin{array}{rcl} d \in \mathsf{Par}(I) & \Rightarrow & \mathcal{F}(d) \in \mathsf{Par}'(\mathcal{F}(I)) \\ (R, R', \mathsf{trans}) \in \mathsf{Reacts} & \Rightarrow & (\mathcal{F}(R), \mathcal{F}(R'), \mathsf{trans}') \in \mathsf{Reacts}' \\ & & \qquad \text{where } \mathcal{F} \circ \mathsf{trans} = \mathsf{trans}' \circ \mathcal{F} \\ \mathsf{act}(C) & \subseteq & \mathsf{act}'(\mathcal{F}(C)) \ . \end{array}$$

Call $\mathcal{F}$ *monoidal* if both $\acute{\mathbf{A}}$ and $\acute{\mathbf{B}}$ are monoidal and $\mathcal{F}$ preserves tensor product. If $\mathcal{F}$ is a (monoidal) inclusion functor then we call $\acute{\mathbf{A}}$ a (*monoidal*) *sub-WRS* of $\acute{\mathbf{B}}$. ■

Proposition 4.6 (WRS functors preserve reaction) *A WRS functor* $\mathcal{F} \colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ *preserves reaction, i.e. if* $g \rightarrowtriangle g'$ *in* $\acute{\mathbf{A}}$ *then* $\mathcal{F}(g) \rightarrowtriangle \mathcal{F}(g')$ *in* $\acute{\mathbf{B}}$.

Clearly WRSs and their functors form a category. An important example of a functor is the support quotient functor, extended to WRSs as follows:

Definition 4.7 (quotient WRS) Let $\acute{\mathbf{A}}$ be a wide reactive system. Then its *support quotient* wide reactive system is based upon the support quotient $\mathbf{A} = \acute{\mathbf{A}}/\overset{\frown}{=}$, with other ingredients defined as follows:

- the parameters are $[d]$, for each parameter d in $\acute{\mathbf{A}}$
- the reaction rules are $([R], [R'], \mathsf{trans})$, for each rule (R, R', trans) in $\acute{\mathbf{A}}$, where in $\mathbf{A}$ we define $\mathsf{trans}([d]) \overset{\text{def}}{=} [\mathsf{trans}(d)]$
- the active sites are given by $\mathsf{act}([D]) \overset{\text{def}}{=} \mathsf{act}(D)$. ■

Proposition 4.8 (quotient reflects reaction) *The support quotient functor* $[\cdot] \colon \acute{\mathbf{A}} \to \mathbf{A}$ *both preserves and reflects reaction, i.e.* $[g] \rightarrowtriangle [g']$ *in* $\mathbf{A}$ *iff* $g \rightarrowtriangle g'$ *in* $\acute{\mathbf{A}}$.

The quotient functor takes a *concrete* WRS, based on a precategory, to an *abstract* WRS based upon a category. In the next section we show how to obtain a behavioural congruence for an arbitrary concrete WRS $\acute{\mathbf{A}}$ with sufficient RPOs. The support quotient $\mathbf{A}$ of $\acute{\mathbf{A}}$ may not possess RPOs, but nevertheless the quotient functor allows us to derive a behavioural congruence for $\mathbf{A}$ also. This use of a concrete WRS with RPOs to supply a behavioural congruence for the corresponding abstract WRS was first represented by the *functorial reactive systems* of Leifer's thesis [22].

5 Wide transition systems

We now consider how to equip a WRS with a labelled transition system. This will comprise a subset of the ground arrows, called *agents*, together with a set of transitions of a form such as $a \xrightarrow{L} a'$, where a, a' are agents and L is a context with $L \circ a$ defined. Then bisimilarity is defined in the usual way, and we are interested in general conditions under which it will be a congruence.

The definition of labelled transitions of Leifer and Milner [23] was as follows: $a \xrightarrow{L} a'$ if there is a reaction rule (r, r') and an active context D for which (L, D) is an idem pushout (IPO) for (a, r) and $a' = D \circ r'$. We shall do something close to this, but with two refinements.

The first refinement is to equip a transition with information about locality. For an agent $a\colon I$, a transition of the form $a \xrightarrow{L} a'$ tells us the extra context $L\colon I \to J$ needed by a to create a redex, but does not specify *where* this completed redex occurs within $L \circ a$, i.e. at which location in J the reaction takes place. This makes a difference if J has more than unit width. It turns out that, to achieve congruence of bisimilarity, we must index each L-transition by a location in the outer face of L.

Let us illustrate this with a simple example involving bigraphs. We need only the superficial understanding of bigraphs supplied by the introduction.

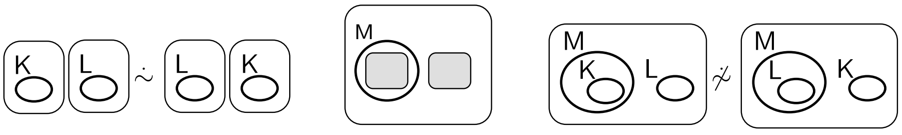
Example 6 (non-congruence) This example shows that bisimilarity based upon unlocated transitions, which we denote by $\dot\sim$, is not in general a congruence for bigraphical systems. Take the signature $\mathcal{K} = \{\mathsf{K}, \mathsf{L}, \mathsf{M}\}$, each with arity zero; let K, L be atomic and M non-atomic but passive. Ports, names and links are irrelevant in this example, so we take interfaces to be just finite ordinals (widths).

Now write $\mathsf{K}, \mathsf{L} : 0 \to 1$ for agents with a single atomic node, and $\mathsf{M} : 1 \to 1$ for the elementary passive context consisting of a single M-node. Let there be a single reaction rule (K, L); it allows the reaction $\mathsf{K} \longrightarrow \mathsf{L}$ in any active context.

Consider the two agents $a, b : 0 \to 2$ illustrated below, where $a = \mathsf{K} \otimes \mathsf{L}$ and $b = \mathsf{L} \otimes \mathsf{K}$. They can each do a transition that turns K into L, i.e. we have

$$a \xrightarrow{\mathsf{id}_2} \mathsf{L} \otimes \mathsf{L} \text{ and } b \xrightarrow{\mathsf{id}_2} \mathsf{L} \otimes \mathsf{L}\ .$$

Because these two transitions do not record the different places at which the reaction occurs, it turns out that $a \dot\sim b$.



$a \dot\sim b$ $\qquad$ $C \overset{\text{def}}{=} \mathsf{M} \mid \mathsf{id}_1$ $\qquad$ $C \circ a \not\dot\sim C \circ b$

Now put a and b in the context $C \overset{\text{def}}{=} \mathsf{M} \mid \mathsf{id}_1 \colon 2 \to 1$, as shown; then we find $C \circ a \not\dot\sim C \circ b$. In $C \circ b$ the K-node is active, so there is a transition $C \circ b \xrightarrow{\mathsf{id}_1}$; but $C \circ a$ has no such transition since its K-node is passive. ■

Our second refinement is concerned with the parametric nature of reaction. Each reaction is based on a redex R together with a parameter d. To reflect this in transitions, we adopt *pair-labels* of the form $L = (L^{\mathsf{red}}, L^{\mathsf{par}})$, where L^{red} arises from the redex and L^{par} from its parameter. In a pair-label L we require the composition $L^{\mathsf{red}} \circ L^{\mathsf{par}}$ to be defined, and indeed it represents the previous single label. Thus each transition will have an underlying *pair* of IPOs, not just a single IPO.

For a pair-label L, in expressions we shall usually write L to mean $L^{\mathsf{red}} \circ L^{\mathsf{par}}$. Also, for any equivalence $\equiv$, we shall take $L \equiv M$ to mean $L^{\mathsf{red}} \equiv M^{\mathsf{red}}$ and $L^{\mathsf{par}} \equiv M^{\mathsf{par}}$. None of the results in Part I and II appears to depend on the use of pair-labels; we adopt them mainly in anticipation of future work that may depend on them.[5] For example it provides a starting point for relating our present work more closely to that of Sewell [37], in which labels themselves are parametric.

With these two refinements, we now define transition systems. We allow for a broad class of transitions, within which we distinguish those based upon IPOs.

$$\begin{tikzcd}
\cdot \arrow[r, "L^{\mathsf{par}}"] & \cdot \arrow[r, "L^{\mathsf{red}}"] & \cdot \\
\cdot \arrow[u, "a"] \arrow[r, "d"'] & \cdot \arrow[u] \arrow[r, "R"'] & \cdot \arrow[u, "D"']
\end{tikzcd}$$

Definition 5.1 (transition) A *transition* consists of a quadruple written $a \xrightarrow{L}\!\triangleright_{\lambda} a'$, where a and a' are ground, L is a pair-label and there exist a reaction rule (R, R', trans), an active context D and a parameter d such that the above diagram commutes, and

$$\begin{array}{rcll} \lambda & = & \mathsf{width}(D)(m) & \text{where } m = \mathsf{width}(\mathsf{cod}(R)) \\ a' & = & D' \circ R' \circ d' & \text{where } D' \bumpeq D \text{ and } d' \bumpeq \mathsf{trans}(d)\ . \end{array}$$

We say that the reaction rule and the diagram *underlie* the transition. A transition is *minimal* if the underlying diagram is an IPO-pair. ■

Definition 5.2 (transition system) Given a WRS $\acute{\mathbf{A}}$, a *(labelled) transition system* $\mathcal{L}$ for $\acute{\mathbf{A}}$ is a pair $(\mathsf{Agi}, \mathsf{Trans})$, where

- Agi is a set of interfaces called the *agent interfaces*; for $I \in \mathsf{Agi}$, the members of $\mathsf{Gr}(I)$ are called *agents* of $\mathcal{L}$.
- Trans is a set of transitions whose sources and targets are agents of $\mathcal{L}$.

The *full* (resp. *standard*) transition system for a WRS consists of all interfaces, together with all (resp. all minimal) transitions. When the WRS concerned is understood we shall denote these two transition systems respectively by FT and ST.

Let $\equiv$ be a static congruence (Definition 3.5) in a WRS equipped with $\mathcal{L}$. Suppose that, for every transition $a \xrightarrow{L}\!\triangleright_{\lambda} a'$ in $\mathcal{L}$, if $a \equiv b$ and $L \equiv M$ —where M is another label of $\mathcal{L}$ with $M \circ b$ defined— then there exist an agent b' and a transition $b \xrightarrow{M}\!\triangleright_{\lambda} b'$ in $\mathcal{L}$ such that $a' \equiv b'$. Then $\equiv$ and $\mathcal{L}$ are said to *respect* one another.

We abbreviate '(labelled) transition system' to TS. A TS $\mathcal{M}$ is a *sub-TS* of $\mathcal{L}$, written $\mathcal{M} \prec \mathcal{L}$, if its interfaces and transitions are included among those of $\mathcal{L}$. ■

[5] Indeed, examples exist where the bisimilarity for single labels is strictly coarser than for pair-labels.

From now on we shall use 'label' to mean a pair-label. Note that 'respect' is mutual between an equivalence and a TS, so that '$\mathcal{L}$ respects $\equiv$' means the same as '$\equiv$ respects $\mathcal{L}$'; we shall use them interchangeably.

Note that our definition of transition presupposes a set of reaction rules, i.e. an *unlabelled* transition relation. Sometimes —for example in CCS— labelled transition systems have been the primary means of providing process dynamics, and unlabelled transitions corresponded to transitions with a 'null' label (τ in CCS). But in this work we are committed to taking reaction rules as primary, because they can be described generally without any presupposition about the interaction discipline of each calculus.

Returning briefly to Example 6 we now see that the location component in transitions allows us to distinguish between the two agents a and b. In fact their only transitions take the respective forms $a \overset{L}{\longrightarrow\!\!\triangleright}_{\{0\}}$ and $b \overset{L}{\longrightarrow\!\!\triangleright}_{\{1\}}$, where $L = (\mathsf{id}, \mathsf{id})$.

We shall meet various different equivalences when we deal with bigraphical WRSs. For support equivalence ($\stackrel{\frown}{=}$) we now deduce from Propositions 3.10(5) and 4.4 that:

Proposition 5.3 (support translation of transitions) *In any wide reactive system* $\acute{\mathbf{A}}$*, the full and standard transition systems respect support equivalence.*

The standard transition system ST is only useful when RPOs exist. Later we shall show how to derive from it a TS for the quotient WRS, which may not possess RPOs. But however transitions are defined or derived, we may define behavioural equivalences and preorders in the usual way. Here we shall limit attention to strong bisimilarity. (Throughout this paper we shall omit 'strong' since we do not define or use weak bisimilarity.)

Definition 5.4 (wide bisimilarity) Let $\acute{\mathbf{A}}$ be a wide reactive system equipped with a TS $\mathcal{L}$. A *simulation (on $\mathcal{L}$)* is a binary relation $\mathcal{S}$ between agents with equal interface such that if $a\mathcal{S}b$ and $a \overset{L}{\longrightarrow\!\!\triangleright}_{\lambda} a'$ in $\mathcal{L}$, then whenever $L \circ b$ is defined there exists b' such that $b \overset{L}{\longrightarrow\!\!\triangleright}_{\lambda} b'$ in $\mathcal{L}$ and $a'\mathcal{S}b'$. A *bisimulation* is a symmetric simulation. Then *bisimilarity (on $\mathcal{L}$)*, denoted by $\sim_{\mathcal{L}}$, is the largest bisimulation (on $\mathcal{L}$). ■

We shall often omit 'on $\mathcal{L}$', and write $\sim$ for $\sim_{\mathcal{L}}$, when $\mathcal{L}$ is understood from the context. This will usually be when $\mathcal{L}$ is ST.

Note the slight departure from the usual definition of bisimulation of Park [31]; here we must require $L \circ b$ to be defined. This is merely a technical detail, provided that the TS respects support translation; for then, whenever $L \circ a$ is defined there will always exist $L' \stackrel{\frown}{=} L$ for which *both* $L' \circ a$ and $L' \circ b$ are defined. Moreover if the WRS is based on a category —in particular if it is a support quotient— then the side-condition holds automatically; in this case the definition of bisimilarity reduces to the standard.

We may now prove our main congruence theorem for WRSs. Its main point is that ST ensures bisimulation congruence. The reader can deduce the (perhaps more obvious!) result that FT ensures the same; simply replace the word 'IPO' by'commuting square' in the proof.

Theorem 5.5 (congruence of wide bisimilarity) *In a wide reactive system with RPOs, equipped with the standard transition system, wide bisimilarity of agents is a congruence; that is, if $a_0 \sim a_1$ then $C \circ a_0 \sim C \circ a_1$.*

(a)

$$\begin{tikzcd} {} \arrow[r, "M^{\mathsf{par}}"] & {} \arrow[r, "M^{\mathsf{red}}"] & {} \\ {} \arrow[u, "C\circ a_0"'] \arrow[r, "d_0"] & {} \arrow[u] \arrow[r, "R_0"] & J_0 \arrow[u, "E_0"] \end{tikzcd}$$

(b)

$$\begin{tikzcd} {} \arrow[r, "M^{\mathsf{par}}"] & {} \arrow[r, "M^{\mathsf{red}}"] & {} \\ {} \arrow[u, "C"'] \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[u] \arrow[r, "L^{\mathsf{red}}"] & {} \arrow[u, "E"] \\ {} \arrow[u, "a_0"'] \arrow[r, "d_0"] & {} \arrow[u] \arrow[r, "R_0"] & J_0 \arrow[u, "D_0"] \arrow[uu, bend right, "E_0"'] \end{tikzcd}$$

(c)

$$\begin{tikzcd} {} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\ {} \arrow[u, "a_1"'] \arrow[r, "d_1"] & {} \arrow[u] \arrow[r, "R_1"] & J_1 \arrow[u, "D_1"] \end{tikzcd}$$

(d)

$$\begin{tikzcd} {} \arrow[r, "M^{\mathsf{par}}"] & {} \arrow[r, "M^{\mathsf{red}}"] & {} \\ {} \arrow[u, "C"'] \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[u] \arrow[r, "L^{\mathsf{red}}"] & {} \arrow[u, "E"] \\ {} \arrow[u, "a_1"'] \arrow[r, "d_1"] & {} \arrow[u] \arrow[r, "R_1"] & J_1 \arrow[u, "D_1"] \arrow[uu, bend right, "E_1"'] \end{tikzcd}$$

Proof The proof is along the lines of Theorem 3.9 in Leifer [22]. We establish the bisimulation

$$\mathcal{S} \overset{\text{def}}{=} \{(C \circ a_0, C \circ a_1) \mid a_0 \sim a_1,\ C \text{ any context}\}\ .$$

Suppose that $a_0 \sim a_1$, and that $C \circ a_0 \xrightarrow{M}\!\!\triangleright_\mu b'_0$, for some label M such that $M \circ C \circ a_1$ is defined. It is enough to find b'_1 such that $C \circ a_1 \xrightarrow{M}\!\!\triangleright_\mu b'_1$ and $(b'_0, b'_1) \in \mathcal{S}$.

There exist a reaction rule $(R_0, R'_0, \mathsf{trans}_0)$ with outer face J_0, an active context E_0 and a parameter d_0 such that diagram (a) is an IPO-pair; moreover if $\mathsf{width}(J_0) = m_0$ then $\mathsf{width}(E_0)(m_0) = \mu$ and $b'_0 = E_0 \circ R'_0 \circ \mathsf{trans}_0(d_0)$. Then because consistent pairs have RPOs, there exists an RPO for (a_0, d_0) relative to the given bound, and using Proposition 3.10(5) we can complete diagram (b) so that every square is an IPO.

D_0 is active at m_0 by Definition 4.3, so the lower squares represent a transition $a_0 \xrightarrow{L}\!\!\triangleright_\lambda a'_0$, where $\lambda = \mathsf{width}(D_0)(m_0)$ and $a'_0 = D_0 \circ R'_0 \circ \mathsf{trans}_0(d_0)$. Also E is active at λ. Since $a_0 \sim a_1$ there is a transition $a_1 \xrightarrow{L}\!\!\triangleright_\lambda a'_1$ with $a'_0 \sim a'_1$. (Note that $L \circ a_1$ is defined, since $M \circ C \circ a_1$ is defined and $M \circ C = E \circ L$.) So there exist a reaction rule $(R_1, R'_1, \mathsf{trans}_1)$ with outer face J_1, an active context D_1 and a parameter d_1 such that diagram (c) is an IPO pair; moreover $\mathsf{width}(D_1)(m_0) = \lambda$ and $a'_1 = D_1 \circ R'_1 \circ \mathsf{trans}_1(d_1)$.

Now replace the lower two squares of (b) by diagram (c), obtaining diagram (d) in which, by Proposition 3.10(5), the large square is an IPO. Moreover $E_1 \overset{\text{def}}{=} E \circ D_1$ is active, since E is active at λ. Hence $C \circ a_1 \xrightarrow{M}\!\!\triangleright_\mu b'_1$ where $b'_1 \overset{\text{def}}{=} E_1 \circ R'_1 \circ \mathsf{trans}_1(d_1)$. Finally $(b'_0, b'_1) \in \mathcal{S}$ as required, because $b'_0 = E \circ a'_0$ and $b'_1 = E \circ a'_1$ with $a'_0 \sim a'_1$. ■

We shall henceforth often omit the adjective ‘wide’ when discussing bisimilarity. We should remark that we are taking (strong) bisimilarity as a representative of many preorders and equivalences; Leifer [22] has proved congruence theorems for several others, and we expect that those results can be transferred to the present setting.

Now, if a WRS is equipped with a TS we can define transitions for various quotient WRSs as follows:

Definition 5.6 (transitions for quotient WRSs) Let $\acute{\mathbf{A}}$ be a WRS equipped with a TS $\mathcal{L}$, and let $\mathcal{F}\colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ be a WRS functor. We say that $\mathcal{F}$ *respects* $\mathcal{L}$ if the static congruence it induces on $\acute{\mathbf{A}}$ respects $\mathcal{L}$. The TS $\mathcal{F}(\mathcal{L})$ *induced* by $\mathcal{F}$ on $\acute{\mathbf{B}}$ has the agent interface $\mathcal{F}(I)$ whenever I is an agent interface of $\mathcal{L}$, and whenever $\mathcal{L}$ has a transition $a \xrightarrow{L}{\triangleright}_{\lambda} a'$ then $\mathcal{F}(\mathcal{L})$ has the transition

$$\mathcal{F}(a) \xrightarrow{\mathcal{F}(L)}{\triangleright}_{\lambda} \mathcal{F}(a') \ . \qquad \blacksquare$$

This definition always makes sense, but it will not always make bisimilarity a congruence in $\acute{\mathbf{B}}$, even if it is so in $\acute{\mathbf{A}}$. However, the next theorem tells us when this will be ensured. Recall that a *full* functor is surjective for each homset.

Theorem 5.7 (transitions induced by functors) *Let $\acute{\mathbf{A}}$ be equipped with a TS $\mathcal{L}$. Let $\mathcal{F}\colon \acute{\mathbf{A}} \to \acute{\mathbf{B}}$ be a full WRS functor that is the identity on objects and respects $\mathcal{L}$. Then the following hold for $\mathcal{F}(\mathcal{L})$:*

(1) $a \sim b$ in $\acute{\mathbf{A}}$ iff $\mathcal{F}(a) \sim \mathcal{F}(b)$ in $\acute{\mathbf{B}}$.

(2) If bisimilarity is a congruence in $\acute{\mathbf{A}}$ then it is a congruence in $\acute{\mathbf{B}}$.

(3) Both (1) and (2) hold when $\mathcal{F} = [\cdot]\colon \acute{\mathbf{A}} \to \mathbf{A}$, the support quotient functor.

Proof (1) ($\Rightarrow$) We establish in $\acute{\mathbf{B}}$ the bisimulation

$$\mathcal{R} = \{(\mathcal{F}(a), \mathcal{F}(b)) \mid a \sim b\} \ .$$

Let $a \sim b$ in $\acute{\mathbf{A}}$, and let $p = \mathcal{F}(a)$, $q = \mathcal{F}(b)$ and $p \xrightarrow{M}{\triangleright}_{\lambda} p'$ in $\acute{\mathbf{B}}$. Then by definition of the induced TS we can find L and a' such that $M = \mathcal{F}(L)$ and $p' = \mathcal{F}(a')$, and $a \xrightarrow{L}{\triangleright}_{\lambda} a'$ in $\acute{\mathbf{A}}$ with $L \circ b$ defined. So for some b' we have $b \xrightarrow{L}{\triangleright}_{\lambda} b'$ with $a' \sim b'$. It follows that $q \xrightarrow{M}{\triangleright}_{\lambda} q'$ in $\acute{\mathbf{B}}$, where $q' = \mathcal{F}(b')$ and $(p', q') \in \mathcal{R}$, so we are done.

(1) ($\Leftarrow$) We establish in $\acute{\mathbf{A}}$ the bisimulation

$$\mathcal{S} = \{(a, b) \mid \mathcal{F}(a) \sim \mathcal{F}(b)\} \ .$$

Let $\mathcal{F}(a) \sim \mathcal{F}(b)$ in $\acute{\mathbf{B}}$, and let $p = \mathcal{F}(a)$, $q = \mathcal{F}(b)$ where $a \xrightarrow{L}{\triangleright}_{\lambda} a'$ in $\acute{\mathbf{A}}$ with $L \circ b$ defined. Then $p \xrightarrow{M}{\triangleright}_{\lambda} p'$ in $\acute{\mathbf{B}}$, where $M = \mathcal{F}(L)$ and $p' = \mathcal{F}(a')$. So for some q' we have $q \xrightarrow{M}{\triangleright}_{\lambda} q'$ with $p' \sim q'$. This transition must arise from a transition $b_1 \xrightarrow{L_1}{\triangleright}_{\lambda} b_1'$ in $\acute{\mathbf{A}}$, where $q = \mathcal{F}(b_1)$, $M = \mathcal{F}(L_1)$ and $q' = \mathcal{F}(b_1')$. But then $b_1 \equiv b$ and $L_1 \equiv L$, where $\equiv$ is the equivalence induced by $\mathcal{F}$; we also have $L \circ b$ defined, and $\mathcal{L}$ respects $\equiv$, so we can find b' for which $b \xrightarrow{L}{\triangleright}_{\lambda} b'$ and $b_1' \equiv b'$. But also $(a', b') \in \mathcal{S}$ so we are done.

(2) Assume that bisimilarity in $\acute{\mathbf{A}}$ is a congruence. In $\acute{\mathbf{B}}$, let $p \sim q$ and let G be a context with $G \circ p$ and $G \circ q$ defined. Then there exist a, b, C in $\acute{\mathbf{A}}$ with $p = \mathcal{F}(a)$, $q = \mathcal{F}(b)$ and $G = \mathcal{F}(C)$, and with $C \circ a$ and $C \circ b$ defined. From (1)($\Leftarrow$) we have $a \sim b$, hence by assumption $C \circ a \sim C \circ b$. Applying the functor $\mathcal{F}$ we have from (1)($\Rightarrow$) that $G \circ p \sim G \circ q$ in $\acute{\mathbf{B}}$, as required.

(3) The result follows immediately from Proposition 5.3. $\blacksquare$

In a later section we shall set up bigraphical reactive systems as WRSs. Then, using the theorems we have just proved —or close analogues of them— we shall derive TS and deduce behavioural congruences for them.

We now turn to a question that arises strongly in applications. Our standard TS, containing only the minimal transitions, is of course much smaller than the full TS. But it turns out that in particular cases we can reduce the standard TS still further, without affecting bisimilarity. We introduce here the basic concepts to make this idea precise, since they do not depend at all on the notion of bigraph.

Definition 5.8 (relative bisimulation, adequacy) Assume given a TS $\mathcal{L}$, with a sub-TS $\mathcal{M}$. A *relative bisimulation for $\mathcal{M}$ (on $\mathcal{L}$)* is a symmetric relation $\mathcal{S}$ such that whenever $a\mathcal{S}b$, then for every transition $a \xrightarrow{L}_{\lambda} a'$ in $\mathcal{M}$, with $L \circ b$ defined, there exists b' such that $b \xrightarrow{L}_{\lambda} b'$ in $\mathcal{L}$ and $a'\mathcal{S}b'$. Define *relative bisimilarity for $\mathcal{M}$ (on $\mathcal{L}$)*, denoted by $\sim^{\mathcal{M}}_{\mathcal{L}}$, to be the largest relative bisimulation for $\mathcal{M}$ (on $\mathcal{L}$).

We call $\mathcal{M}$ *adequate (for $\mathcal{L}$)* if $\sim^{\mathcal{M}}_{\mathcal{L}}$ coincides with $\sim_{\mathcal{L}}$ on the agents of $\mathcal{M}$; if $\mathcal{M}$ has interfaces $\mathcal{I}$, we write this as $\sim^{\mathcal{M}}_{\mathcal{L}} = \sim_{\mathcal{L}} \upharpoonright \mathcal{I}$. ■

When $\mathcal{L}$ is understood we may omit 'on $\mathcal{L}$'; equally we may write $\sim^{\mathcal{M}}$ for $\sim^{\mathcal{M}}_{\mathcal{L}}$. Note that, for $a \sim^{\mathcal{M}}_{\mathcal{L}} b$, we require b only to match the transitions of a that lie in $\mathcal{M}$, and b's matching transition need not lie in $\mathcal{M}$. This means that relative bisimilarity is in general not transitive, so it is not in itself a behavioural equivalence.

The value of relative bisimilarity lies in the case when $\mathcal{M}$ is adequate for $\mathcal{L}$, for then the proof technique of relative bisimulation can relieve us of the task of checking a large class of transitions. Indeed it may be the case that fewer labels are employed in $\mathcal{M}$-transitions than in $\mathcal{L}$-transitions; then we only have to consider transitions involving this smaller set of labels.

Even at this abstract level of WRSs, we can draw attention to possibilities for a transition system $\mathcal{M}$ adequate for $\mathcal{L}$, in particular when $\mathcal{L}$ is ST. A simple example depends on the fact that ST is *closed under isomorphism*, i.e. if $a \xrightarrow{L}_{\lambda} a'$ is a transition of ST then so is $\iota a \xrightarrow{\kappa L \iota^{-1}}_{\lambda} \kappa a'$ for any isos ι and κ. (We are omitting '$\circ$' when composing with an iso.) Then when checking for bisimilarity with a given a, intuitively it should suffice to consider not *every* transition of a, but only one in every iso class. This holds more generally:

Proposition 5.9 (representative transitions) *Let $\mathcal{L}$ be a transition system closed under isomorphism, and let $\mathcal{M} \prec \mathcal{L}$ be a sub-TS. Suppose that, for every transition $a \xrightarrow{L}_{\lambda} a'$ in $\mathcal{L}$, there is a transition $a \xrightarrow{\kappa L}_{\lambda} \kappa a'$ in $\mathcal{M}$ for some iso κ. Then $\mathcal{M}$ is adequate for $\mathcal{L}$.*

Proof We show that $\mathcal{R} = \{(\iota a, \iota b) \mid a \sim^{\mathcal{M}}_{\mathcal{L}} b\}$ is an $\mathcal{L}$-bisimulation. Let $a \sim^{\mathcal{M}}_{\mathcal{L}} b$, and let $\iota a \xrightarrow{L}_{\lambda} a'$ in $\mathcal{L}$. We must find a matching $\mathcal{L}$-transition for ιb.

Since isomorphism preserves transitions in $\mathcal{L}$, there is an $\mathcal{L}$-transition $a \xrightarrow{L\iota}_{\lambda} a'$. So by assumption there is an $\mathcal{M}$-transition $a \xrightarrow{\kappa L \iota}_{\lambda} a'' \stackrel{\text{def}}{=} \kappa a'$.

Since $a \sim^{\mathcal{M}}_{\mathcal{L}} b$ there is an $\mathcal{L}$-transition $b \xrightarrow{\kappa L \iota}_{\lambda} b''$ with $a'' \sim^{\mathcal{M}}_{\mathcal{L}} b''$. Applying two isos, there is an $\mathcal{L}$-transition $\iota b \xrightarrow{L}_{\lambda} b' \stackrel{\text{def}}{=} \kappa^{-1} b''$. But $a' = \kappa^{-1} a''$, so $(a', b') \in \mathcal{R}$ and we are done. ■

We shall exploit this result in Section 15.

A deeper example of adequacy arises from the intuition that the transitions that really matter are those where the agent 'contributes' to the underlying reaction, i.e. a supplies a 'part' of the redex R, leaving the label L to supply the rest. We can make this precise in terms of support: we are interested in transitions a whose underlying redex R is such that $|a| \cap |R| \neq \emptyset$. We call such transitions *engaged*.

Intuitively, we may conjecture that the engaged transitions are adequate, for the standard TS. We shall later prove this for a particular class of bigraphical reactive systems, broad enough to include the π-calculus and the ambient calculus. It is very nice when the conjecture holds, for it means that the only significant labels L are those whose leading part L^{red} is strictly contained in some redex R.

We now look at a rather well-behaved kind of sub-TS. For arbitrary $\mathcal{M} \prec \mathcal{L}$ and any given pair (L, λ), it is possible that $\mathcal{M}$ contains some but not all of the (L, λ)-transitions in $\mathcal{L}$. But if this is not the case —i.e. if such pairs determine which transitions are in $\mathcal{M}$— then the situation is somewhat simpler:

Definition 5.10 (definite sub-TS) Let $\mathcal{M} \prec \mathcal{L}$. Call $\mathcal{M}$ *definite for* $\mathcal{L}$ if, for all pairs (L, λ) and all transitions of $\mathcal{L}$

$$a \xrightarrow{L}_{\lambda} a' \in \mathcal{M} \text{ iff } b \xrightarrow{L}_{\lambda} b' \in \mathcal{M} \; . \qquad \blacksquare$$

Then immediately we deduce that a relative bisimilarity is an absolute one:

Proposition 5.11 (definite sub-TS) *If $\mathcal{M}$ is definite for $\mathcal{L}$ then $\sim_{\mathcal{M}} = \sim_{\mathcal{L}}^{\mathcal{M}}$.*

If $\mathcal{M}$ is definite and adequate for $\mathcal{L}$, we can deduce an important corollary for later use. To illustrate it, suppose that $\mathcal{L}$ is the standard TS. If we are interested only in agents at $\mathcal{I}$, and are able to establish that $\mathcal{M}$ with interfaces $\mathcal{I}$ is definite and adequate for $\mathcal{L}$, then we can deduce congruence for bisimilarity on $\mathcal{M}$. More generally:

Corollary 5.12 (adequate congruence) *Let $\mathcal{M}$, with interfaces $\mathcal{I}$, be definite and adequate for $\mathcal{L}$. Then*

(1) The bisimilarities on $\mathcal{M}$ and $\mathcal{L}$ coincide at $\mathcal{I}$, i.e. $\sim_{\mathcal{M}} = \sim_{\mathcal{L}} \restriction \mathcal{I}$.

(2) If $\sim_{\mathcal{L}}$ is a congruence, then $\sim_{\mathcal{M}}$ is a congruence; that is, for any $C \colon I \to J$ where $I, J \in \mathcal{I}$, if $a \sim_{\mathcal{M}} b$ then $C \circ a \sim_{\mathcal{M}} C \circ b$.

Part II
Bigraphical Reactive Systems

In Part II we begin by defining the notion of a *pure bigraph* formally, in terms of its two constituents: a *place graph* representing locality and a *link graph* representing connectivity. We continue by defining these two notions in turn, ensuring that they enjoy the categorical properties that we shall need. We then combine them, yielding a theory of pure bigraphs where locality and connectivity are totally independent. A short section is devoted to the algebraic theory of pure bigraphs, showing that they possess a simple complete axiomatisation.

We proceed to relax the independence of locality and connectivity, in a controlled manner, in defining *binding bigraphs*; these allow certain *local* names to have a scope consisting of a particular bigraphical region. Properties of binding bigraphs are derived from those of the underlying pure ones.

Finally we introduce dynamics, in the form of reaction rules, yielding bigraphical reactive systems (BRSs). These are shown to be a special case of WRSs. We therefore apply Part I to yield labelled transition systems and behavioural congruences, for both pure and binding BRSs.

6 Pure bigraphs: definition

In this section we define the notion of *pure bigraph* formally, in terms of the constituent notions of *place graph* and *link graph*, which are dealt with in the following two sections. Then in Section 9 we resume the study of pure bigraphs, combining the properties of its constituents. In Section 10 we develop their algebraic theory. In Section 11 we define a *binding bigraphs* as an enrichment of the pure ones; we ensure that they enjoy the properties that allow us to apply the theory developed in Part I. Finally, in Section 12 we give the central definition of a *bigraphical reactive system* (BRS) and study its dynamic behaviour; then we apply the results of Part I to derive labelled transitions and congruences for both pure and binding BRSs.

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*. ■

Note that a signature need not be finite, or even denumerable. Thus a bigraph, though itself finite, may denote an element of a continuous state space. We shall not here exploit this possibility, but we comment further on it in Section 16.

As we saw in Section 1 of Part I, a non-atomic node —one with a non-atomic control— may contain other nodes. A node's control determines its ports, and if the control is active then reactions are permitted inside the node. A passive node —such as a get-node in the π-calculus— can be thought of as a script, or program, awaiting activation; this must take the form of a reaction that destroys the node boundary.

In refinements of the theory a signature may carry further information, such as a *sign* and/or a *type* for each port. The sign may be used, for example, to enforce the restriction that each negative port is connected to exactly one positive port, as in action calculi [9, 25]. Another possible refinement is a *kind* assigned to each node, determining the controls of the nodes it may contain. (Our atomic nodes already represent the most restrictive kind.) In Part III we shall define an important refinement that allows names to have *scope*, and controls to *bind* names. The theory of *pure* bigraphs, where names have no scope, is prerequisite to understanding all these refinements.

We begin by defining *concrete* bigraphs. The definition is 'top-down'; here we define a bigraph as the combination of two parts, and in the following sections we define those parts themselves.

Definition 6.2 (concrete pure bigraph) A *(concrete) pure bigraph* over the signature $\mathcal{K}$ takes the form $G = (V, E, ctrl, G^{\mathsf{P}}, G^{\mathsf{L}}) : I \to J$ 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 *name set*. Its first two components V and E are finite sets of *nodes* and *edges* respectively. The third component $ctrl \colon V \to \mathcal{K}$, a *control map*, assigns a control to each node. The remaining two are:

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

Place graphs and link graphs are defined in Definitions 7.1 and 8.1 respectively. ■

We refer to these as concrete bigraphs because their nodes and edges have identity. Thus we shall work with a supported precategory of bigraphs, because there we shall be able to find RPOs. The support of a concrete bigraph consists of its nodes and edges; in terms of the definition, $|G| = V + E$. In Section 9 we shall take the quotient by support equivalence to obtain *abstract* bigraphs. As is usual in graph theory, we shall omit the adjectives 'concrete' and 'abstract' when they are unimportant or implied by the context.

We shall normally work with a fixed but unspecified signature. We refer to G as the *combination* of its constituents G^{P} and G^{L}; we write it as $G = \langle G^{\mathsf{P}}, G^{\mathsf{L}} \rangle$. A place graph can be combined with a link graph iff they have the same node set and control map. In Section 9 we revisit bigraphs, developing their structure by combining attributes from their constituent place graphs and link graphs.

7 Place graphs

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

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*. ■

The acyclicity condition makes the parent map prnt represent a forest of n unordered trees. The sites and roots provide the means of composing the forests of two place graphs; each root of the first is planted in a distinct site of the second. Figure 9 shows two simple examples of composition, $B_0 \circ A_0$ and $B_1 \circ A_1$. Formally:

Definition 7.2 (precategory of place graphs) The precategory ´PLG has finite ordinals as objects and place graphs as arrows. The composition $A_1 \circ A_0 : m_0 \to m_2$ of two place graphs $A_i = (V_i, \mathit{ctrl}_i, \mathit{prnt}_i) : m_i \to m_{i+1}$ $(i = 0, 1)$ is defined when the two node sets are disjoint; then $A_1 \circ A_0 \overset{\text{def}}{=} (V, \mathit{ctrl}, \mathit{prnt})$ where $V = V_0 \uplus V_1$, $\mathit{ctrl} = \mathit{ctrl}_0 \uplus \mathit{ctrl}_1$, and $\mathit{prnt} = (\mathsf{Id}_{V_0} \uplus \mathit{prnt}_1) \circ (\mathit{prnt}_0 \uplus \mathsf{Id}_{V_1})$. The identity place graph at m is $\mathsf{id}_m \overset{\text{def}}{=} (\emptyset, \emptyset_{\mathcal{K}}, \mathsf{Id}_m) : m \to m$. ■

It is easy to check that $A \circ \mathsf{id} = A = \mathsf{id} \circ A$, and that composition is associative. Note that ´PLG is supported, with node sets V as support.

Here are some basic properties:

Definition 7.3 (barren, sibling, active, passive) A node or root is *barren* if it has no children. Two places are *siblings* if they have the same parent. A site s of A is *active* if $\mathit{ctrl}(v)$ is active whenever $v > s$; otherwise s is *passive*. If s is active (resp. passive) in A, we also say that A is *active* (resp. *passive*) *at* s. ■

When dealing with many place graphs A, B, $\ldots$, instead of indexing their parent maps as prnt_A, prnt_B etc. we shall find it more convenient to abuse notation and denote the parent map of a place graph A again by A. The context will prevent ambiguity; for example in $B \circ A$ we are talking of place graphs, while in $B(A(v))$ we are talking of their parent maps. Thus $(B \circ A)(v)$ means the parent map of the composite place graph $B \circ A$ applied to the node v. Note especially that $(B \circ A)(v)$ differs from $B(A(v))$; in fact if $v \in V_A$ then $(B \circ A)(v)$ is equal to $A(v)$ if this is a node, otherwise equal to $B(A(v))$.

Proposition 7.4 (isomorphisms 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.*

What is a suitable tensor product for ´PLG? We do not want $A \otimes B$ to have the effect of merging nodes from A and B. So we adopt a partial tensor product, with $A \otimes B$ defined exactly when the node sets are disjoint, in which case its node set is $V_A \uplus V_B$. Intuitively, the tensor product of two place graphs consists in placing them side-by-side.

Definition 7.5 (tensor product) The *tensor product* $\otimes$ in ´PLG is defined as follows: On objects, we take $m \otimes n \overset{\text{def}}{=} m + n$. For two place graphs $A_i : m_i \to n_i$ $(i = 0, 1)$ we take $A_0 \otimes A_1 : m_0 + m_1 \to n_0 + n_1$ to be defined when A_0 and A_1 have disjoint node sets; for the parent map, we first adjust the sites and roots of A_1 by adding them to m_0 and n_0 respectively, then take the union of the two parent maps. ■

Epimorphisms (epis) will play a central role, both for place graphs and for link graphs. Monomorphisms (monos) will also be used. Recall that in the category of sets with functions the epis and monos are the surjective and injective functions respectively. Here we find something analogous:

Proposition 7.6 (epis and monos in place graphs) *In ´PLG, a place graph is an epi iff no root is barren; it is mono iff no two sites are siblings.*

We shall now prove that RPOs always exist in place graphs, and we show how to construct them. We first give a simple intuition. Let $\vec{D}$ be a bound for $\vec{A}$; we wish to build an RPO $(\vec{B}, B)$ as shown in the diagram below. To form $\vec{B}$, we first truncate $\vec{D}$ by removing the roots, and all nodes not present in $\vec{A}$. Then for the upper interface of $\vec{B}$, we create a new parent (a root) for each place orphaned by the truncation, equating these new roots only when required so that $B_0 \circ A_0 = B_1 \circ A_1$.

Notation When considering a pair $\vec{A} : h \to \vec{m}$ of place graphs with common sites h, we shall adopt a convention for naming their nodes. We denote the node set of A_i $(i = 0, 1)$ by V_i, and denote $V_0 \cap V_1$ by V_2. Recall that $\bar{\imath}$ means $1 - i$ for $i \in 2$. We shall use $v_i, v_i', \ldots$ to range over V_i $(i = 0, 1, 2)$, and r_i, r_i' to range over the roots m_i $(i = 0, 1)$. We shall also use $w_2, w_2', \ldots$ to range over $h \uplus V_2$; this is useful because shared sites behave just like shared nodes in our construction of pushouts. ■

We shall now give a construction for RPOs in ´PLG.

Construction 7.7 (RPOs in place graphs) An RPO $(\vec{B} \colon \vec{m} \to \hat{m}, B \colon \hat{m} \to p)$, for a pair $\vec{A} \colon h \to \vec{m}$ of place graphs relative to a bound $\vec{D} \colon \vec{m} \to p$, will be built in three stages. We use the notational conventions introduced above.

$$\begin{tikzcd}
 & & p & & \\
 & & \hat{m} \arrow[u, "B"'] & & \\
m_0 \arrow[uurr, "D_0"] \arrow[urr, "B_0"'] & & & & m_1 \arrow[uull, "D_1"'] \arrow[ull, "B_1"] \\
 & & {} \arrow[ull, "A_0"] \arrow[urr, "A_1"'] & &
\end{tikzcd}$$

nodes: If V_i are the nodes of A_i $(i = 0, 1)$ then the nodes of D_i are $V_{\bar{\imath}} \backslash V_2 \uplus V_3$ for some V_3. Define the nodes of B_i and B to be $V_{\bar{\imath}} \backslash V_2$ $(i = 0, 1)$ and V_3 respectively.

interface: Construct the shared codomain $\hat{m}$ of $\vec{B}$ as follows. First, define the roots in each m_i that must be mapped into $\hat{m}$:

$$m'_i \overset{\text{def}}{=} \{r \in m_i \mid D_i(r) \in V_3 \uplus p\} \ .$$

Next, on the disjoint sum $m'_0 + m'_1$, define $\cong$ to be the smallest equivalence for which $(0, r_0) \cong (1, r_1)$ whenever $A_0(w) = r_0$ and $A_1(w) = r_1$ for some $w \in h \uplus V_2$. Then define the codomain up to isomorphism:

$$\hat{m} \overset{\text{def}}{=} (m'_0 + m'_1)/\cong \ .$$

For each $r \in m'_i$ we denote the $\cong$-equivalence class of (i, r) by $\widehat{i, r}$.

parents: Define B_0 to simulate D_0 as far as possible (B_1 is similar):

$$\begin{array}{lll}
\text{For } r \in m_0 : & B_0(r) \overset{\text{def}}{=} & \begin{cases} \widehat{0, r} & \text{if } r \in m'_0 \\ D_0(r) & \text{if } r \notin m'_0 \end{cases} \\
\text{For } v \in V_1 \setminus V_2 : & B_0(v) \overset{\text{def}}{=} & \begin{cases} \widehat{1, r} & \text{if } A_1(v) = r \in m_1 \\ D_0(v) & \text{if } A_1(v) \notin m_1 \ . \end{cases}
\end{array}$$

Finally define B, to simulate both D_0 and D_1:

$$\begin{array}{ll}
\text{For } \hat{r} \in \hat{m} : & B(\hat{r}) \overset{\text{def}}{=} D_i(r) \ \text{ where } \widehat{i, r} = \hat{r} \\
\text{For } v \in V_3 : & B(v) \overset{\text{def}}{=} D_i(v) \ .
\end{array}$$

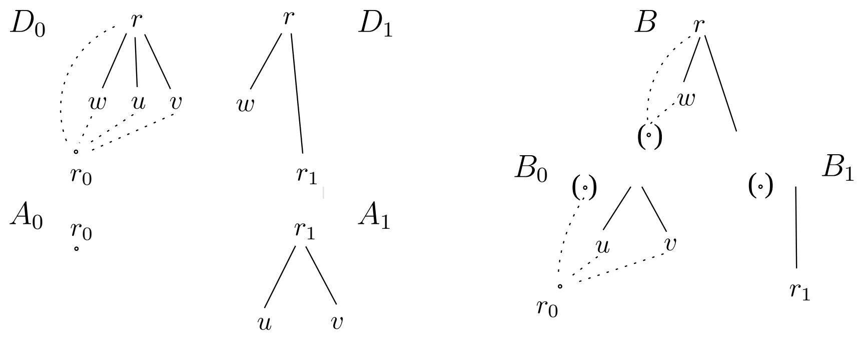
■

Several checks are necessary to ensure that this definition is sound; that is, that the right-hand sides in the clauses defining the parent maps B_0 and B are well-defined places in B_0 and B respectively. These points are checked in Appendix A.1, which gives the proof of the following:

Theorem 7.8 (RPOs in place graphs) *In ´PLG, whenever a pair $\vec{A}$ of place graphs has a bound $\vec{D}$, there exists an RPO $(\vec{B}, B)$ for $\vec{B}$ to $\vec{D}$, and Construction 7.7 yields such an RPO.*

For the behavioural theory of bigraphs we need to know not only how to construct each RPO (which we do for place graphs and link graphs separately), but also how to characterise the set of IPOs for a pair $\vec{A}$ with common domain. For then, when A_1 is a redex, we shall know all the labelled transitions of A_0. For place graphs, an immediate question is: how does our RPO $(\vec{B}, B)$ vary, when we keep $\vec{A}$ fixed but vary the given bound $\vec{D}$? One corollary of our next theorem will be that, if $\vec{A}$ are both epi, then $\vec{B}$ remains fixed, and only B varies; thus $\vec{A}$ is this case has a unique IPO — which is in fact a pushout. But in general $\vec{B}$ will vary, so there will be many IPOs.

This phenomenon will be important for our transition systems, and also occurs in link graphs, so it is worth seeing a simple example. The diagram shows a pair $\vec{A}$ in which A_0 consists only of a barren root r_0, while A_1 has two nodes u, v. There is a bound $\vec{D}$ with shared root r and an extra node w. Keeping D_1 fixed, we can vary D_0 by choosing $D_0(r_0)$ to be any of $\{w, u, v, r\}$ while keeping $D_0 \circ A_0$ fixed (since r_0 is barren in A_0). The diagram also indicates how the pair $\vec{B}$ of the RPO varies; for $D_0(r_0) \in \{u, v\}$ we take $B_0(r_0) = D_0(r_0)$, while for $D_0(r_0) \in \{w, r\}$ we take $B_0(r_0)$ to be an extra root (shown in parentheses), which also appears (barren) in B_1.



This example illustrates all the possible IPOs $\vec{B}$ for a given pair $\vec{A}$; each barren root r_i of A_i may be mapped in B_i *either* to a special root *or* to any node. In the former case the composite $B_i \circ A_i$ has a special root as a trace of r_i, but in the latter case it retains no such trace; so we shall call the latter case an *elision*.

Before constructing IPO families formally, we must answer the question: Under what conditions does a pair $\vec{A}$ have a bound at all? If a bound exists we call $\vec{A}$ *consistent*, and our next step is to define certain conditions on $\vec{A}$ that are necessary and sufficient for consistency. Roughly speaking, these conditions ensure that A_0 and A_1 treat their shared members (all the sites and some of the nodes) compatibly; then a bound $\vec{B}$ can exist, since B_0 can extend A_0 to include 'the part of A_1 not shared with A_0'. Such a bound will also be an IPO if, roughly, it adds no more than necessary for this.

Definition 7.9 (consistency conditions for place graphs) We define three *consistency* conditions on a pair $\vec{A} : h \to \vec{m}$ of place graphs. We let i range over $\{0, 1\}$; also recall that w_2, w_2' range over $h \uplus V_2$, the shared places.

CP0 $\quad ctrl_0(v_2) = ctrl_1(v_2)$
CP1 $\quad$ If $A_i(w) \in V_2$ then $w \in h \uplus V_2$ and $A_{\bar{\imath}}(w) = A_i(w)$
CP2 $\quad$ If $A_i(w_2) \in V_i \setminus V_2$ 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 $A_i(w) = A_i(w_2)$. ■

It may be helpful to express CP1 and CP2 in words; they are both to do with children of nodes in A_i. If $i = 0$, CP1 says that if the parent of any place w in A_0 is a node shared with A_1, then w is also shared and has the same parent in A_1. CP2 says, on the other hand, that if the parent of a shared place w_2 in A_0 is an *unshared* node, then its parent in A_1 must be a root, and further that any sibling of w_2 in A_1 must also be its sibling in A_0.

Necessity of these conditions is easy, and we omit the proof:

Proposition 7.10 (consistency in place graphs) *If the pair $\vec{A}$ has a bound, then the consistency conditions hold.*

Before going further, it may be helpful to see a simple example.

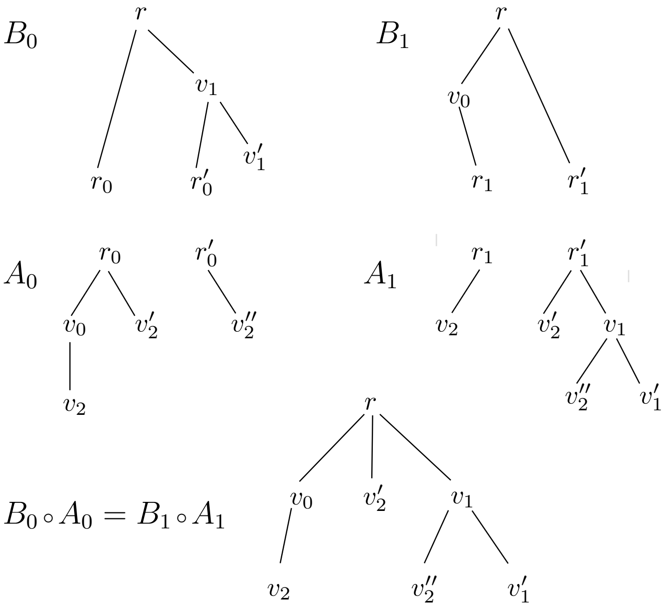


Figure 9: A consistent pair $\vec{A}$ of place graphs, with bound $\vec{B}$

Example 7 (consistent place graphs) Consider the pair $\vec{A}$ in Figure 9, each with two roots and no sites; nodes with subscript 2 are shared. (Controls are not shown). It is worth checking that the consistency conditions hold. What happens if an extra node u is added to A_1 as a sibling of v_2? If u is unshared then CP2 is violated, so no bound can exist. If u is shared, then to preserve the consistency conditions —in particular CP2— u must also become a sibling of v_2 in A_0; then $\vec{B}$ remains a bound. ■

Now, assuming the consistency conditions of Definition 7.9, we shall prove that there exist one or more IPOs for $\vec{A}$. (Thus, since any IPO is a bound, we shall also have shown that the consistency conditions are sufficient for a bound to exist.) The idea behind the following construction is that if $\vec{A}$ are both epis then there is a unique IPO; but every barren root r of $\vec{A}$ allows a variation, as we saw earlier.

Construction 7.11 (IPOs in place graphs) Assume the consistency conditions for the pair of place graphs $\vec{A} : h \to \vec{m}$. We define a family of IPOs $\vec{C} : \vec{m} \to n$ for $\vec{A}$ as follows.

nodes: Take the nodes of C_i to be $V_{\overline{i}} \backslash V_2$.

interface: For $i = 0, 1$ choose any subset ℓ_i of the barren roots in m_i. Set $k_i = m_i \backslash \ell_i$. Define $k_i' \subseteq k_i$, the roots to be mapped to the codomain n, by

$$k_i' \overset{\text{def}}{=} \{r \in k_i \mid \forall v \in V_2.\ A_i(v) = r \Rightarrow A_{\overline{i}}(v) \in m_{\overline{i}}\}\ .$$

Next, on the disjoint sum $k'_0 + k'_1$, define $\simeq$ to be the smallest equivalence such that $(0, r_0) \simeq (1, r_1)$ whenever $A_0(w) = r_0$ and $A_1(w) = r_1$ for some $w \in h \uplus V_2$. Then define the codomain up to isomorphism by

$$n \overset{\text{def}}{=} (k'_0 + k'_1)/\simeq \, .$$

For each $r \in k'_i$ we denote the $\simeq$-equivalence class of (i, r) by $\widehat{i, r}$.

parents: Choose two functions $\eta_i : \ell_i \rightarrow V_{\bar{\imath}} \backslash V_2$ $(i = 0, 1)$, arbitrary except that $\eta_i(r)$ is a non-atomic node for all $r \in \ell_i$. Then define the parent map $C_0 : m_0 \rightarrow n$ as follows (C_1 is similar):

For $r \in m_0$:

$$C_0(r) \overset{\text{def}}{=} \begin{cases} \widehat{0, r} & \text{if } r \in k'_0 \\ A_1(v) & \text{if } r \in k_0 \backslash k'_0, \text{ for } v \in h \uplus V_2 \text{ with } A_0(v) = r \\ \eta_0(r) & \text{if } r \in \ell_0 \end{cases}$$

For $v \in V_1 \backslash V_2$:

$$C_0(v) \overset{\text{def}}{=} \begin{cases} \widehat{1, r} & \text{if } A_1(v) = r \in m_1 \\ A_1(v) & \text{if } A_1(v) \notin m_1 \, . \end{cases}$$

■

The maps η_i are called *elisions*; this refers to the fact that the barren roots ℓ_i in A_i are not exported in the IPO interface n, but instead mapped into the body of C_i. There is a distinct IPO for each choice of ℓ_i and η_i. However the IPO will be unique if $\ell_i = \emptyset$ is forced $(i = 0, 1)$. This can happen for one of two reasons: either A_i has no barren roots; or $V_{\bar{\imath}} \backslash V_2$ is empty (i.e. all nodes of $A_{\bar{\imath}}$ are shared), so no elision can exist.

We have to show that the definition of C_0 is sound. Thus in the first clause for $C_0(r)$ we must ensure that $v \in V_2$ exists such that $A_0(v) = r$, and that each such v yields the same value $A_1(v)$ in $V_1 \backslash V_2$; in the second clause for $C_0(v)$ we must ensure that $r \in k'_1$. The consistency conditions do ensure this, and also that $C_0 \circ A_0 = C_1 \circ A_1$.

We can now validate Construction 7.11:

Theorem 7.12 (characterising IPOs for place graphs) *A pair $\vec{C} : \vec{m} \rightarrow n$ is an IPO for $\vec{A} : h \rightarrow \vec{m}$ iff it is generated (up to isomorphism) by Construction 7.11.*

Proof (outline) We work up to isomorphism.

($\Rightarrow$) Recall that a bound $\vec{B}$ for $\vec{A}$ is an IPO iff it is the legs of an RPO for $\vec{A}$ w.r.t. some bound $\vec{D}$. So assume such a $\vec{B} : \vec{m} \rightarrow m$ built by Construction 7.7, and recall the subsets $m'_i \subseteq m_i$ and the equivalence $\cong$ over $m'_0 + m'_1$ defined there. Now apply Construction 7.11 to create a pair $\vec{C} : \vec{m} \rightarrow n$, by choosing sets $\vec{\ell}$ and elisions $\vec{\eta}$ as follows:

$$\begin{aligned} \ell_i \; &\overset{\text{def}}{=} \; \{r \in m_i \mid r \text{ barren in } A_i, \; D_i(r) \in V_{\bar{\imath}}\} \\ \eta_i : \ell_i \rightarrow V_{\bar{\imath}} \; &\overset{\text{def}}{=} \; D_i \restriction \ell_i \, . \end{aligned}$$

Then indeed $\vec{C}$ coincides with $\vec{B}$. To prove this, first show that k'_0, k'_1 and $\simeq$ in the IPO construction coincide with m'_0, m'_1, $\cong$ in the RPO construction; hence the codomain n of $\vec{C}$ coincides with the codomain m of $\vec{B}$. Then show that the parent maps C_i coincide with B_i. Thus every IPO is a bound built by Construction 7.11.

($\Leftarrow$) To prove the converse, consider any bound $\vec{C}\colon \vec{m} \to n$ built by Construction 7.11, for some sets $\vec{\ell}$ and elisions $\vec{\eta}$. Now apply Construction 7.7 to yield an RPO $(\vec{B}, B)$ for $\vec{A}$ to $\vec{C}$.

Then indeed $\vec{B}$ coincides with $\vec{C}$. To prove this, first show that m'_0, m'_1 and $\cong$ in the RPO construction coincide with k'_0, k'_1, $\simeq$ in the IPO construction; hence the codomain of $\vec{B}$ coincides with the codomain n of $\vec{C}$. Then show that the parent maps B_i coincide with C_i. Thus every bound built by Construction 7.11 is an IPO. ■

We shall finish this section by introducing an important subprecategory of place graphs, motivated by development to be studied in Section 12.

Definition 7.13 (hard place graphs) A *hard* place graph is one in which no root or non-atomic node is barren. They form a sub-precategory denoted by $\textsc{Plg}_\mathsf{h}$. ■

The condition on roots ensures that hard place graphs are epi. This means, as we have seen, that a consistent pair always has a unique IPO, i.e. a pushout. The extra condition, that a non-atomic node must not be barren, makes some of the mathematics simpler; for example, if $B \circ A$ is hard then so are both A and B. Moreover, no change to the IPO (or pushout) construction is needed, as we now see:

Proposition 7.14 (pushouts for hard place graphs) *If $\vec{A}$ is a consistent pair of hard place graphs, then the pushout $\vec{B}$ built in* $\acute{}\textsc{Plg}$ *by Construction 7.11 is also hard, and is indeed a pushout in* $\textsc{Plg}_\mathsf{h}$.

There is another connection between $\acute{}\textsc{Plg}$ and $\acute{}\textsc{Plg}_\mathsf{h}$. Let $\mathcal{K}$ be any signature, and choose a new atomic control $\triangle$ with zero arity; adjoin $\triangle$ to $\mathcal{K}$ to form $\mathcal{K}^\triangle$. We can make any arrow G of $\acute{}\textsc{Plg}(\mathcal{K})$ into a hard place graph in $\acute{}\textsc{Plg}_\mathsf{h}(\mathcal{K}^\triangle)$ by adding a $\triangle$-node as a child of any barren root or node. We shall call $\triangle$-nodes *place nodes*. Now let us say that two bigraphs F and G in $\acute{}\textsc{Plg}_\mathsf{h}(\mathcal{K}^\triangle)$ are *place-equivalent*, $F \equiv_\triangle G$, if they differ only in occurrences of place nodes. Then place equivalence is a static congruence (Definition 3.5). Also there exists a quotient precategory $\acute{}\textsc{Plg}_\mathsf{h}(\mathcal{K}^\triangle)/{\equiv_\triangle}$, whose arrows are place-equivalence classes of hard place graphs, and where the support of each equivalence class is just the support of each member less its place nodes. Furthermore this quotient precategory is isomorphic with $\acute{}\textsc{Plg}(\mathcal{K})$.

The reader may safely omit the rest of this section until reading Section 14. Until then we shall work with hard place graphs because their IPOs are pushouts, which helps to avoid elisions. But at that point we need to invoke place equivalence in order to forget place nodes. For this purpose, in order to transfer our results to an abstract setting, we need to know certain properties of $\equiv_\triangle$. We prepare for this now by showing that the operation which generates $\equiv_\triangle$ —i.e. the addition or removal of a single place node $\triangle_u$— does not affect the pushout property, under certain conditions.

In the following two propositions, for ease of notation, we shall use $\triangle$ to mean a fresh place node $\triangle_u$ distinct from all others present. The proofs appear in the Appendix.

Proposition 7.15 (first pushout variation) *Let $\vec{B}$ be a bound for $\vec{A}$ in* $\acute{}\textsc{Plg}_\mathsf{h}(\mathcal{K}^\triangle)$*. 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 $(A_0^\triangle, A_1)$.*

(a)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
 & {} \arrow[ul, "A_0"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1^{\triangle}"'] \arrow[ul, "B_1^{\triangle}"] \\
 & {} \arrow[ul, "A_0^{\triangle}"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

There are other ways of adding a single place node to a square consisting of a bound $\vec{B}$ for $\vec{A}$, and preserving the bound. Which of these ways will preserve the pushout property in both directions, as in the last proposition? If we add $\triangle$ to both B_0 and B_1 then we *lose* the pushout; for we violate the IPO property that the nodes of an IPO for $\vec{A}$ must be among the nodes of $\vec{A}$. What about adding $\triangle$ to both A_0 and A_1? In this case we may *gain* a pushout where only a bound previously existed. This is best seen in reverse; if $\vec{B}$ is a pushout for the augmented pair $\vec{A^{\triangle}}$, then by deleting $\triangle$ from the latter we may remove the only shared node, thus leaving a merging of roots that should not occur in a pushout.

However, by adding a constraint we obtain the following:

Proposition 7.16 (second pushout variation) *Let $\vec{B}$ be a bound for $\vec{A}$ in $\acute{\ }\mathrm{PLG}_{\mathsf{h}}(\mathcal{K}^{\triangle})$. 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 $\vec{\mathbf{A}}_n$, it is also a pushout for $\vec{A}$.

(a)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
 & {} \arrow[ul, "A_0"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
 & {} \arrow[ul, "A_0^{\triangle}"] \arrow[ur, "A_1^{\triangle}"'] &
\end{tikzcd}$$

8 Link graphs

Link graphs capture the connectivity of bigraphs, ignoring their nesting. The treatment here is significantly simpler than the previous treatment [28], though compatible with it.[6] There is a close formal analogy in the treatment of place graphs and link graphs.

As with place graphs, we assume a signature $\mathcal{K}$ assigning to each *control* K an arity $ar(K)$. We also assume an infinite set $\mathcal{X}$ of *names*.

Definition 8.1 (link graph) A *link graph* $A = (V, E, \mathit{ctrl}, \mathit{link}) : X \to Y$ has finite sets X of *inner names*, Y of *(outer) names*, V of *nodes* and E of *edges*. It also has a function $\mathit{ctrl} : V \to \mathcal{K}$ called the *control map*, and a function $\mathit{link} : X \uplus P \to E \uplus Y$ called the *link map*, where $P \overset{\text{def}}{=} \sum_{v \in V} ar(\mathit{ctrl}(v))$ is the set of *ports* of A.

We shall call the inner names X and ports P the *points* of A, and the edges E and outer names Y its *links*. ■

The outer and inner names are for interfacing, and will be important in defining composition. When we talk of a 'name' without adjective, we mean an outer name.

Here are some basic properties:

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. ■

An idle *name* is sometimes needed; for example we may want to consider two bigraphs as members of the same homset, even if one of them uses a name x and the other does not. On the other hand an idle *edge* serves no useful purpose, but may be created by composition. Sometimes we shall need to ensure that the property of leanness (no idle edges) is preserved by certain constructions.

Definition 8.3 (precategory of link graphs) The precategory ´LIG 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 $A_i = (V_I, \mathit{ctrl}_i, E_i, \mathit{link}_i) : X_i \to X_{i+1}$ $(i = 0, 1)$ is defined when their node sets and edge sets are disjoint; then $A_1 \circ A_0 \overset{\text{def}}{=} (V, \mathit{ctrl}, E, \mathit{link})$ where $V = V_0 \uplus V_1$, $\mathit{ctrl} = \mathit{ctrl}_0 \uplus \mathit{ctrl}_1$, $E = E_0 \uplus E_1$ and $\mathit{link} = (\mathsf{Id}_{E_0} \uplus \mathit{link}_1) \circ (\mathit{link}_0 \uplus \mathsf{Id}_{E_1})$. The identity link graph at X is $\mathsf{id}_m \overset{\text{def}}{=} (\emptyset, \emptyset_{\mathcal{K}}, \emptyset, \mathsf{Id}_X) : X \to X$. ■

Note that ´LIG is supported, with node-edge sets $V + E$ as support sets.

We can describe the composite link map link of $A_1 \circ A_0$ as follows, considering all possible arguments $p \in X_0 \uplus P_0 \uplus P_1$:

$$\mathit{link}(p) = \begin{cases} \mathit{link}_0(p) & \text{if } p \in X_0 \uplus P_0 \text{ and } \mathit{link}_0(p) \in E_0 \\ \mathit{link}_1(x) & \text{if } p \in X_0 \uplus P_0 \text{ and } \mathit{link}_0(p) = x \in X_1 \\ \mathit{link}_1(p) & \text{if } p \in P_1 \ . \end{cases}$$

[6] The main difference is that we here give identity not only to the nodes, but also to the links, in a link graph. Using our present terminology, we previously defined links in terms an equivalence over points and names, each link being an equivalence class. This avoided introducing an explicit link set, but the manipulation of equivalences required to exhibit and characterise RPOs and IPOs was much harder than with the present treatment.

By analogy with place graphs, we often denote the link map of A simply by A.

Proposition 8.4 (isomorphisms in link graphs) *An arrow $\iota : X \to Y$ in ´LIG is an isomorphism iff it has no nodes or edges and its link map is a bijection from X to Y.*

Note that the names in an interface are identified alphabetically, not positionally. This difference is mathematically unimportant. Alphabetical names are convenient for link graphs just as they are convenient in the λ-calculus, and they also lead naturally to forms of parallel product that are familiar from process calculi. But in defining tensor product we have to require disjoint interfaces:

Definition 8.5 (tensor product) The *tensor product* $\otimes$ in ´LIG is defined as follows: On objects, $X \otimes Y$ is simply the union of sets required to be disjoint. For two link graphs $A_i : X_i \to Y_i$ $(i = 0, 1)$ we take $A_0 \otimes A_1 : X_0 \otimes X_1 \to Y_0 \otimes Y_1$ to be defined when the interface products are defined and when A_0 and A_1 have disjoint node sets and edge sets; then we take the union of the link maps. ■

There is an important variant of tensor product that merges outer names, i.e. does not require them to be disjoint. This has fewer algebraic properties than the tensor (categorically, it is not a bifunctor), but will be important in modelling process calculi:

Definition 8.6 (parallel product) The *parallel product* $\mid$ in ´LIG is defined as follows: On objects, $X \mid Y \overset{\text{def}}{=} X \cup Y$. On link graphs $A_i : X_i \to Y_i$ $(i = 0, 1)$ we define $A_0 \mid A_1 : X_0 \otimes X_1 \to Y_0 \mid Y_1$ whenever X_0 and X_1 are disjoint, by taking the union of link maps. ■

Again we shall need epis and monos, and we have the following:

Proposition 8.7 (epis and monos in link graphs) *A link graph is epi iff no name is idle; it is mono iff no two inner names are peers.*

Notation When considering a pair $\vec{A} : W \to \vec{X}$ of link graphs with common domain W, we shall adopt a convention for naming their nodes, ports and edges. We denote the node set of A_i $(i = 0, 1)$ by V_i, and denote $V_0 \cap V_1$ by V_2. We shall use $v_i, v_i', \ldots$ to range over V_i $(i = 0, 1, 2)$. Similarly we use $p_i \in P_i$ and $e_i \in E_i$ for ports and edges $(i = 0, 1, 2)$. However, we shall sometimes use p_i also for points, i.e. $p_i \in W \uplus P_i$; the context will resolve any ambiguity. ■

One of the reasons for equipping link graphs with explicit edge sets, as well as node sets, is that we get a simple RPO theory. Also, as the reader will have noticed, there is a striking formal analogy between link graphs and place graphs. On closer inspection, the analogy appears to break down. For a parent map is $\mathit{prnt}\colon h \uplus V \to V \uplus m$ where both the domain and codomain include the nodes V, while a link map is $\mathit{link}\colon W \uplus P \to E \uplus X$ where the sets P and E are disjoint; so unlike a parent map, a link map cannot be iterated, i.e. a link graph has no notion of *nesting*. Nonetheleless, The RPO theories are almost identical, and we present them as similarly as possible.

We first give the same intuition as for place graphs. Suppose $\vec{D}$ is a bound for $\vec{A}$, and we wish to construct the RPO $(\vec{B}, B)$. To form $\vec{B}$, we first truncate $\vec{D}$ by removing the names, and all points and edges not present in $\vec{A}$. Then for the outer face of $\vec{B}$, we create a new link (a name) for each point unlinked by the truncation, equating these new names only when required so that $B_0 \circ A_0 = B_1 \circ A_1$. Formally:

Construction 8.8 (RPOs in link graphs) An RPO $(\vec{B}\colon \vec{X} \to \hat{X}, B\colon \hat{X} \to Z)$, for a pair $\vec{A}\colon W \to \vec{X}$ of link graphs relative to a bound $\vec{D}\colon \vec{X} \to Z$, will be built in three stages. Since RPOs are preserved by isomorphism, we assume X_0, X_1 disjoint. We use the notational conventions introduced above.

nodes and edges: If V_i are the nodes of A_i $(i = 0, 1)$ then the nodes of D_i are $V_{\bar{\imath}} \backslash V_2 \uplus V_3$ for some V_3. Define the nodes of B_i and B to be $V_{\bar{\imath}} \backslash V_2$ $(i = 0, 1)$ and V_3 respectively. Edges E_i are treated exactly analogously, and ports P_i inherit the analogous treatment from nodes.

interface: Construct the shared codomain $\hat{X}$ of $\vec{B}$ as follows. First, define the names in each X_i that must be mapped into $\hat{X}$:

$$X'_i \overset{\text{def}}{=} \{x \in X_i \mid D_i(x) \in P_3 \uplus Z\} \; .$$

Next, on the disjoint sum $X'_0 + X'_1$, define $\cong$ to be the smallest equivalence for which $(0, x_0) \cong (1, x_1)$ whenever $A_0(p) = x_0$ and $A_1(p) = x_1$ for some point $p \in W \uplus P_2$. Then define the codomain up to isomorphism:

$$\hat{X} \overset{\text{def}}{=} (X'_0 + X'_1)/{\cong} \; .$$

For each $x \in X'_i$ we denote the $\cong$-equivalence class of (i, x) by $\widehat{i, x}$.

links: Define B_0 to simulate D_0 as far as possible (B_1 is similar):

$$\begin{array}{lll}
\text{For } x \in X_0 : & B_0(x) \overset{\text{def}}{=} & \begin{cases} \widehat{0, x} & \text{if } x \in X'_0 \\ D_0(x) & \text{if } x \notin X'_0 \end{cases} \\
\text{For } p \in P_1 \backslash P_2 : & B_0(p) \overset{\text{def}}{=} & \begin{cases} \widehat{1, x} & \text{if } A_1(p) = x \in X_1 \\ D_0(p) & \text{if } A_1(p) \notin X_1 \; . \end{cases}
\end{array}$$

Finally define B, to simulate both D_0 and D_1:

$$\begin{array}{ll}
\text{For } \hat{x} \in \hat{X} : & B(\hat{x}) \overset{\text{def}}{=} D_i(x) \;\; \text{where } x \in X_i \text{ and } \widehat{i, x} = \hat{x} \\
\text{For } p \in P_3 : & B(p) \overset{\text{def}}{=} D_i(p) \; .
\end{array}$$

■

This definition can be proved sound; for it can be shown that the right-hand sides in the clauses defining link maps B_i and B are well-defined links in B_i and B respectively. Then we can prove the following (the proof appears in Appendix A.2):

Theorem 8.9 (RPOs in link graphs) *In ´LIG, Whenever a pair $\vec{A}$ of link graphs has a bound $\vec{D}$, there exists an RPO $(\vec{B}, B)$ for $\vec{B}$ to $\vec{D}$, and Construction 8.8 yields such an RPO.*

We now proceed to characterise all the IPOs for a given pair $\vec{A}\colon W \to \vec{X}$ of link graphs, just as we did for place graphs. Fortunately, the formal analogy between the two allows us to omit proofs, but we shall exhibit the construction in full.

Again we ask: how does our link graph RPO $(\vec{B}, B)$ vary, when we keep $\vec{A}$ fixed but vary the given bound $\vec{D}$? The answer is the same: if $\vec{A}$ are both epi, then $\vec{B}$ remains fixed and only B varies, so that in this case there is a pushout. But, as with place graphs, we need to treat the general case. The first step is to establish consistency conditions.

Definition 8.10 (consistency conditions for link graphs) We define three *consistency* conditions on a pair $\vec{A} : W \to \vec{X}$ of place graphs. We use p to 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 $\quad ctrl_0(v_2) = ctrl_1(v_2)$
CL1 $\quad$ If $A_i(p) \in E_2$ then $p \in W \uplus P_2$ and $A_{\bar{\imath}}(p) = A_i(p)$.
CL2 $\quad$ If $A_i(p_2) \in E_i \setminus E_2$ then $A_{\bar{\imath}}(p_2) \in X_{\bar{\imath}}$, and if also $A_{\bar{\imath}}(p) = A_{\bar{\imath}}(p_2)$
$\quad$ then $p \in W \uplus P_2$ and $A_i(p) = A_i(p_2)$. ■

Again, let us express CL1 and CL2 in words. If $i = 0$, CL1 says that if the link of any point p in A_0 is closed and shared with A_1, then p is also shared and has the same link in A_1. CL2 says, on the other hand, that if the link of a shared point p_2 in A_0 is closed and *unshared*, then its link in A_1 must be open, and further that any peer of p_2 in A_1 must also be its peer in A_0.

Proposition 8.11 (consistency in link graphs) *If the pair $\vec{A}$ has a bound, then the consistency conditions hold.*

Before going further, it may be helpful to see a simple example.

Example 8 (consistent link graphs) Consider the pair $\vec{A}\colon \emptyset \to \vec{X}$ of link graphs in Figure 10, where $X_0 = \{x_0, y_0, z_0\}$ and $X_1 = \{x_1, y_1\}$. Nodes with subscript 2 are shared. (Controls are not shown). The pair is consistent, with bound $\vec{B}$ as shown. It is worth checking the consistency conditions. ■

Now, assuming the consistency conditions of Definition 8.10, we shall construct a nono-empty family of IPOs for $\vec{A}$; the construction exactly follows the analogy with place graphs. As before, it is clear that when $\vec{A}$ are both epi there are no elisions, and hence the IPO is unique and hence pushout.

Construction 8.12 (IPOs in link graphs) Assume the consistency conditions for the pair of link graphs $\vec{A} : W \to \vec{X}$. We define a family of IPOs $\vec{C}\colon \vec{X} \to Y$ for $\vec{A}$ as follows.

nodes and edges: Take the nodes and edges of C_i to be $V_{\bar{\imath}} \setminus V_2$ and $E_{\bar{\imath}} \setminus E_2$.

interface: For $i = 0, 1$ choose any subset L_i of the names X_i such that all members of L_i are idle. Set $K_i = X_i \setminus L_i$. Define $K_i' \subseteq K_i$, the names to be mapped to the codomain Y, by

$$K_i' \stackrel{\text{def}}{=} \{x_i \in K_i \mid \forall p \in P_2.\, A_i(p) = x_i \Rightarrow A_{\bar{\imath}}(p) \in X_{\bar{\imath}}\} .$$

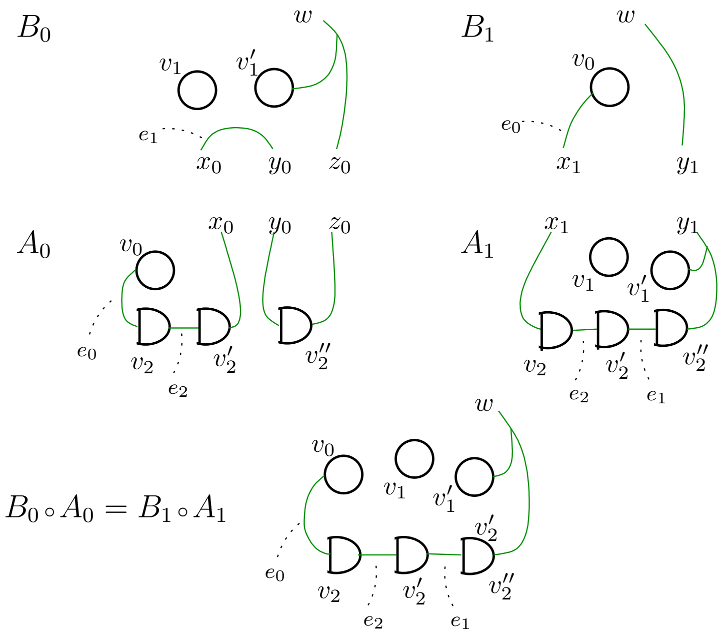


Figure 10: A consistent pair $\vec{A}$ of link graphs, with bound $\vec{B}$

Next, on the disjoint sum $K_0' + K_1'$, define $\simeq$ to be the smallest equivalence such that $(0, x_0) \simeq (1, x_1)$ whenever $A_0(p) = x_0$ and $A_1(p) = x_1$ for some $p \in W \uplus P_2$. Then define the codomain up to isomorphism:

$$Y \stackrel{\text{def}}{=} (K_0' + K_1')/\simeq\ .$$

For each $x \in K_i'$ we denote the $\simeq$-equivalence class of (i, x) by $\widehat{i, x}$.

links: Choose two arbitrary functions $\eta_i : L_i \to E_{\bar{\imath}} \backslash E_2$ $(i = 0, 1)$. Then define the link maps $C_i : X_i \to Y$ as follows (we give C_0; C_1 is similar):

For $x \in X_0$:

$$C_0(x) \stackrel{\text{def}}{=} \begin{cases} \widehat{0, x} & \text{if } x \in K_0' \\ A_1(p) & \text{if } x \in K_0 \backslash K_0', \text{ for } p \in W \uplus P_2 \text{ with } A_0(p) = x \\ \eta_0(x) & \text{if } x \in L_0 \end{cases}$$

For $p \in P_1 \backslash P_2$:

$$C_0(p) \stackrel{\text{def}}{=} \begin{cases} \widehat{1, x} & \text{if } A_1(p) = x \in X_1 \\ A_1(p) & \text{if } A_1(p) \notin X_1\ . \end{cases}$$

■

The maps η_i are called *elisions*; this refers to the fact that the idle names L_i in A_i are not exported in the IPO interface Y, but instead mapped into the body of C_i. There is a distinct IPO for each choice of L_i and η_i. However the IPO will be unique if

$L_i = \emptyset$ is forced. This can happen for one of two reasons: either A_i has no idle names; or $E_{\overline{\imath}} \backslash E_2$ is empty (i.e. all edges of $A_{\overline{\imath}}$ are shared), so no elision can exist.

The soundness of the above definition, and the fact that $\vec{C}$ is a bound, are both established by analogy with the corresponding results for place graphs. Similarly the following characterisation theorem, stating that our construction creates all and only IPOs for $\vec{A}$, is proved analogously to Theorem 7.12 for place graphs:

Theorem 8.13 (characterising IPOs for link graphs) *A pair $\vec{C} : \vec{X} \to Y$ is an IPO for $\vec{A} : W \to \vec{X}$ iff it is generated (up to isomorphism) by Construction 8.12.*

9 Pure bigraphs: development

We now develop the theory of pure bigraphs, based upon Definition 6.2. First we introduce the obvious precategory by combining ´PLG and ´LIG:

Definition 9.1 (precategory of pure concrete bigraphs) The precategory ´BIG($\mathcal{K}$) of pure concrete bigraphs over a signature $\mathcal{K}$ has pairs $I = \langle m, X\rangle$ as objects (*interfaces*) and bigraphs $G = (V, E, ctrl_G, G^{\mathsf{P}}, G^{\mathsf{L}}) \colon I \to J$ as arrows (*contexts*). We call I the *inner face* of G, and J the *outer face*. If $H : J \to K$ is another bigraph with node set disjoint from V, then their composition is defined directly in terms of the composition of the components as follows:

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

The identities are $\langle \mathsf{id}_m, \mathsf{id}_X\rangle : I \to I$, where $I = \langle m, X\rangle$. The subprecategory ´BIG$_{\mathsf{h}}$ consists of *hard* bigraphs, those with place graphs in ´PLG$_{\mathsf{h}}$. ■

Throughout this section, unless otherwise stated the definitions and results apply equally to ´BIG and ´BIG$_{\mathsf{h}}$. The whole section is about pure bigraphs, in contrast to the *binding bigraphs* to be studied in Section 11, so we shall omit the adjective 'pure' here. We shall also omit 'concrete' for the time being; but in Definition 9.12 we shall introduce *abstract* bigraphs, via a forgetful functor. We shall continue to omit the signature $\mathcal{K}$ except when it is important. We now combine some familiar place graph and link graph structures to yield bigraph structures.

Proposition 9.2 (isomorphisms in bigraphs) *The isomorphisms in* ´BIG *are all combinations $\iota = \langle \iota^{\mathsf{P}}, \iota^{\mathsf{L}}\rangle$ of a place graph isomorphism and a link graph isomorphism.*

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

$$G_0 \otimes G_1 \stackrel{\text{def}}{=} \langle G_0^{\mathsf{P}} \otimes G_1^{\mathsf{P}}, G_0^{\mathsf{L}} \otimes G_1^{\mathsf{L}}\rangle \colon I_0 \otimes I_1 \to J_0 \otimes J_1$$

when the interfaces exist and the node sets are disjoint. This combination is well-formed, since its constituents share the same node set. ■

It is routine to verify the axioms for a partial tensor product. In fact bigraphs are an instance of a mathematical structure that we introduced in Section 4:

Theorem 9.4 (bigraphs are wide monoidal) *For any signature $\mathcal{K}$, the precategories ´BIG($\mathcal{K}$) and ´BIG$_{\mathsf{h}}$($\mathcal{K}$) are wide monoidal; the origin is $\epsilon = \langle 0, \emptyset\rangle$, and the interface $\langle n, X\rangle$ has width n.*

Proof We leave the details for the reader to check. Note first that ´BIG and ´BIG$_{\mathsf{h}}$ are supported, with support sets of the form $V + E$ — a disjoint sum of a node set and an edge set. The width functor on arrows (bigraphs) is given as follows: for G :

$\langle m, X\rangle \to \langle n, Y\rangle$, its width is the function sending each site $s \in m$ to the unique root $r \in n$ such that $r >_G s$. ■

Following Section 4 we use lower case latters $a, b, \ldots$ for *ground* bigraphs, those with inner face ϵ, and we write $a\colon \epsilon \to I$ as $a\colon I$. These will represent the agents of our BRSs in Section 12, and will be used to model the agents in a conventional process calculus in Section 15.

Several properties of bigraphs are inherited from place graphs and link graphs. For example:

Proposition 9.5 (epis and monos in bigraphs) *A bigraph G in ´BIG (or ´BIG$_\mathsf{h}$) is epi (resp. mono) iff its components G^P and G^L are epi (resp. mono) in ´PLG (or ´PLG$_\mathsf{h}$) and ´LIG.*

It follows from Theorem 9.4 that when we later equip bigraphs with reaction rules we shall have a WRS, and then we can apply the main congruence theorem, Theorem 5.5, provided that we have enough RPOs. So now we draw together our RPO results for place graphs and link graphs. We deduce from Theorem 7.8 and 8.9 the following:

Corollary 9.6 (RPOs for bigraphs) *In both ´BIG and ´BIG$_\mathsf{h}$ an RPO for $\vec{A}$ to $\vec{D}$ is provided by the triple*

$$(\,\langle B_0^\mathsf{P}, B_0^\mathsf{L}\rangle,\ \langle B_1^\mathsf{P}, B_1^\mathsf{L}\rangle,\ \langle B^\mathsf{P}, B^\mathsf{L}\rangle\,)$$

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

Proof We can check from Constructions 7.7 and 8.8 that each combination in the triple is well formed, since its two constituents have the same node set. Once this is established, the RPO property is easily verified by diagram chasing. ■

Now we shall consider IPOs for bigraphs. We can use Theorems 7.12 and 8.13 to prove that:

Corollary 9.7 (IPOs for bigraphs) *A pair $\vec{B}$ is an IPO for $\vec{A}$ in ´BIG or ´BIG$_\mathsf{h}$ iff $\vec{B}^\mathsf{P}$ is a place graph IPO for $\vec{A}^\mathsf{P}$ and $\vec{B}^\mathsf{L}$ is a link graph IPO for $\vec{A}^\mathsf{L}$.*

Proof It is enough to prove it just for ´BIG.

($\Rightarrow$) Assume the IPO in ´BIG. Then in ´PLG, by definition $\vec{B}^\mathsf{P}$ is a bound for $\vec{A}^\mathsf{P}$. We need to show that $(\vec{B}^\mathsf{P}, \mathsf{id})$ is an RPO for $\vec{A}^\mathsf{P}$ to $\vec{B}^\mathsf{P}$. So, for any other candidate RPO $(\vec{C}^\mathsf{P}, C^\mathsf{P})$, we must find a unique mediating arrow between the intended RPO and this candidate.

It can be shown that the members of $(\vec{C}^\mathsf{P}, C^\mathsf{P})$ have the same support as the members of $(\vec{B}^\mathsf{P}, \mathsf{id})$. So we may form the triple of combinations

$$(\,\langle C_0^\mathsf{P}, B_0^\mathsf{L}\rangle,\ \langle C_1^\mathsf{P}, B_1^\mathsf{L}\rangle,\ \langle C^\mathsf{P}, \mathsf{id}\rangle\,)$$

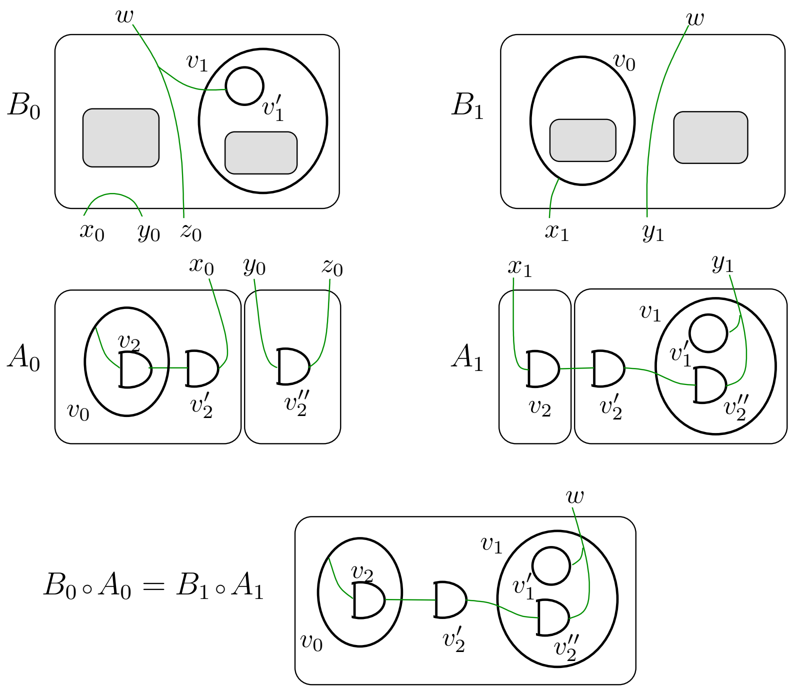


Figure 11: A consistent pair $\vec{A}$ of bigraphs, with IPO $\vec{B}$

(with suitable interfaces), and also prove it to be a candidate RPO in ´BIG for $\vec{A}$ to $\vec{B}$. Hence there is a unique mediating arrow between the given RPO $(\vec{B}, \mathsf{id})$ and this candidate. The place graph constituent of this mediator then provides the required unique mediator in ´PLG, and we are done. A similar argument applies also to ´LIG.

($\Leftarrow$) Assuming IPOs in ´PLG and ´LIG, by routine diagram chasing we can verify the IPO property in ´BIG. ■

Example 9 (Bigraph IPOs) To illustrate IPOs in ´BIG, we can combine Example 7 for place graphs and Example 8 for link graphs, since they have the same node sets. In both cases the bounds $\vec{B}$ are IPOs, and indeed pushouts because the graphs $\vec{A}$ are epi. The combination is shown in Figure 11. Again, both of the bigraphs $\vec{A}$ are epi, so our results show that the bound $\vec{B}$ is again an IPO and a pushout. ■

We now give a few special cases of IPOs. First, some pushouts (hence also IPOs) that are easy to verify for any precategory:

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

Next, tensor product preserves IPOs with disjoint support:

Proposition 9.9 (tensor IPO) *In any of* ´PLG*,* ´PLG$_\mathsf{h}$*,* LIG*,* ´BIG *or* ´BIG$_\mathsf{h}$*, 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 $\vec{A} \otimes \vec{B}$.*

An important corollary, with the help of Proposition 9.8, is when the two given IPOs contain identities:

(a)

$$\begin{tikzcd}
I \otimes J' \arrow[r, "\mathsf{id}_I \otimes B"] & I \otimes J \\
\arrow[u, "A \otimes \mathsf{id}_{J'}"] \arrow[r, "\mathsf{id}_{I'} \otimes B"'] & I' \otimes J \arrow[u, "A \otimes \mathsf{id}_J"']
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
I \arrow[r, "\mathsf{id}_I \otimes b"] & I \otimes J \\
\arrow[u, "a"] \arrow[r, "b"] & J \arrow[u, "a \otimes \mathsf{id}_J"']
\end{tikzcd}$$

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

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

We shall call a bigraph *lean* if its link graph is lean, i.e. has no idle edges. In Section 12 we shall need to transform IPOs by the addition or subtraction of idle edges. Let us write A^E for the result of adding a set E of fresh idle edges to A. The following is easy to prove from the IPO construction for link graphs:

Proposition 9.11 (IPOs, idle edges and leanness) *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).

We now turn to abstract bigraphs. To get them from concrete bigraphs, we wish to factor out the identity of nodes and edges; we also wish to forget any idle edges. So we define an equivalence $\Bumpeq$ that is a little coarser than support equivalence ($\widehat{=}$):

Definition 9.12 (Abstract pure bigraphs and their category) Two concrete bigraphs A and B are *lean-support equivalent*, written $A \Bumpeq B$, if after discarding any idle edges they are support equivalent. The category BIG$(\mathcal{K})$ of *abstract pure bigraphs* has the same objects as ´BIG$(\mathcal{K})$, and its arrows are lean-support equivalence classes of concrete bigraphs. Lean-support equivalence is clearly a static congruence (Definition 3.5). The associated quotient functor, assured by Definition 3.6, is

$$[\![\cdot]\!] : \acute{}\textsc{Big}(\mathcal{K}) \to \textsc{Big}(\mathcal{K}) \, .$$

The definition of BIG$_\mathsf{h}$ is analogous, with the restriction of $[\![\cdot]\!]$ to ´BIG$_\mathsf{h}$ as quotient functor. ■

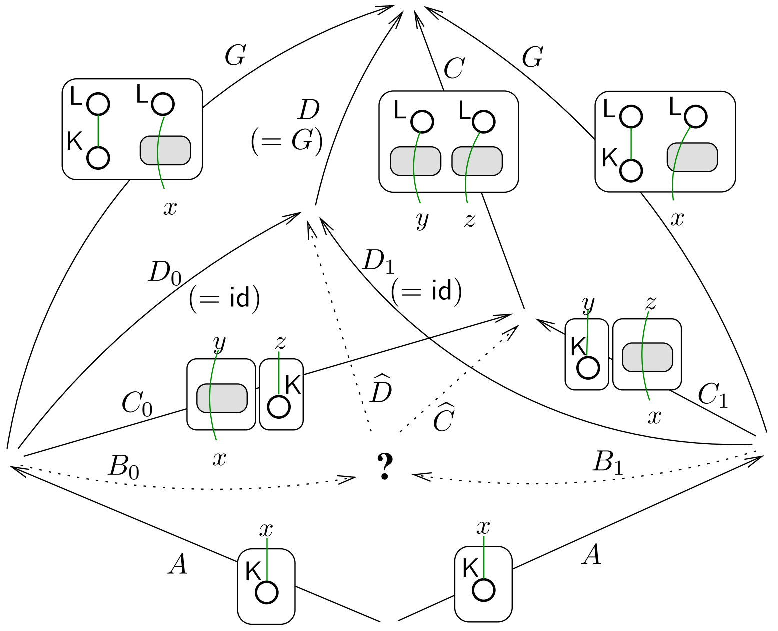


Figure 12: Two abstract bigraphs may lack an RPO

Note that there are natural abstract versions of place graphs and link graphs. But we have little use for them, for we cannot combine an abstract place graph with an abstract link graph to form an abstract bigraph! (The combination only makes sense when nodes have identity.) The deeper reason for studying concrete bigraphs is that they possess RPOs. This will allow us Section 12 to derive a behavioural congruence for ´Big, and then to show how to transfer it, under certain assumptions, to Big.

To see why we cannot work directly in Big, let us see how it lacks some of the structure present in ´Big. A simple example of this is that the functor $[\![\cdot]\!]$ loses epis; for example, the abstract bigraph A in Figure 12 is not epi, though (since it has no idle names) all its $[\![\cdot]\!]$-preimages are epi. More seriously, Big lacks RPOs in general; this we shall now show.

Example 10 (abstract bigraphs lack RPOs) Let the controls K and L be atomic with arity 1. Figure 12 shows two candidate RPOs for the pair (A, A) of abstract bigraphs w.r.t. the pair (G, G). The candidates are $(\vec{C}, C)$ and $(\vec{D}, D)$, where $D_i = \mathsf{id}$ $(i = 0, 1)$ and $D = G$. Speaking informally, the first candidate keeps the two K-nodes in (A, A) distinct, whereas the second candidate coalesces them. These two treatments of node-occurrences cannot be properly distinguished in abstract bigraphs; that is why an RPO fails to exist in this example.

To see this, suppose an RPO $(\vec{B}, B)$ exists. Then there must be mediators $\widehat{C}$, $\widehat{D}$ to the two candidates, as shown, making the diagram commute. But this leads to a contradiction, as follows. First, B_0 and B_1 must have empty support, since for example

$\widehat{D} \circ B_0 = D_0 = \mathsf{id}$. From $B_0 \circ A = B_1 \circ A$ it can then be deduced that $B_0 = B_1$.[7] It follows that $C_0 = \widehat{C} \circ B_0 = \widehat{C} \circ B_1 = C_1$, a contradiction. ■

We shall finish this section by introducing some terminology and operations that will be needed in following sections.

Notations and terminology We often abbreviate an interface $\langle 0, X\rangle$ to X, and $\{x\}$ to x; similarly we abbreviate $\langle m, \emptyset\rangle$ to m. Thus the interfaces $\emptyset$ and 0 are identical with the origin ϵ, and indeed the identity id_ϵ may be written variously as ϵ, $\emptyset$ or 0.

A bigraph with interfaces of zero width (and hence having no nodes) is called a *wiring*. The wirings ω are generated by composition and tensor product from two basic forms: $/x \colon x \to \epsilon$, called *closure*; and a function $\sigma \colon X \to Y$, not necessarily surjective, called a *substitution*. For $X = \{x_1, \ldots, x_n\}$ we write $/X$ for $/x_1 \otimes \cdots \otimes /x_n$, a multiple closure. For vectors $\vec{x}$ and $\vec{y}$ of equal length, with the x_i distinct, we write $\vec{y}/\vec{x}$ or $(y_0/x_0, y_1/x_1, \ldots)$ for the surjective substitution $x_i \mapsto y_i$. The unique substitution with empty domain and codomain Z is written $Z \colon \epsilon \to Z$. Every substitution σ can be expressed uniquely as $\sigma = \tau \otimes Z$, with τ surjective. We let α range over *renamings*, the bijective substitutions.

An interface is *prime* if it has width 1. We shall often write a prime interface $I = \langle 1, X\rangle$ as $\langle X\rangle$; note in particular that $1 = \langle\emptyset\rangle$. A *prime* bigraph $P \colon m \to \langle X\rangle$ has no inner names and a prime outer face. An important prime is $\mathit{merge} \colon m \to 1$; it has no nodes, and simply maps m sites to a single root. A bigraph $G \colon m \to \langle n, X\rangle$ with no inner names is converted by merge into a prime $(\mathit{merge} \otimes \mathsf{id}_X) \circ G$.

A bigraph is *discrete* if it has no edges, and its link map is bijective. This means that every point is open, no two points are peers, and no name is idle.

For any non-atomic control K with arity k and sequence $\vec{x}$ of k distinct names we define the discrete *ion* $K_{v,\vec{x}} \colon 1 \to \langle\vec{x}\rangle$ to have a single K-node v, whose ports are severally linked to $\vec{x}$. We omit the subscript v when it can be understood. For a discrete prime P with names Y the composite $(K_{\vec{x}} \otimes \mathsf{id}_Y) \circ P$ is a discrete *molecule*. If K is atomic it has no ion, but we define the discrete *atom* $K_{\vec{x}} \colon \epsilon \to \langle\vec{x}\rangle$; it resembles an ion but possesses no site. An arbitrary (non-discrete) ion, molecule or atom is gained by composing $\omega \otimes \mathsf{id}_1$ with a discrete one.

We often omit $\ldots \otimes \mathsf{id}_I$ in compositions, when there is no ambiguity; examples from above are $\mathit{merge} \circ G$ for $(\mathit{merge} \otimes \mathsf{id}_X) \circ G$ and $K_{\vec{x}} \circ P$ for $(K_{\vec{x}} \otimes \mathsf{id}_Y) \circ P$.

Given a wiring $\omega \colon Y \to Z$ we may restrict its link map to any subset $X \subseteq Y$, yielding the *restricted* wiring $\omega \restriction X \colon X \to Z$. Then, if the outer face of G is $\langle m, X\rangle$ we may write simply ωG for $(\omega \restriction X \otimes \mathsf{id}_m) \circ G$. ■

Note that every atom and every molecule is prime, but whereas an atom is ground, a molecule need not be (it can have sites). The reader may wonder why primes do not have inner names. This is what allows us to prove a prime factorisation property in Proposition 9.17(2).

We now look at variants of the tensor product, which reflect more closely the notion of 'parallel composition' familiar in process calculi. Although these operations apply

[7] This would be immediate if A were epi, but it is not! (Even though its representatives in ´BIG are epi.) However, a specific argument can be given in this case.

to arbitrary bigraphs, they are especially significant when applied to ground bigraphs, because these will model processes.

Process calculi often have a parallel product $p \,\|\, q$ or $p \,|\, q$, in which the processes p and q may share names. We therefore extend the parallel product '$\,|\,$' of link graphs (Definition 8.6) as follows:

Definition 9.13 (parallel product) The *parallel product* of two bigraphs is defined on interfaces by $\langle m, X\rangle \,\|\, \langle n, Y\rangle \overset{\text{def}}{=} \langle m+n, X \cup Y\rangle$, and on bigraphs by

$$G_0 \,\|\, G_1 \overset{\text{def}}{=} \langle G_0^{\mathsf{P}} \otimes G_1^{\mathsf{P}}, G_0^{\mathsf{L}} \,|\, G_1^{\mathsf{L}}\rangle \colon I_0 \otimes I_1 \to J_0 \,\|\, J_1$$

when the interfaces exist and the node sets are disjoint. ■

It is easy to verify that $\|$ is associative, with unit ϵ. We insist that G_0 and G_1 have disjoint *inner names*, this ensures that their parallel product is well-formed. Note that it keeps the regions of G_0 and G_1 separate; this was its purpose in the remote reaction rule for the π-calculus shown in Figure 6.

Another way of constructing $G_0 \,\|\, G_1$, which we shall use later in extending the product to binding bigraphs, is to disjoin the names of G_0 and G_1, then take the tensor product and merge the names again:

Proposition 9.14 (parallel product) *Let $G_0 \,\|\, G_1$ be defined. Then*

$$G_0 \,\|\, G_1 = \sigma(G_0 \otimes \tau G_1) \ ,$$

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 $\sigma(w_i) = \sigma(z_i) = z_i$ ($i \in n$).

We shall continue to use $\,|\,$ to combine two wirings; in fact $\omega_0 \,|\, \omega_1$ (as defined for link graphs) means the same as $\omega_0 \,\|\, \omega_1$. We shall also abuse notation by extending $\,|\,$ to primes:

Definition 9.15 (prime product) The *prime product* of prime interfaces is given by

$$\langle 1, X\rangle \,|\, \langle 1, Y\rangle \overset{\text{def}}{=} \langle 1, X \cup Y\rangle \ .$$

For two prime bigraphs $\vec{P} \colon \vec{I} \to \vec{J}$, with $I_0 \otimes I_1$ defined, we define their *prime product* by

$$P_0 \,|\, P_1 \ \overset{\text{def}}{=} \ \mathit{merge} \circ (P_0 \,\|\, P_1) \colon I_0 \otimes I_1 \to J_0 \,|\, J_1 \ . \quad ■$$

Again $\,|\,$ is associative, with unit 1 when applied to primes. We have chosen the symbol that is used in CCS and the π-calculus, since the correspondence will turn out to be exact. We shall find it useful to abuse notation even further, and write $\omega \,|\, P$ instead of $\omega \,\|\, P$ when joining a wiring to a prime. To summarise: $G_0 \,|\, G_1$ is defined whenever G_1 have width ≤ 1, and it has the maximum of their widths.

Let us now consider *discrete* bigraphs. In a precise sense they fully complement wiring:

Proposition 9.16 (underlying discrete bigraph) *Every bigraph G in ´BIG or ´BIG$_\mathsf{h}$ can be expressed uniquely (up to iso) as $G = (\omega \otimes \mathsf{id}_n) \circ D$, where ω is a wiring and D is discrete.*

We shall call this unique factorisation of G a *discrete normal form* (DNF). It applies equally to abstract bigraphs, and indeed it will play an important part in the complete axiomatisation of BIG which is the subject of a later section.

Discreteness is very well-behaved. It is clear that both composition and tensor product preserve it, and more:

Proposition 9.17 (synthesis and analysis of discrete bigraphs) *In ´BIG or ´BIG$_\mathsf{h}$ the discrete pure bigraphs form a monoidal sub-precategory. Moreover*

(1) Every discrete $D\colon \langle m, X\rangle \to \langle n, Y\rangle$ 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) If (D', G') is an IPO for (G, D) and D is discrete, then D' is discrete.

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

Note that a renaming is discrete but not prime (since it has zero width); this is why (1) has such a factor. This unique factorisation depends on the fact that primes have no inner names. In the special case that D is ground, the factorisation in (1) is just $D = d_0 \otimes \cdots \otimes d_{n-1}$, a product of prime discrete ground bigraphs.

We have to make one more preparation for Section 12 on dynamics. When we define parametric reaction rules we must allow them to replicate some of their parameters and discard others. We shall call this operation on parameters *instantiation*. The following definition ensures that names are shared between all copies of a parameter, and uses support translation to ensure that the several copies are given disjoint supports.

Definition 9.18 (instantiation) An *instantiation ϱ from (width) m to (width) n*, which we write $\varrho :: m \to n$, is determined by a function $\overline{\varrho}\colon n \to m$. For any X this function defines the map

$$\varrho\colon \mathsf{Gr}\langle m, X\rangle \to \mathsf{Gr}\langle n, X\rangle$$

as follows. Decompose $g\colon \langle m, X\rangle$ into $g = \omega(d_0 \otimes \cdots \otimes d_{m-1})$, with $\omega\colon Y \to X$ and each d_i prime and discrete. Then define

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

where $e_i \simeq d_{\overline{\varrho}(i)}$ for $i \in n$. This map is well-defined (up to support translation), by Propositions 9.16 and 9.17. ■

Note that the names of $e_0 \parallel \cdots \parallel e_{n-1}$ may be fewer than Y, because ϱ may not be surjective. But by our convention the outer names of $\varrho(g)$ are determined by the outer names of ω, i.e. X.

We deduce two important properties of wirings:

Proposition 9.19 (wiring an instance) *Wiring commutes with instantiation; that is,*

$$\omega\varrho(a) \simeq \varrho(\omega a) \; .$$

Proof Let $a\colon \langle m, X\rangle$, with $\varrho :: m \to m'$. Take the DNF $a = \omega' d$. Then $\varrho(a) = \omega' a'$, where $a' = d'_0 \parallel \cdots \parallel d'_{m'-1}$ with each $d'_i \simeq d_{\overline{\varrho}(i)}$. So

$$\begin{array}{rcl} \varrho(\omega a) & \simeq & \varrho(\omega(\omega' d)) \\ & = & \varrho((\omega \circ \omega')d) \\ & \simeq & (\omega \circ \omega')a' \\ & = & \omega(\omega' a') \\ & = & \omega\varrho(a) \; . \end{array}$$

■

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

$$\omega(F \parallel G) = \omega F \parallel \omega G \;\; \textit{and} \;\; \omega(F \mid G) = \omega F \mid \omega G \; .$$

Proof Routine. ■

We can now deduce how to apply instantiation to a product of primes:

Proposition 9.21 (instantiating a product) *Let $a_i\colon \langle Y_i\rangle$ be prime and ground ($i \in m$), and let $Y = \bigcup_i Y_i$. 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 $b_j \simeq a_{\overline{\varrho}(j)}$ for $j \in n$.

Proof First express each a_i in DNF, using discrete d_i with disjoint name sets. Then apply Propositions 9.19 and 9.20. ■

Thus, although instantiation breaks up a ground bigraph in general, it does not break up a prime; in fact, applied to a product of primes, it simply reassembles copies of the prime factors. Also, if we instantiate $G \circ a$ where a is prime, then a will not be broken up but the result may contain several copies of a. This fact, which will be important for Section 12, means that $\varrho(G \circ a)$ can be transformed into $\varrho(G \circ b)$ by replacing a finite number of occurrences of a by b. Formally:

Proposition 9.22 (instantiating with prime component) *Let $G\colon \langle X\rangle \to \langle m, Y\rangle$ 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 $\alpha_i\colon X \to X_i$ ($i \in k$), there exists a context $C\colon \langle k, \bigcup_i X_i\rangle \to \langle n, Y\rangle$ 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 $a, b\colon \langle X\rangle$ we have $(\varrho(G \circ a), \varrho(G \circ b)) \in \mathcal{S}^*$, where

$$\mathcal{S} = \{(H \circ a, H \circ b) \mid H \; \textit{any context}\} \; .$$

Proof For the first part, apply Propositions 9.16 and 9.17 to express G in terms of a product of prime discrete factors. Then use the fact that all but one of these factors is ground (since G has prime inner face) to obtain the equation

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

where π is a permutation, F has prime outer face and all of the right-hand side (except a) is independent of a. Finally we use Proposition 9.21 to obtain an expression for $\varrho(G \circ a)$ involving several support-disjoint copies of a, as required.

For the second part, define for each $i \in k$

$$c_i = C \circ (b_0 \otimes \cdots \otimes b_{i-1} \otimes a_i \otimes \cdots \otimes a_{k-1}) ,$$

so that c_i differs from c_{i+1} by the replacement of a single copy of a by b. For the required result we only need to observe that (c_i, c_{i+1}) is in $\mathcal{S}$, by choosing the context

$$E_i = C \circ (b_0 \otimes \cdots \otimes b_{i-1} \otimes \langle \alpha_i \rangle \otimes a_{i+1} \cdots \otimes a_{k-1}) . \qquad \blacksquare$$

We have now taken the theory of pure bigraphs as far as required for the dynamics of bigraphs, which we introduce in Section 12.

10 Algebra of pure bigraphs

In this section we diverge from our main theme, the behavioural theory of bigraphs, to sharpen our understanding of their algebraic structure. The reader may safely study later sections without reading this one.

Ths algebra we develop here is just for abstract bigraphs BIG. These are our primary model; we introduced concrete bigraphs ´BIG mainly to obtain – in Part III – a behavioural theory, which can then be transferred to BIG. However, it is likely that the algebraic theory for BIG will, with minor modifications, be valid also for ´BIG.

We shall find that there is a simple complete axiomatisation of pure bigraphs. There are also two useful kinds of normal form. One of them is in terms of discrete bigraphs, and is useful for proving the completeness of our axioms; the other uses the parallel products $\|$ and $|$, and is better fitted for practical applications. We begin by defining our algebraic signature (not to be confused with the control signature $\mathcal{K}$), consisting of elementary bigraphs sufficient to generate all bigraphs.

Elementary bigraphs We define six *elementary* forms of pure bigraph:

$/x$	:	$x \to \epsilon$	closure
$[f]$	:	$X \to Y$	substitution ($f \colon X \to Y$)
1	:	$\epsilon \to 1$	a barren root
merge	:	$2 \to 1$	map two sites to one root
$\gamma_{m,n}$	:	$m{+}n \to n{+}m$	swap m with n places
$K_{\vec{x}}$	:	$1 \to \langle 1, \vec{x}\rangle$	a discrete ion ($\vec{x}$ distinct).

We shall show that they generate all bigraphs by composition and tensor product.

The first two elements generate all wirings, i.e. the node-free link graphs, as explained in Section 9. The notation $[f]$ is introduced here for clarity, distinguishing a bigraphical substitution from its underlying functional substitution f. Recall that σ ranges over bigraphical substitutions, and α over renamings.

The next three elements generate all *placings*, i.e. the node-free place graphs. For example $\mathit{merge}_m \colon m \to 1$, which merges m sites, can be defined for all $m \geq 0$ by

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

Note that $\mathit{merge}_1 = \mathsf{id}$, and hence $\mathit{merge}_2 = \mathit{merge}$. Note also that the unit 1 is absent in hard bigraphs BIG$_\mathsf{h}$. We use $\pi : m \to m$ to range over *permutations*, those placings generated by the $\gamma_{m,n}$. Every *isomorphism* ι is the product $\pi \otimes \alpha$ of a renaming and a permutation. The usual symmetries of a strict symmetric monoidal category are defined by extending the place symmetries $\gamma_{m,n}$ as follows:

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

Finally, given all these node-free elements, we require only the discrete ions $K_{\vec{x}}$ to express everything in BIG. In particular, we can express a discrete atom as $K_{\vec{x}} \circ 1$.

Discrete normal forms The following proposition shows the expressive power of the elementary bigraphs. Further, it shows that every bigraph in Big can be expressed in a kind of normal form, called *discrete normal form* (DNF). We shall consistently use D, Q and N to stand for discrete, discrete prime and discrete molecular bigraphs.

Proposition 10.1 (discrete normal form) *In* Big *each bigraph G, discrete D, discrete prime Q and discrete molecule N can be expressed in discrete parts by an equation of the respective following form (recall that α is a renaming, π a permutation):*

$$\begin{array}{rcl}
G & = & (\omega \otimes \mathsf{id}_n) \circ D \\
D & = & \alpha \otimes ((Q_0 \otimes \cdots \otimes Q_{n-1}) \circ \pi) \\
Q & = & (\mathit{merge}_{n+p} \otimes \mathsf{id}_Y) \circ (\mathsf{id}_n \otimes N_0 \otimes \cdots \otimes N_{p-1}) \circ \pi \\
N & = & (K_{\vec{x}} \otimes \mathsf{id}_Y) \circ Q \ .
\end{array}$$

Moreover, the expression is unique up to certain isomorphisms on the parts.

By applying the equations to any bigraph expression G, we transform it into DNF; after applying the first two equations once, we apply the last two repeatedly. Note that the unit 1 occurs as a special case of Q when $n = p = 0$.

Axiomatisation We now address the question: What set of axioms is complete in the sense that every valid equation in terms of the elementary bigraphs is provable? The answer turns out to be rather simple; the axioms are shown in the table below.

Categorical axioms:

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

Link axioms:

$$\begin{array}{rcll}
[f] \circ [g] & = & [f \circ g] & \\
[f] \otimes [g] & = & [f \uplus g] & \\
/y \circ [f] & = & /X & (f\colon X \to y) \\
[f] \circ (x \otimes \mathsf{id}_X) & = & [f \upharpoonright X] & (f\colon x \otimes X \to Y)
\end{array}$$

Place axioms:

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

The categorical axioms are standard for a strict symmetric monoidal category. But note that the tensor product is defined only when interfaces have disjoint name sets; thus the equations are required to hold only when both sides are defined. What is remarkable is that no axioms at all are required on ions, which are the only non-empty elements. Thus bigraphs are a rather free structure.

Theorem 10.2 (Complete axiomatisation) *Two expressions, constructed from the elements by composition and tensor product, denote the same bigraph in* BIG *if and only if they are in the congruence generated by the axioms.*

Proof The proof of the theorem is quite detailed, and we only give a brief outline here. The 'if' direction, soundness, just requires an easy proof that each of the axioms is valid. The 'only if' direction, completeness, requires two steps. First we show, by induction on the structure of expressions, that the equality between an arbitrary expression and its DNF is provable from the axioms. Second, since DNFs are only unique up to certain isomorphisms, we show that the equality between isomorphic DNFs is also provable from the axioms. ■

Connected normal forms The discrete normal form (DNF) was important for our proof of completeness of the axioms. Moreover, the tensor product is heavily used in our axiomatisation; in particular its bifunctoriality

$$(A_1 \otimes B_1) \circ (A_0 \otimes B_0) = (A_1 \circ A_0) \otimes (B_1 \circ B_0)$$

plays a very important part.

On the other hand parallel products like $\|$ and $|$, which allow the sharing of names (so do not preserve discreteness) are found very natural in process calculi and in programming; the main purpose of a combinator such as $|$ in the π-calculus is to combine expressions that use the same channel, and which may therefore communicate via that channel, as in the reaction rule $\overline{x}y \,|\, x(z).P \to \{y/z\}P$. See all the examples in Section 2, where these combinators are used; see also the discussion of parametric reaction rules in Section 12 to follow.

In fact we can find a normal form that is in a sense opposite to DNF. Whereas in DNF we pull all wiring to the outermost, we can adopt instead the strategy of pushing it inwards as far as we can. This is achieved by using $\|$ and $|$ in place of $\otimes$, and by pushing closures $/Z$ inwards wherever possible. We call the result *connected normal form* (CNF); it is embodied in the following proposition, analogous to Proposition 10.1. We use P and M for primes and molecules (not necessarily discrete):

Proposition 10.3 (connected normal form) *In* BIG *each bigraph G, prime P and molecule M can be expressed by an equation of the respective following form (recall that σ is a substitution and π a permutation):*

$$\begin{array}{rcl} G &=& (/Z \otimes \mathsf{id}_n) \circ (\sigma \,\|\, ((P_0 \,\|\, \cdots \,\|\, P_{n-1}) \circ \pi) \\ P &=& (/Z \otimes \mathsf{id}_{m+n}) \circ (\mathsf{id}_m \,|\, M_0 \,|\, \cdots \,|\, M_{n-1}) \circ \pi \\ M &=& (/Z \otimes \mathsf{id}_1) \circ (K_{\vec{x}} \,|\, \mathsf{id}_Y) \circ P \end{array}$$

where, in each case, any member $z \in Z$ is a name of at least two members of the ensuing product ($\|$ or $|$). The names $\vec{x}$ need not be distinct. Moreover, in each case the expression is unique up to a renaming of Z and certain isomorphisms on the parts.

We regard CNF as important for practical use. For example, although we shall not do so, it is not hard to derive the CNF for $F \circ G$, $F \,\|\, G$ or $F \,|\, G$ from the CNFs for F and G. We shall also leave open whether there exists a pleasant axiomatisation that uses these parallel products in place of the tensor product.

11 Binding bigraphs

The reader who is interested in dynamics rather than in adding binding to bigraphs can safely skip this section and proceed to the dynamic theory in Section 12. He or she will also be able to read Sections 13 and 14 of Part III, interpreting them in pure bigraphs. However, binding is needed for Section 15 on the asynchronous π-calculus.

In Section 9 we studied a pure form of bigraph in which placing and linking are completely independent structures over a set of nodes. In doing so we found several roles played by a *place*: it may define a neighborhood within which reactions can be confined (e.g. in the ambient calculus); in contrast it may prevent reaction until its boundary is removed (e.g. in the π-calculus); it may define the site for a parameter of a reaction rule, thereby determining what may be replicated or discarded.

In this section we shall relax the independence of placing and linking by defining yet another role for a place: it may define the scope of a bound link. This will allow us to represent, among other things, the input prefix of the π-calculus (which binds a name). The first step we take is to enrich signatures.

Definition 11.1 (binding signature) A *binding signature* $\mathcal{K}$ is like a pure signature (Definition 6.1), except that the arity of a control $K\colon h \to k$ now consists of a pair of finite ordinals: the *binding arity* h and the *free arity* k, determining the number of *binding* and *non-binding* ports of any K-node. If K is atomic then $h = 0$. ■

For example, for the π-calculus controls, we have $\mathsf{get}\colon 1 \to 1$ and $\mathsf{send}\colon 0 \to 2$ (see Examples 2, 3, 4 and 6).

We wish to define a binding bigraph G in terms of an underlying pure one, in which all points linked to a binding port of a node u lie inside u. These points may be inner names as well as ports; to ensure that these inner names transmit the scope discipline to any other bigraph composed at the inner face of G, we enrich interfaces as follows:

Definition 11.2 (binding interface) A *binding interface* takes the form $I = \langle m, \vec{k}, X\rangle$, where the width m and the *global names* X are as before, and the new component $\vec{k} = k_0, \ldots, k_{m-1}$ is a sequence of finite ordinals, indexing for each $r \in m$ a set $\{(r, j) \mid j \in k_r\}$ of *local names*; we say these are *located at* r.

Denote the local names of I by X^ℓ and set $X^{\mathsf{u}} = X^\ell \uplus X$. Then we call $I^{\mathsf{u}} = \langle m, X^{\mathsf{u}}\rangle$ the pure interface *underlying* I. ■

We call an interface *global* if all its names are global. Note that we represent local names positionally, but continue to represent global names alphabetically. We are now ready for the main definition:

Definition 11.3 (binding bigraphs) A *(concrete) binding bigraph* $G\colon I \to J$ consists of an (underlying) pure bigraph $G^{\mathsf{u}}\colon I^{\mathsf{u}} \to J^{\mathsf{u}}$ with extra structure as follows. Declare its *binders* to be the binding ports of its nodes together with the local names of its outer face J. Then G must satisfy the following:

> SCOPE RULE: If p is a binder located at a node or root w, then every peer p' of p must be located at a place w' (a site or a node) such that $w' <_{G^{\mathsf{u}}} w$.

In the precategory ´$\textsc{Bbg}(\mathcal{K})$ of (concrete) binding bigraphs over $\mathcal{K}$, composition and identities are defined as for the underlying pure bigraphs; they are easily found to respect the scope rule. The forgetful functor

$$\mathcal{U} \colon ´\textsc{Bbg}(\mathcal{K}) \to ´\textsc{Big}(\mathcal{K})$$

forgets binding; it sends each I to I^{u} and each G to G^{u}.

The analogous definition holds also for hard binding bigraphs ´$\textsc{Bbg}_{\mathsf{h}}(\mathcal{K})$. ■

As for pure bigraphs, a link is *open* if it is a name, otherwise *closed*. In binding bigraphs we have a further distinction: a link is *bound* if it contains a binder, otherwise *free*. These terms also extend to the points in the link. The scope rule ensures that every bound link in $G \colon I \to J$ has exactly one binder and that all global inner names are free; local inner names (i.e. local names of I) can be free or bound.

Note that, in considering names, the global/local distinction is with reference to interfaces, while the free/bound distinction is with reference to bigraphs. For example, for $G \colon I \to J$, an inner name of G may be a *local* name in I but *free* in G.

We shall say that a bigraph $G \colon I \to J$ is *free* if its outer face J is global, i.e. every outer name of G is free. To avoid confusion, note that G itself may still not be a pure bigraph, i.e. it may have closed links that are bound. Free bigraphs will be important in what follows; their characteristic is that no bound point of G can become free in $F \circ G$, for any F.

We shall now embark on recording or verifying several properties of binding bigraphs, and how each relates to the corresponding property in Section 9 for pure bigraphs. Where the correspondence is easy we do not give a new definition, proposition etc; we merely cite the corresponding one for pure bigraphs.

In many cases the correspondence is illuminated by an attribute of the forgetful functor $\mathcal{U}$. Let us recall the relevant attributes. First, $\mathcal{U}$ is *faithful* but not *full*; that is, restricted to each particular homset ´$\textsc{Bbg}(I, J)$ or ´$\textsc{Bbg}_{\mathsf{h}}(I, J)$ it is injective, but not surjective.

A functor can be characterised by the properties that it *preserves* or *reflects*. Letting Δ range over arbitrary commuting diagrams, a functor $\mathcal{F}$ is said to *preserve* a property Φ of diagrams if $\Phi(\Delta) \Rightarrow \Phi(\mathcal{F}(\Delta))$, and to *reflect* Φ if $\Phi(\mathcal{F}(\Delta)) \Rightarrow \Phi(\Delta)$. An interesting example will be that $\mathcal{U}$ preserves the RPO property but does not reflect it. This means that to find an RPO in binding bigraphs we cannot take *any* $\mathcal{U}$-preimage of an RPO in pure bigraphs; but we shall find that one *particular* preimage, unique up to iso, is indeed an RPO.

For the remainder of this section we shall extend to binding bigraphs several properties of pure bigraphs, in the order in which they were treated in Section 9.

Binding bigraphs: Elementary notions

isomorphisms (Proposition 9.2) An iso $\iota \colon \langle m, \vec{k}, X \rangle \to \langle m, \vec{\ell}, Y \rangle$ of binding bigraphs combines a permutation $\pi \colon m \to m$ of places with a bijection $X \to Y$ of global names and, for each place $r \in m$, a bijection $k_r \to \ell_{\pi(r)}$ between the local names at corresponding places. $\mathcal{U}$ both preserves and reflects isos.

tensor product (Definition 9.3) The tensor product of interfaces $I = \langle m, \vec{k}, X\rangle$ and $J = \langle n, \vec{\ell}, Y\rangle$, where X and Y are disjoint, is $I \otimes J = \langle m+n, \vec{k}\vec{\ell}, X \uplus Y\rangle$. The tensor product of two binding bigraphs $G_i : I_i \to J_i$ ($i = 0, 1$), where $I_0 \otimes I_1$ and $J_0 \otimes J_1$ are defined, is just $G_0^{\mathsf{u}} \otimes G_1^{\mathsf{u}}$, but after disjoining the local inner and outer names names of G_1 from those of G_0.

$\mathcal{U}$ neither preserves nor reflects tensor product. This is essentially because of the partial nature of the tensor; a more standard categorical tensor product would be total, because it would *disjoin* name sets before taking their product. This would complicate our application of bigraphs to process calculi. However the following shows that the tensor is well-behaved for binding bigraphs:

wide monoidal (Theorem 9.4) $\acute{}\mathrm{B}\textsc{bg}(\mathcal{K})$ and $\acute{}\mathrm{B}\textsc{bg}_{\mathsf{h}}(\mathcal{K})$ are wide monoidal precategories.

epis and monos (Proposition 9.5) A binding bigraph is epi (resp. mono) iff its underlying pure bigraph is epi (resp. mono). $\mathcal{U}$ both preserves and reflects epis and monos.

This concludes the elementary notions for binding bigraphs. ■

We now move on to the central properties of relative and idem pushouts. To ensure that RPOs and IPOs exist we have to take a little more care, but can still rely mostly on the corresponding results for pure bigraphs. In what follows we shall use the terms *binding* and *pure* RPO to mean RPOs in binding and in pure bigraphs respectively; the results apply to $\acute{}\mathrm{B}\textsc{bg}_{\mathsf{h}}$ and $\acute{}\mathrm{B}\textsc{ig}_{\mathsf{h}}$ and well as to $\acute{}\mathrm{B}\textsc{bg}$ and $\acute{}\mathrm{B}\textsc{ig}$. We shall also talk of binding IPOs and pure IPOs.

Construction 11.4 (building a binding RPO) Let $\vec{D}\colon \vec{I} \to K$ be a bound for $\vec{A}\colon H \to \vec{I}$ in binding bigraphs. We wish to build in a binding RPO

$$(\vec{B}\colon \vec{I} \to I,\ B\colon I \to K)\ .$$

We first build a pure RPO $(\vec{B}', B')$ for $\vec{A}^{\mathsf{u}}$ to $\vec{D}^{\mathsf{u}}$, using the separate constructions for place graphs and link graphs (Constructions 7.7 and 8.8). Let $I' = \langle m, X'\rangle$ be the interface of the RPO, i.e. the inner face of B'.

We shall build the required $(\vec{B}, B)$ with interface $I = \langle m, \vec{k}, X\rangle$ so that for some link isomorphism $\iota\colon I' \to I^{\mathsf{u}}$ in $\acute{}\mathrm{B}\textsc{ig}$ (see the diagram)

$$\vec{B}^{\mathsf{u}} = \iota \circ \vec{B}' \text{ and } B^{\mathsf{u}} \circ \iota = B'\ .$$

In general, several such triples exist as bounds for $\vec{A}$ relative to $\vec{D}$; they vary with the chosen isomorphism ι, because for a given name $x \in X'$ we may choose $\iota(x)$ either to be a global name in X or to be a local name located at some place $r \in m$. They will not all be binding RPOs; this amounts to saying that $\mathcal{U}$ does not reflect the RPO property from pure to binding bigraphs.

BINDING BIGRAPHS

$$\begin{tikzcd}
 & & {} & & \\
 & & & & \\
I_0 \arrow[uurr, "D_0"] \arrow[rr, dashed, "B_0"] & & I \arrow[uu, dashed, "B"'] & & I_1 \arrow[uull, "D_1"'] \arrow[ll, dashed, "B_1"'] \\
 & & {} \arrow[ull, "A_0"] \arrow[urr, "A_1"'] & &
\end{tikzcd}$$

PURE BIGRAPHS

$$\begin{tikzcd}
 & & {} & & \\
 & & I^{\mathsf{u}} \arrow[u, dashed, "B^{\mathsf{u}}"'] & & \\
I_0^{\mathsf{u}} \arrow[uurr, "D_0^{\mathsf{u}}"] \arrow[urr, dashed, "B_0^{\mathsf{u}}"] \arrow[rr, "B'_0"] & & I' \arrow[u, dashed, "\iota"'] \arrow[uu, bend left, "B'"] & & I_1^{\mathsf{u}} \arrow[uull, "D_1^{\mathsf{u}}"'] \arrow[ull, dashed, "B_1^{\mathsf{u}}"'] \arrow[ll, "B'_1"'] \\
 & & {} \arrow[ull, "A_0^{\mathsf{u}}"] \arrow[urr, "A_1^{\mathsf{u}}"'] & &
\end{tikzcd}$$

In fact we shall choose I and ι so that $\vec{B}$ and B obey the scope rule, and also — subject to that constraint— so that I contains as many local names as possible. By Construction 8.8 (and using its notation), each $x \in X'$ is linked in B'_i to one or more names $x_i \in X_i^{\mathsf{u}}$ ($i = 0, 1$), where X_i^{u} is the name set of A^{u}. If any such name x_i is a global name of A_i then we choose $\iota(x)$ to be global in I^{u} — i.e. a member of X. Otherwise every such x_i is a local name of A_i, so we choose one of them, located at some site s in I_i; we then choose $\iota(x)$ to be located at the unique $r \in m$ such that $s < r$ in B'_i. This determines both $(\vec{B}, B)$ and I up to isomorphism. ■

Note that the scope rule on $\vec{A}$ ensures that the definition of r is independent of the choice of local name x_i linked to x. The construction is therefore well defined.

Proposition 11.5 (binding RPOs) *A binding RPO for $\vec{A}$ to $\vec{D}$ is provided by Construction 11.4, via Corollary 9.6.*

Proof Let $(\vec{B}, B)$ be as in the construction. Note that $(\vec{B}, B)^{\mathsf{u}}$ is a pure RPO for $\vec{A}^{\mathsf{u}}$ to $\vec{D}^{\mathsf{u}}$. Let $(\vec{C}, C)$, with interface J, be any bound for $\vec{A}$ relative to $\vec{D}$. We must find a unique mediating arrow $\widehat{C}\colon I \to J$.

$$\begin{tikzcd}
 & & {} & & \\
 & & J^{\mathsf{u}} \arrow[u, "C^{\mathsf{u}}"] & & \\
I_0^{\mathsf{u}} \arrow[uurr, "D_0^{\mathsf{u}}"] \arrow[urr, "C_0^{\mathsf{u}}"] \arrow[rr, "B_0^{\mathsf{u}}"] & & I^{\mathsf{u}} \arrow[u, dashed, "C'"'] \arrow[uu, bend left, "B^{\mathsf{u}}"] & & I_1^{\mathsf{u}} \arrow[uull, "D_1^{\mathsf{u}}"'] \arrow[ull, "C_1^{\mathsf{u}}"'] \arrow[ll, "B_1^{\mathsf{u}}"'] \\
 & & {} \arrow[ull, "A_0^{\mathsf{u}}"] \arrow[urr, "A_1^{\mathsf{u}}"'] & &
\end{tikzcd}$$

Now $(\vec{C}, C)^{\mathsf{u}}$, with interface J^{u}, is a bound for $\vec{A}^{\mathsf{u}}$ relative to $\vec{D}^{\mathsf{u}}$. Hence there is a unique mediating arrow C' as shown in the diagram. We now claim that C' obeys the scope rule. This can be shown by a case analysis for the bound points in C', and depends on three facts: first, that $\vec{B}^{\mathsf{u}}$ and $\vec{C}^{\mathsf{u}}$ obey the scope rule; second, that each name $x \in I^{\mathsf{u}}$ is linked in B_i^{u} ($i = 0$ or 1) to some name $x_i \in I_i^{\mathsf{u}}$; third, that by construction x is local iff all such x_i are local.

It follows that $C' = \widehat{C}^{\mathsf{u}}$ for some mediating arrow $\widehat{C}\colon I \to J$; also that this is unique, since $C'\colon I^{\mathsf{u}} \to J^{\mathsf{u}}$ is a unique mediating arrow and $\mathcal{U}$ is a faithful functor. This completes the proof. ■

Corollary 11.6 (preserving RPOs) *The forgetful functor $\mathcal{U}$ preserves RPOs; that is, if $(\vec{C}, C)$ is a binding RPO for $\vec{A}$ to $\vec{D}$ then $(\vec{C}, C)^{\mathsf{u}}$ is a pure RPO for $\vec{A}^{\mathsf{u}}$ to $\vec{D}^{\mathsf{u}}$.*

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 Construction 11.4. Then, since RPOs are unique up to isomorphism, there is a mediating iso $\iota\colon I \to J$ between these two RPOs. Also from the construction we know that $(\vec{B}, B)^{\mathsf{u}}$ is a pure RPO for $\vec{A}^{\mathsf{u}}$ to $\vec{D}^{\mathsf{u}}$, and we have a mediating iso $\iota^{\mathsf{u}}\colon I^{\mathsf{u}} \to J^{\mathsf{u}}$ between this RPO and the relative bound $(\vec{C}, C)^{\mathsf{u}}$. But isomorphism preserves the RPO property, so $(\vec{C}, C)^{\mathsf{u}}$ is also a pure IPO. ■

Let us now turn to binding IPOs. Their construction —unlike RPOs— depends upon a set of consistency conditions, and we find that one extra condition is needed in binding bigraphs. Then we show how to construct a family of binding IPOs for any consistent pair $\vec{A}\colon H \to \vec{I}$, based upon the corresponding construction of pure IPOs. Finally we check that this indeed yields all binding IPOs for $\vec{A}$, up to isomorphism.

Definition 11.7 (consistency conditions) Let $\vec{A}$ be a pair of binding bigraphs with common inner face. We define three conditions for $\vec{A}$ to be consistent:

CP Conditions CP0 – CP2 for the underlying place graphs (Definition 7.9);
CL Conditions CL0 – CL2 for the underlying link graphs (Definition 8.10);
CB If p is a shared point, bound and closed in A_i but open in $A_{\bar{\imath}}$, then $A_{\bar{\imath}}(p)$ is a local name. ■

To see the need for CB, suppose that $\vec{B}$ is a bound for $\vec{A}$; let $A_0(p) = e$ be bound and closed, and $A_1(p) = x$, a name. Then $(B_0 \circ A_0)(p) = e$, hence also $(B_1 \circ A_1)(p_2) = e$; thus $B_1(x) = e$, and the scope rule for B_1 requires x to be local.

As we did for place graphs and link graphs, we find that these conditions are necessary and sufficient for the existence of both bounds and IPOs in binding bigraphs. We can also characterise the binding IPOs in terms the pure ones. We deal with these in a single theorem as follows (note that the sufficiency of the consistency conditions follows from clause (2) of the theorem):

Theorem 11.8 (binding IPOs)

(1) The consistency conditions CP, CL *and* CB *are necessary for the existence of bounds in binding bigraphs.*

(2) Let $\vec{A}$ satisfy the consistency conditions and $\vec{A}^{\mathsf{u}}$ have a pure IPO $\vec{B}'$. Then $\vec{A}$ has a binding IPO $\vec{B}$, with $\vec{B}^{\mathsf{u}}$ isomorphic to $\vec{B}'$.

(3) If $\vec{A}$ has a binding IPO $\vec{B}$, then $\vec{A}^{\mathsf{u}}$ has a pure IPO $\vec{B}^{\mathsf{u}}$.

Proof (1) Suppose $\vec{B}$ bounds $\vec{A}$ in binding bigraphs. Then $\vec{B}^{\mathsf{u}}$ bounds $\vec{A}^{\mathsf{u}}$ in pure bigraphs; so immediately CP and CL hold, since they are necessary for a bound in place graphs and link graphs respectively. The condition CB can be established by an argument based on the preceding discussion.

(2) Suppose that $\vec{A}$ satisfies the consistency conditions, and $\vec{A}^{\mathsf{u}}$ has a pure IPO $\vec{B}'$. (Note that conditions CP and CL ensure at least one such IPO.) We construct a binding bigraph $\vec{B}$ and a pure isomorphism ι such that $\vec{B}^{\mathsf{u}} = \iota \circ \vec{B}'$; the construction proceeds as in Construction 11.4 but uses condition CB on $\vec{A}$. Then $\vec{B}^{\mathsf{u}}$ is also a pure IPO, hence $(\vec{B}, \mathsf{id})^{\mathsf{u}}$ is an RPO for $\vec{A}^{\mathsf{u}}$ to $\vec{B}^{\mathsf{u}}$. Therefore, using the construction and corollary in this special case, we conclude that $\vec{B}$ is a binding IPO for $\vec{A}$.

(3) Finally, the third result is a special case of Corollary 11.6. ■

Thus, when the pair $\vec{A}$ of binding bigraphs is consistent, there is a precise correspondence between its binding IPOs and the pure IPOs of $\vec{A}^{\mathsf{u}}$.

We now lift further static properties to binding bigraphs, especially IPO properties. Note especially that pure bigraphs are a sub-precategory of binding bigraphs: those without binders. Although we shall not labour the point, our theory for binding bigraphs is a conservative extension of that for pure bigraphs; that is, every property of binding bigraphs, when restricted to pure ones, coincides with the corresponding property of the latter as previously defined.

Binding bigraphs: Further properties

special IPOs (Propositions 9.8 and 9.9, Corollary 9.10 and Proposition 9.11) The containment pushout, and the tensoring of two IPOs with disjoint supports, hold unchanged in binding bigraphs. Also, the notion of a *lean* bigraph —one with no idle edges— is unchanged, because of course an idle edge cannot be bound, so properties relating IPOs to idle edges and leanness are unchanged. It follows that lean-support equivalence ($\overset{\frown}{\smile}$), which extends support equivalence ($\widehat{=}$) by discarding idle edges, also unchanged. This leads to the following:

abstract bigraphs (Definition 9.12) An abstract binding bigraph is a lean-support equivalence class of concrete ones. For any signature $\mathcal{K}$ this leads to the category $\mathrm{B}\textsc{bg}(\mathcal{K})$, and the quotient functor $[\![\cdot]\!]\colon \acute{}\mathrm{B}\textsc{ig}(\mathcal{K}) \to \mathrm{B}\textsc{ig}(\mathcal{K})$. Similarly for hard binding bigraphs we have $[\![\cdot]\!]\colon \acute{}\mathrm{B}\textsc{ig}_{\mathsf{h}}(\mathcal{K}) \to \mathrm{B}\textsc{ig}_{\mathsf{h}}(\mathcal{K})$.

ground bigraphs As before, a *ground* bigraph is one with inner face ϵ.

interfaces The general form of interface is $I = \langle m, \vec{k}, X\rangle$. If it is global we may write it as $\langle m, X\rangle$; if it has no places (so $m = 0$) we may write it as X; if it has no names we may write it as m. I is *prime* if it has width $m = 1$, i.e. $I = \langle 1, (k), X\rangle$; then we may write it as $\langle (k), X\rangle$, or as (k) if $X = \emptyset$, or as $\langle X\rangle$ if $k = 0$. Note that $\langle X\rangle$, with unit width, differs from X with zero width.

wirings As before, a *wiring* is a bigraph whose faces have zero width. Thus it has no nodes, and takes the form $\omega\colon X \to Y$. We retain our notations $/X$ and $\vec{y}/\vec{x}$ for closure and substitution of global names.

prime bigraphs A bigraph G is *prime* if its outer face is prime and its inner face has no global names; thus $G\colon \langle m, \vec{k}, \emptyset\rangle \to \langle (\ell), X\rangle$. Note particularly that it may have local inner names. An important prime is the *datum* $\ulcorner \vec{x} \urcorner\colon (k) \to \langle \vec{x}\rangle$, which links k local inner names one by one to k free names $\vec{x} = x_1, \ldots, x_k$.

abstraction Dual to the datum is a the new operation of *abstraction* on primes P. For $P\colon I \to \langle (k), x \uplus X\rangle$ we may form the abstraction $(x)\,P\colon I \to \langle (1{+}k), X\rangle$, which turns a free name of P into a local name. (This cannot violate the scope rule, because P has no global inner names.) If $\vec{x} = x_1, \ldots, x_k$ is a vector of distinct names we write $(\vec{x})\,P$ for $(x_1)\,\cdots(x_k)\,P$; then we have the dual properties

$$(\ulcorner\vec{x}\urcorner \otimes \mathsf{id}_X)\circ(\vec{x})\,P = P \;\text{ and }\; (\vec{x})\ulcorner\vec{x}\urcorner = \mathsf{id}_{(k)}\ .$$

discreteness The notion of discreteness becomes more subtle for binding bigraphs. Recall that a link is *free* if it is not bound by a *binder* — a local name or a binding port; also that a *point* is a port or an inner name. A binding bigraph is *discrete* if every free link is an (outer) name and has exactly one point. This is a conservative extension of discreteness for pure bigraphs; it imposes no constraint on bound links.

ions, atoms, molecules The definition of an ion must now allow for binding. For any non-atomic control $K\colon h \to k$ and sequence $\vec{x}$ of k distinct names we define the *free discrete ion* $K_{\vec{x}}\colon (h) \to \langle\vec{x}\rangle$ whose h local inner names are bound by a single K-node. Then for any prime discrete P with outer face $\langle (h), Y\rangle$, where Y is disjoint from the names $\vec{x}$, we call $(K_{\vec{x}} \otimes Y)\circ P$ a *free discrete molecule*. For atomic K a *free discrete atom* is just $K_{\vec{x}}\colon \epsilon \to \langle\vec{x}\rangle$ as before. An arbitrary ion, molecule or atom is got by imposing wiring and abstracting free names.

This concludes our second list of properties for binding bigraphs, including a taxonomy which is a conservative extension of the taxonomy of pure ones. ■

Finally we (conservatively) extend the important operations and decompositions from pure to binding bigraphs, and add some new ones.

Binding bigraphs: Operations and decompositions

parallel product (Definition 9.13, Proposition 9.14) Extending the previous definition, the parallel product of two interfaces $J_i = \langle n_i, \vec{k}_i, Y_i\rangle$ $(i = 0, 1)$ keeps their local names disjoint but may share their global names:

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

We define parallel product on binding bigraphs by the equation in Proposition 9.14.

prime product (Definition 9.15) Extending the previous definition, the prime product of two prime interfaces is

$$\langle\vec{k}, X\rangle \mid \langle\vec{\ell}, Y\rangle \stackrel{\text{def}}{=} \langle\vec{k}\vec{\ell}, X \cup Y\rangle\ .$$

The expression of the prime product of two prime binding bigraphs in terms of their parallel product is just as before.

underlying discrete bigraph (Proposition 9.16) The previous unique decomposition of any bigraph G in terms of its underlying discrete one is extended almost exactly:

$$G = (\omega \otimes \mathsf{id}_I) \circ D \ ,$$

where ω is a wiring, D is discrete and I has no free names. As before, this is called the *discrete normal form* (DNF) of G.

synthesis and analysis of discrete bigraphs (Proposition 9.17) Again the discrete binding bigraphs form a monoidal sub-precategory. The factorisation of a discrete $D \colon \langle m, \vec{k}, X\rangle \rightarrow \langle n, \vec{\ell}, Y\rangle$ into a tensor product of prime discrete factors is as before (where of course $\vec{k}$ and $\vec{\ell}$ were essentially zero-vectors), and the IPO properties of discrete bigraphs are as before — but replacing the wiring component $\omega \otimes \mathsf{id}_n$ by $\omega \otimes \mathsf{id}_I$ where I has no free names.

instantiation (Definition 9.18) We replace instantations $\varrho :: m \rightarrow n$ for pure bigraphs by instantations $\varrho :: I \rightarrow J$ for binding bigraphs, where the interfaces $I = \langle m, \vec{k}\rangle$ and $J = \langle n, \vec{\ell}\rangle$ have no free names. The instantiation is again determined by the underlying function $\overline{\varrho} \colon n \rightarrow m$, which must also satisfy $\ell_j = k_{\overline{\varrho}(j)}$ for all $j \in n$. For any X, this condition allows the map

$$\varrho \colon \mathsf{Gr}(X \otimes I) \rightarrow \mathsf{Gr}(X \otimes J)$$

to be defined just as before in terms of DNF, but with wiring component $\omega \otimes \mathsf{id}_n$ replaced by $\omega \otimes \mathsf{id}_I$ where I has no free names.

wiring an instance (Proposition 9.19) The proof that $\varrho(\omega a) = \omega\varrho(a)$ for all instantiations ϱ and wirings ω proceeds as before.

wiring a product (Proposition 9.20) The proofs that $\omega(F \parallel G) = \omega F \parallel \omega G$ and that $\omega(F \mid G) = \omega F \mid \omega G$ proceed as before.

instantiating a product (Proposition 9.21) The proof proceeds as before that if the a_i are prime, with total name set Y, then

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

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

instantiating with prime component (Proposition 9.22) This proposition asserts that if G has prime inner face I and $a, b \colon I$ are two prime agents, then the two instances $\varrho(G \circ a)$ and $\varrho(G \circ b)$ are similar, in the sense that one can be transformed into the other by replacing several occurrences of a by b. This property is vital for proofs of bisimilarity, and its proof proceeds as for pure bigraphs.

This concludes our extension of operations and decompositions to binding bigraphs. ■

We shall not refine the algebraic theory of Section 10 for binding bigraphs. We conjecture that the definitions and results need only minor adjustments, chiefly concerning data and abstractions, but we do not need them for this paper.

To conclude: We have established the static theory of binding bigraphs as a conservative extension of that for pure bigraphs. The refinements were mostly minor; only with RPOs and IPOs was a non-trivial extra argument needed. In later sections, where we develop the dynamics of binding bigraphs, we shall often appeal to a definition or result about pure bigraphs that has been extended or refined here to binding bigraphs; then we shall add a superscript ‘b’ to the reference, for example ‘Definition 9.18$^{\mathsf{b}}$’.

12 Reactions and transitions

In Section 11 we ensured the existence of RPOs in binding bigraphs, and defined some useful structural properties such as discreteness. We are now ready to specialise the definitions and theory for wide reactive systems (WRSs) in Section 4, to obtain bigraphical reactive systems (BRSs). We do it here for binding bigraphs. The binding BRSs form the objects of a category whose arrows are WRS functors; the pure BRSs constitute a subcategory. Readers who have omitted reading Section 11 on binding bigraphs may interpret this whole section in terms of pure bigraphs; to do this, simply read ´BIG and BIG for ´BBG and BBG, and ignore the superscript 'b' on references to definitions and results.

To define the notion of BRS, the main remaining step is to define parametric reaction rules over (binding) bigraphs, and the main result we obtain is a congruence theorem —both for ´BBG and for ´BBG$_{\mathsf{h}}$— which we are then able to transfer to abstract bigraphs in BBG and BBG$_{\mathsf{h}}$.

Let us consider reaction rules for a BRS, recalling both the abstract Definition 4.3 for WRSs and the examples in Section 2. What should the parameters $\mathsf{Par}(I)$ be, and what should the transform maps $\mathtt{trans} : \mathsf{Par}(I) \to \mathsf{Gr}(I')$ be? Parameters will be *discrete*; recall from Section 11 that a binding bigraph is discrete if every free link is open (i.e. a free name) and has exactly one point. This does not limit the applicability of the reaction rules, because from Proposition 9.16$^{\mathsf{b}}$ we can obtain everything by combining discreteness with wiring; moreover, the technical development is smoother with discrete parameters. For the transforms, we must allow for both replication and discard of parameters, so we use the *instantiations* of Definition 9.18$^{\mathsf{b}}$.

Definition 12.1 (reaction rules for bigraphs) A *ground (reaction) rule* is a pair (r, r'), where r and r' are ground with the same outer face. Given a set of ground rules, the *reaction relation* $\longrightarrow\!\!\!\triangleright$ over agents is the least such that $D \circ r \longrightarrow\!\!\!\triangleright D \circ r'$ for each active D and each ground rule (r, r').

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

$$(\, R \colon I \to J,\ R' \colon I' \to J,\ \varrho \,)$$

where the inner faces I and I' have widths m and m' but no free names.[8] The third component $\varrho :: I \to I'$ is an instantiation (Definition 9.18$^{\mathsf{b}}$). For every X and discrete $d \colon X \otimes I$ the parametric rule generates the ground reaction rule

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

Note that as $d = d_0 \otimes \cdots \otimes d_{m-1}$ is discrete (where each d_i is prime) the instance $\varrho(d)$ takes the form

$$\varrho(d) = X \parallel d_{\overline{\varrho}(0)} \parallel \cdots \parallel d_{\overline{\varrho}(m'-1)} \colon X \otimes I'\ .$$

Thus, if some d_i occurs more than once in the instance, the bound names of each copy may be treated differently by the reactum R'. On the other hand, the free names of d are exported to the context that surrounds the reaction.

[8] For pure bigraphs we shall have $I = m$ and $I' = m'$; then the instantiation is $\varrho :: m \to m'$.

It follows from this remark that requiring d to be discrete does not limit reaction, because the outer context can impose any wiring present in a non-discrete parameter. However, it does limit transitions, essentially because the parameter part L^{par} of a label L reflects the discreteness of d. Without the limitation, the parameter underlying a transition $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$ might impose an arbitrary substitution on the part of a shared with it, exceeding the role a parameter should play.

The function $\overline{\varrho}$ underlying an instantiation need not be injective, so it allows for *replication* of parameters in the reactum. Similarly it need not be surjective, so it allows for the *discard* of parameters.

Definition 12.2 (bigraphical reactive system) A *bigraphical reactive system (BRS)* over a signature $\mathcal{K}$ consists of ´$\mathrm{B{\small BG}}(\mathcal{K})$ equipped with a set ´Reacts of reaction rules closed under support equivalence ($\bumpeq$). We denote it —and similarly for ´$\mathrm{B{\small BG}}_{\mathsf{h}}(\mathcal{K})$— by

$$´\mathrm{B{\small BG}}(\mathcal{K},\ ´\mathsf{Reacts})\ . \qquad ■$$

Proposition 12.3 (a BRS is a WRS) *Every bigraphical reactive system is a wide reactive system.*

Proof First, it is easy to see that ´$\mathrm{B{\small BG}}(\mathcal{K})$ and ´$\mathrm{B{\small BG}}_{\mathsf{h}}(\mathcal{K})$ are wide precategories, for any signature $\mathcal{K}$. The support of a bigraph is the disjoint sum $V + E$ of its node set and edge set, the width of an interface $\langle m, \vec{k}, X\rangle$ is m, and we have already discussed the width of a bigraph. The other details are easy to check.

For the reaction rules, their parameters $\mathsf{Par}(I)$ at any interface are just the discrete agents $d\colon I$. The activity map act is given by

$$\mathsf{act}(C) \stackrel{\text{def}}{=} \{s \in m \mid \forall v.\ v >_C s \Rightarrow v \text{ active}\}\ .$$

Lastly the transform of each reaction rule is provided by its instantiation ϱ. ■

This result ensures that BRSs inherit from WRSs the definition of transitions $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$ based upon IPO pairs, and the standard transition system ST induced from its reaction rules. They also inherit the definition of bisimilarity, and so we have the following immediate corollary of Theorem 5.5:

Corollary 12.4 (congruence of wide bisimilarity) *In any concrete BRS equipped with the standard transition system* ST, *wide bisimilarity of agents is a congruence.*

We would now like to transfer ST, together with its congruence property, to the abstract BRS $\mathrm{B{\small BG}}(\mathcal{K},\mathsf{Reacts})$, where $\mathrm{B{\small BG}}(\mathcal{K})$ is defined by the quotient functor $[\![\cdot]\!]$ of Definition 9.12$^{\mathsf{b}}$, and Reacts is obtained from ´Reacts also by $[\![\cdot]\!]$. (Similarly for $\mathrm{B{\small BG}}_{\mathsf{h}}$.)

Now recall that this functor, the quotient by lean-support equivalence ($\Bumpeq$), is a little coarser than the quotient by support equivalence ($\bumpeq$), because it discards idle edges. To transfer the congruence result we must prove that $\Bumpeq$ respects ST. For this purpose, we have to impose a slight constraint upon the reaction rules ´Reacts, namely that every redex is *lean* — i.e., recalling Section 9, it has no idle edges. We then deduce the crucial property of lean-support equivalence:

Proposition 12.5 (transitions respect equivalence) *In a concrete BRS with all redexes lean, equipped with* ST*:*

(1) In every transition label L, both components are lean.

(2) Transitions respect lean-support equivalence ($\Bumpeq$) in the sense of Definition 5.2. That is, for every transition $a \overset{L}{\longrightarrow\!\!\triangleright}_{\lambda} a'$, if $a \Bumpeq b$ and $L \Bumpeq M$ where M is another label with $M \circ b$ defined, then there exists a transition $b \overset{M}{\longrightarrow\!\!\triangleright}_{\lambda} b'$ for some b' such that $a' \Bumpeq b'$.

Proof For the first part, use Proposition $9.11^{\mathsf{b}}(1)$ and the fact that every discrete agent is lean. For the second part, use Proposition $9.11^{\mathsf{b}}(2)$; the assumption that each redex is lean ensures that it cannot share an idle edge with the agent a. ■

We are now ready to transfer the congruence result of Corollary 12.4 from concrete to abstract BRSs. The following is immediate by invoking Theorem 5.7:

Corollary 12.6 (behavioural congruence in abstract BRSs) *Let $\acute{\mathbf{A}}$ be a concrete BRS with all redexes lean, equipped with* ST*. Let $[\![\cdot]\!] : \acute{\mathbf{A}} \to \mathbf{A}$ be the quotient functor by lean-support equivalence. Then*

(1) $a \sim b$ in $\acute{\mathbf{A}}$ iff $[\![a]\!] \sim [\![b]\!]$ in $\mathbf{A}$.

(2) Bisimilarity is a congruence in $\mathbf{A}$.

This concludes the elementary theory of bigraphical reactive systems. In Part III we shall refine these results for two important classes of BRS, thereby obtaining more tractable transition systems. Part III ends by applying the results to a bigraphical presentation of an asynchronous π-calculus, exactly recovering its standard behavioural theory.

Part III
Specialisation and Application

The class of *simple* BRSs is introduced; they include models of both the π-calculus and the mobile ambient calculus. Their structural properties allow us to simplify the transition systems that were derived more generally in Part II. In particular, we prove an important *adequacy* theorem; it asserts that in the derived transition system for a simple BRS it is enough to confine attention to those transitions $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$ in which the agent a contributes non-trivially to the underlying reaction.

We then narrow the simple BRSs still further, to the *basic* BRSs. The purpose is to obtain a nice characterisation of the labels involved in the derived transition systems; this result is verified using the techniques of relative pushouts.

We proceed to encode a finite asynchronous π-calculus as a basic BRS. The first result —independently of dynamics— is that two processes are structurally congruent if and only if their representing bigraphs coincide. Then it turns out that the labels of the derived transition system correspond well with the standard labels. Finally, we prove that the bisimilarity induced by the bigraph representation of this calculus coincides with two standard congruences, strong bisimilarity and strong barbed bisimilarity. This supports the claim that bigraphical systems are consistent with previous work in process calculi.

The final section explores several lines for further research, including suggestions both for varying the technical presentation of bigraphs and for enriching their domain of application.

13 Simple BRSs and adequacy

We shall now specialise our theory by defining the *simple* BRSs, whose redexes have certain structural properties. As predicted in Section 5, working in $\acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}$ we are then able to show that engaged transitions on free prime agents are adequate for the standard transition system ST. This yields a tractable transition system, which we can then transfer to abstract BRSs over $\mathrm{BBG}_{\mathsf{h}}$, yielding a bisimilarity that is a congruence. The class of simple BRSs admits both the π-calculus and the ambient calculus.

Recall from Section 8 that a link is *open* if it belongs to the outer face, otherwise *closed*, and that these properties are inherited by the points of the link. Recall also from Section 11 that a bigraph is *free* if every open link is free.

Definition 13.1 (simple BRSs) Call a bigraph *open* if every free link is open. Call it *guarding* if no inner name is open, and no site has a root as parent. Call it *simple* if

- it has no idle names and no barren regions;
- no two inner names are peers, and no two sites are siblings;
- it is free, prime, open and guarding.

A BRS is *simple* if all its redexes are simple. ■

The first two conditions are easy to accept; indeed, we see no purpose for a redex which fails them. In a concrete BRS they equate respectively to the epi and mono properties. However, we define simpleness as above because we would like it to be preserved by the quotient functor $[\![\cdot]\!]$ (which preserves neither epis nor monos).

Again, guarding is an easy condition to accept. So the main simpleness constraints are freeness, primeness and openness. It remains to be seen how far we can relax these three constraints without weakening our results.

We give without proof three easy properties of openness:

Proposition 13.2 (openness properties)

(1) A composition $F \circ G$ is open iff both F and G are open.

(2) Every open bigraph is also lean (i.e. has no idle edges).

(3) If $\vec{B}$ is an IPO for $\vec{A}$ and A_1 is open, then B_0 is open.

For the rest of this section we are concerned only with hard BRSs, i.e. over $\acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K})$ or $\mathrm{BBG}_{\mathsf{h}}(\mathcal{K})$ for some signature $\mathcal{K}$. To see where simpleness is used in our adequacy proof we shall underline each use of a simpleness condition. The first consequence of simpleness in a hard concrete BRS is that, if we limit consideration to *free* agents, we can often avoid the elisions that arise in IPOs and RPOs:

Proposition 13.3 (transition pushouts) *In a hard concrete BRS, if the IPO pair underlying a standard transition of a free agent has a simple redex then its main rectangle and right-hand square are pushouts.*[9]

[9] It is easy to show that the left-hand square need not be a pushout; in fact it may be an elisive IPO. This arises because *d*, though discrete, may have idle *bound* names. This cannot happen in *pure* bigraphs, so in that case the left-hand square is indeed a pushout.

Proof The typical IPO pair underlying a transition $a \overset{L}{\longrightarrow\!\!\triangleright}_{\lambda} a'$ is shown in the diagram below. Let $r = (\mathsf{id}_W \otimes R) \circ d$ be the ground redex. It will be enough to show that in the two IPOs, (L, D) for (a, r) and (L^{red}, D) for $(D^{\mathsf{par}}, \mathsf{id}_W \otimes R)$, there can be no elisions. There can be no place elisions because we are working with hard place graphs, so we need only consider link elisions.

In the first case, since R is open and every parameter d (being discrete) is open, then so are r and L by Proposition 13.2. So any elision of a name x of a would be to a bound closed link in L, violating the scope rule for L because x is free. On the other hand no names of r can be elided, since it has no idle names because R —and hence r— is epi.

The argument in the second case is similar except that, unlike a, D^{par} may have bound names. But since a is free, and the left square is an IPO, any outer name in D^{par} can only be bound if it is linked by D^{par} to a bound name of d; therefore it is a busy name and cannot be elided. The rest of the argument for the second case is as for the first case, so we are done. ■

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "a"] \arrow[r, "d"'] & {} \arrow[u, "D^{\mathsf{par}}"'] \arrow[r, "\mathsf{id}_W \otimes R"'] & {} \arrow[u, "D"']
\end{tikzcd}$$

We shall also need a specific property of transitions with simple redexes:

Lemma 13.4 *In a hard concrete BRS, let the IPO pair underlying a standard transition (as shown) have a simple redex R. Suppose that $|D^{\mathsf{par}}| \cap |R| = \emptyset$. Then $D^{\mathsf{par}} = D' \otimes \mathsf{id}_I$ for some D', up to isomorphism, where I is the inner face of R.*

Proof We first prove that, for each name y (necessarily local) in I, there is a local outer name z of D^{par} such that $D^{\mathsf{par}}(y) = z$ and y has no peers in D^{par}.

Since R is guarding and open, $R(y) = p$ for some binding port p, so by commutativity $(L^{\mathsf{red}} \circ D^{\mathsf{par}})(y) = p$. But p is not a port in D^{par} by assumption, so for some outer name z we have $D^{\mathsf{par}}(y) = z$ and $L^{\mathsf{red}}(z) = p$. Also p is binding, so z is local by the scope rule for L^{par}.

Now suppose y has a peer, i.e. $D^{\mathsf{par}}(q) = z$ for some point $q \neq y$. Then we have $(L^{\mathsf{red}} \circ D^{\mathsf{par}})(q) = p$, whence also $R(q) = p$. If q is an inner name this contradicts R mono; if q is a node port then q is also in R, contradicting $|D^{\mathsf{par}}| \cap |R| = \emptyset$. Hence no such q can exist.

The result follows by a similar argument showing that each site s of I has a root as parent and has no siblings. ■

We now turn to engaged transitions, especially those involving free prime agents.

Definition 13.5 (engaged transitions) A standard transition of a is said to be *engaged* if it can be based on a reaction with redex R such that $|a| \cap |R| \neq \emptyset$.

We denote by FPE the transition system of free prime interfaces and engaged transitions. We write $\sim^{\textsc{fpe}}$ for $\sim^{\textsc{fpe}}_{\textsc{st}}$, bisimilarity for FPE relative to ST. ■

Now we would like to prove that $\sim^{\text{FPE}}$ is adequate for standard bisimilarity (Definition 5.8), i.e. $\sim^{\text{FPE}} = \sim$ restricted to free prime interfaces; for then, when a and b are free prime agents, to establish $a \sim b$ we need only prove $a \sim^{\text{FPE}} b$. For this purpose, we need only match each *engaged* transition of a (resp. b) by an arbitrary transition of b (resp. a). This is a lighter task than matching *all* transitions.

In proving that $a \sim^{\text{FPE}} b$ implies $a \sim b$ for free prime a and b, we have to show how b can match the *non-engaged* transitions of a, and the antecedent only tells us how to match the *engaged* ones. However, it turns out that a non-engaged transition of a can be suitably matched by *any* b (whether or not $a \sim^{\text{FPE}} b$). This is intuitively not surprising, because a contributes nothing to such a transition, so replacing it by b should not prevent the transition occurring.

We begin with a lemma that justifies this intuition, even in the case that a may contribute to the *parameter* of the reaction.

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "a"] \arrow[r, "d"'] & {} \arrow[u, "D^{\mathsf{par}}"'] \arrow[r, "\mathsf{id}_W \otimes R"'] & {} \arrow[u, "D"']
\end{tikzcd}$$

Lemma 13.6 *In a hard concrete BRS let a be free and prime, with a standard transition $a \xrightarrow{L}_{\lambda} a'$ based upon the reaction rule (R, R', ϱ), with underlying IPO pair as shown in the above diagram.*

Let R be simple, and assume that $|a| \cap |R| = \emptyset$ but that $|a| \cap |d| \neq \emptyset$. Then $|a| \subseteq |d|$, and moreover L^{red} and a' can be expressed to within isomorphism in the form

$$L^{\mathsf{red}} = \mathsf{id}_{W'} \otimes R \quad \textit{and} \quad a' = (\mathsf{id}_{W'} \otimes R') \circ \varrho(L^{\mathsf{par}} \circ a) \ .$$

Proof From Lemma 13.4 we find that D^{par} takes the form $D^{\mathsf{par}} = D' \otimes \mathsf{id}_I$ up to isomorphism, where D' has domain W (with zero width).

We now claim that D' has no nodes. For there exists a node $u \in |a| \cap |d|$; if there exists any $v \in |D'|$ then also $v \in |a|$, hence (because a is prime) we would have u, v in the same region of $L^{\mathsf{par}} \circ a$ but different regions of $D^{\mathsf{par}} \circ d$, contradicting $L^{\mathsf{par}} \circ a = D^{\mathsf{par}} \circ d$. Thus $|a| \subseteq |d|$, and $D^{\mathsf{par}} = \omega \otimes \mathsf{id}_I$, with $\omega : W \to W'$ a wiring.

By Proposition 13.3 the right-hand square in the diagram is a pushout, and hence a tensor IPO by Corollary 9.10$^{\mathsf{b}}$, so up to isomorphism we have

$$L^{\mathsf{red}} = \mathsf{id}_{W'} \otimes R \ \text{ and } \ D = \omega \otimes \mathsf{id}_J \ .$$

Thus we have the required equation for L^{red}, and for the other equation we calculate

$$\begin{array}{rcl}
a' & = & D \circ (\mathsf{id}_W \otimes R') \circ \varrho(d) \\
 & = & (\omega \otimes \mathsf{id}_J) \circ (\mathsf{id}_W \otimes R') \circ \varrho(d) \\
 & = & (\omega \otimes R') \circ \varrho(d) \\
 & = & (\mathsf{id}_{W'} \otimes R') \circ (\omega \otimes \mathsf{id}_{I'}) \circ \varrho(d) \\
(*) & = & (\mathsf{id}_{W'} \otimes R') \circ \varrho((\omega \otimes \mathsf{id}_I) \circ d) \\
 & = & (\mathsf{id}_{W'} \otimes R') \circ \varrho(L^{\mathsf{par}} \circ a)
\end{array}$$

where at $(*)$ we commute an instantiation with a wiring, by Proposition 9.19$^{\mathsf{b}}$. ■

We can now prove the adequacy theorem.

Theorem 13.7 (adequacy of engaged transitions) *In a hard concrete BRS that is simple and equipped with* ST*, the free prime engaged transitions are adequate; that is, engaged bisimilarity $\sim^{\text{FPE}}$ coincides with bisimilarity $\sim$ on free prime agents.*

Proof It is immediate that $\sim\ \subseteq\ \sim^{\text{FPE}}$ restricted to free primes. For the converse we must prove that if a_0 and a_1 are free and prime then $a_0 \sim^{\text{FPE}} a_1$ implies $a_0 \sim a_1$. For this purpose, we shall show that

$$\mathcal{S} = \{(C \circ a_0, C \circ a_1) \mid a_0 \sim^{\text{FPE}} a_1,\ \text{both free and prime}\}$$

is a standard bisimulation up to transitive closure. This will suffice, for by taking $C = \mathsf{id}$ we deduce that $\sim^{\text{FPE}} \subseteq \sim$.

Suppose that $a_0 \sim^{\text{FPE}} a_1$. Let $C \circ a_0 \xrightarrow{M}\!\!\triangleright_\mu b'_0$ be any standard transition, with $M \circ C \circ a_1$ is defined. We must find b'_1 such that $C \circ a_1 \xrightarrow{M}\!\!\triangleright_\mu b'_1$ and $(b'_0, b'_1) \in \mathcal{S}$.

There exist a reaction rule (R_0, R'_0, ϱ_0) and an underlying IPO pair as in diagram (a) below, with E_0 active; moreover if $\mathsf{width}(J_0) = m_0$ then $\mathsf{width}(E_0)(m_0) = \mu$ and $b'_0 = E_0 \circ R'_0 \circ \varrho_0(d_0)$. Then by taking RPOs we can complete diagram (b) so that every square is an IPO.

(a)

$$\begin{tikzcd}
{} \arrow[r, "M^{\mathsf{par}}"] & {} \arrow[r, "M^{\mathsf{red}}"] & {} \\
{} \arrow[u, "C\circ a_0"] \arrow[r, "d_0"'] & {} \arrow[u] \arrow[r, "\mathsf{id}_{W_0}\otimes R_0"'] & J_0 \arrow[u, "E_0"']
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
{} \arrow[r, "M^{\mathsf{par}}"] & {} \arrow[r, "M^{\mathsf{red}}"] & {} \\
{} \arrow[u, "C"] \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[u] \arrow[r, "L^{\mathsf{red}}"] & {} \arrow[u, "E"'] \\
{} \arrow[u, "a_0"] \arrow[r, "d_0"'] & {} \arrow[u] \arrow[r, "\mathsf{id}_{W_0}\otimes R_0"'] & J_0 \arrow[u, "D_0"'] \arrow[uu, bend right, "E_0"']
\end{tikzcd}$$

(c)

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "a_1"] \arrow[r, "d_1"'] & {} \arrow[u] \arrow[r, "\mathsf{id}_{W_1}\otimes R_1"'] & J_1 \arrow[u, "D_1"']
\end{tikzcd}$$

Now D_0 is active at m_0, so $a_0 \xrightarrow{L}\!\!\triangleright_\lambda a'_0$ where

$$\lambda = \mathsf{width}(D_0)(m_0) \text{ and } a'_0 = D_0 \circ R'_0 \circ \varrho_0(d_0)\ .$$

Also E is active at λ. Moreover, by Proposition 13.3, the lower rectangle and lower right-hand square in diagram (b) are pushouts. Also $b'_0 = E \circ a'_0$.

Since $M \circ C \circ a_1$ is defined we deduce that $L \circ a_1$ is defined, and we proceed to show in three separate cases the existence of a transition $a_1 \xrightarrow{L}\!\!\triangleright_\lambda a'_1$, with underlying IPO pair as shown in diagram (c). (Note that we cannot immediately infer this from $a_0 \sim^{\text{FPE}} a_1$, since the transition of a_0 may not lie in FPE.) Substituting this diagram for the lower squares in (b), we can infer a transition $C \circ a_1 \xrightarrow{M}\!\!\triangleright_\mu b'_1$ where $b'_1 = E \circ a'_1$. In each of the three cases we then argue that $(b'_0, b'_1) \in \mathcal{S}^*$, thus completing the proof of the theorem.

Case 1 Suppose the transition $a_0 \xrightarrow{L}_{\lambda} a'_0$ is not engaged, i.e. $|a_0| \cap |R_0| = \emptyset$. Suppose also that $|a_0| \cap |d_0| = \emptyset$. Then the lower large rectangle of (b), being a pushout, is tensorial; so up to isomorphism we have

$$L = id_H \otimes ((\mathsf{id}_{W_0} \otimes R_0) \circ d_0) \ \text{ and } \ D_0 = a_0 \otimes \mathsf{id}_{W_0 \otimes J_0} \ .$$

Then we calculate

$$\begin{aligned} a'_0 &= D_0 \circ (\mathsf{id}_{W_0} \otimes R'_0) \circ \varrho_0(d_0) \\ &= a_0 \otimes ((\mathsf{id}_{W_0} \otimes R'_0) \circ \varrho_0(d_0)) \\ &= E' \circ a_0 \text{ where } E' = \mathsf{id}_H \otimes ((\mathsf{id}_{W_0} \otimes R'_0) \circ \varrho_0(d_0)) \ . \end{aligned}$$

Therefore in this case we form diagram (c) by taking $d_1 = d_0$, $D_1 = D_0$ and $R_1 = R_0$; this forms a pair of IPOs and so

$$a_1 \xrightarrow{L}_{\lambda} a'_1 \overset{\text{def}}{=} E' \circ a_1 \ .$$

Then for the context $C' \overset{\text{def}}{=} E \circ E'$ we have $b'_0 = C' \circ a_0$ and $b'_1 = C' \circ a_1$; but $a_0 \sim^{\text{FPE}} a_1$, so we have $(b'_0, b'_1) \in \mathcal{S}$ as required.

Case 2 Suppose the transition $a_0 \xrightarrow{L}_{\lambda} a'_0$ is not engaged, i.e. $|a_0| \cap |R_0| = \emptyset$, but that $|a_0| \cap |d_0| \neq \emptyset$. Then since a_0 is prime, from Lemma 13.6 we find that, up to isomorphism, $L^{\mathsf{red}} = (\mathsf{id}_{W'} \otimes R_0)$ and

$$a'_0 = (\mathsf{id}_{W'} \otimes R'_0) \circ \varrho_0(L^{\mathsf{par}} \circ a_0) \ .$$

We shall now find a similar transition for a_1. We first consider $L^{\mathsf{par}} \circ a_1$. Since d_0 is discrete we know by Proposition 9.17$^{\mathsf{b}}$(2) that L^{par} is discrete; by Proposition 9.16$^{\mathsf{b}}$ we can find a wiring $\omega_1 \colon W_1 \to W'$ and discrete $d_1 \colon W_1 \otimes I_0$ such that $L^{\mathsf{par}} \circ a_1 = (\omega_1 \otimes \mathsf{id}_{I_0}) \circ d_1$, and moreover by Proposition 9.17$^{\mathsf{b}}$(3) this represents a pushout. So, by adjoining a tensorial pushout, we have an IPO pair as shown:

$$\begin{tikzcd} {} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}} = \mathsf{id}_{W'} \otimes R_0"] & {} \\ {} \arrow[u, "a_1"] \arrow[r, "d_1"'] & {} \arrow[u, "\omega_1 \otimes \mathsf{id}_{I_0}"'] \arrow[r, "\mathsf{id}_{W_1} \otimes R_0"'] & {} \arrow[u, "\omega_1 \otimes \mathsf{id}_{J_0}"'] \end{tikzcd}$$

Therefore by manipulations as in Lemma 13.6 we have

$$\begin{aligned} a_1 \xrightarrow{L}_{\lambda} a'_1 &\overset{\text{def}}{=} (\omega_1 \otimes \mathsf{id}_J) \circ (\mathsf{id}_{W_1} \otimes R'_0) \circ \varrho_0(d_1) \\ &= (\mathsf{id}_{W'} \otimes R'_0) \circ \varrho_0(L^{\mathsf{par}} \circ a_1) \ . \end{aligned}$$

Comparing this with the similar form of a'_0, and since $a_0 \sim^{\text{FPE}} a_1$ (both free and prime), we appeal to Proposition 9.22$^{\mathsf{b}}$ to find a sequence $c_0, \ldots c_k$ such that $b'_0 = c_0$, $c_k = b'_1$ and $(c_{i-1}, c_i) \in \mathcal{S}$ for $0 < i \leq k$, and thus $(b'_0, b'_1) \in \mathcal{S}^*$ as required.

Case 3 Suppose the transition $a_0 \xrightarrow{L}\!\!\triangleright_{\lambda} a'_0$ is engaged, i.e. $|a_0| \cap |R_0| \neq \emptyset$. Then since R_0 is free and prime, by considering the IPO (L, D_0) and the outer face of D_0 we find that a'_0 is free and prime, so the transition lies in FPE. But $a_0 \sim^{\text{FPE}} a_1$, so there is a transition $a_1 \xrightarrow{L}\!\!\triangleright_{\lambda} a'_1$ for some free prime a'_1 such that $a'_0 \sim^{\text{FPE}} a'_1$; hence $C \circ a_1 \xrightarrow{M}\!\!\triangleright_{\mu} b'_1 \overset{\text{def}}{=} E \circ a'_1$, and thus $(b'_0, b'_1) \in \mathcal{S}$ as required. ■

As we have seen in case 3 of the proof, when a simple transition $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$ is engaged, and a is free and prime, then so is a'. Thus, in proving the bisimilarity of prime agents, we can indeed confine attention to bisimulations containing only free prime agents.

Simpleness and adequacy makes it easy to verify two desirable properties of idle names (though they also hold more generally):

Proposition 13.8 (idle names and bisimilarity) *In a hard concrete BRS that is simple and equipped with* ST*,*

(1) $a \sim b$ *iff* $x \otimes a \sim x \otimes b$*.*

(2) $a \sim b$ *does not imply that* a *and* b *have the same idle names.*

Proof (1) For the forward implication, use congruence. For the converse, we shall verify that $\mathcal{S} = \{(a, b) \mid x \otimes a \sim x \otimes b\}$ is a bisimulation.

Let $a\mathcal{S}b$, and consider a transition consider a transition $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$. We easily deduce that $x \otimes a \xrightarrow{\mathsf{id}_x \otimes L}\!\!\triangleright_{\lambda} x \otimes a'$, hence $x \otimes b \xrightarrow{\mathsf{id}_x \otimes L}\!\!\triangleright_{\lambda} b''$ where $x \otimes a' \sim b''$. Assuming simpleness we see (as in the above proposition) that this transition of $x \otimes b$ cannot involve an elision of x. It is then easy to verify that b'' takes the form $x \otimes b'$ (up to isomorphism), where $b \xrightarrow{L}\!\!\triangleright_{\lambda} b'$. But then $a'\mathcal{S}b'$ and we are done.

(2) Consider the asynchronous π-calculus with the rule of Example 1. The agent $/x\,\mathsf{send}\,xy$, consisting of a message whose channel x has been closed, has a single name y that is not idle. On the other hand $y \otimes /x/y\,\mathsf{send}\,xy$ has an idle name y. But neither agent has an engaged transition, so they are bisimilar. ■

We now wish to transfer FPE to abstract BRSs, via the functor

$$[\![\cdot]\!] \colon \acute{}\textsc{Bbg}_{\mathsf{h}}(\mathcal{K}) \to \textsc{Bbg}_{\mathsf{h}}(\mathcal{K}) \,.$$

To do this, we would like to know that FPE is *definite* for ST (see Definition 5.10), for then by Proposition 5.11 we can equate the relative bisimilarity $\sim^{\text{FPE}}$ with the absolute one $\sim_{\text{FPE}}$. For this, we need to know that, from the pair (L, λ) alone, we can determine whether or not a transition $a \xrightarrow{L}\!\!\triangleright_{\lambda} a'$ is engaged.

It turns out that this holds in a wide range of BRSs, including the natural encoding of π-calculus and ambient calculus. This is because they all satisfy a simple structural condition, which we now define.

Definition 13.9 (definite BRS) Define $\mathit{ctrl}(G)$, the *control* of a bigraph G, to be the multiset of controls of its nodes. A BRS is *definite* if, whenever R_0 and R_1 are redexes of different rules, neither $\mathit{ctrl}(R_0)$ nor $\mathit{ctrl}(R_1)$ is a sub-multiset of the other. ■

Note that this property applies equally to concrete and abstract BRSs, and is indeed preserved and reflected by the quotient functor $[\![\cdot]\!]$. We have chosen the term ‘definite’ because, in a concrete BRS, it ensures definiteness of the engaged transitions in relation to $\textsc{st}$, in the sense of Definition 5.10. In fact, with the help of Corollary 5.12, we deduce

Corollary 13.10 (engaged congruence) *In a hard concrete BRS that is both definite and simple:*

(1) The engaged transition system $\textsc{fpe}$ *is definite for* $\textsc{st}$.

(2) Engaged bisimilarity $\sim_{\textsc{fpe}}$ *coincides with standard bisimilarity on prime agents.*

(3) $\sim_{\textsc{fpe}}$ *is a congruence; that is, for any context* C *with free prime interfaces,*

$$a \sim_{\textsc{fpe}} b \textit{ implies } C \circ a \sim_{\textsc{fpe}} C \circ b\ .$$

Now recall from Proposition 13.2 that every simple bigraph is lean. We therefore derive the analogue of Corollary 12.6, with $\textsc{fpe}$ in place of $\textsc{st}$, under extra assumptions:

Corollary 13.11 (engaged congruence in hard abstract BRSs) *Let* $\acute{\mathbf{A}} = \acute{\textsc{B}}\textsc{bg}_{\mathsf{h}}(\mathcal{K})$ *be a hard concrete BRS that is definite and simple. Let* $[\![\cdot]\!] : \acute{\mathbf{A}} \to \mathbf{A}$ *be the quotient functor for lean-support equivalence (≎). Let* $\sim_{\textsc{fpe}}$ *denote bisimilarity both for* $\textsc{fpe}$ *in* $\acute{\mathbf{A}}$ *and for the induced transition system* $[\![\textsc{fpe}]\!]$ *in* $\mathbf{A}$. *Then*

(1) $a \sim_{\textsc{fpe}} b$ *in* $\acute{\mathbf{A}}$ *iff* $[\![a]\!] \sim_{\textsc{fpe}} [\![b]\!]$ *in* $\mathbf{A}$.

(2) Engaged bisimilarity $\sim_{\textsc{fpe}}$ *is a congruence in* $\mathbf{A}$.

Proof First note that the quotient functor satisfies the conditions of Theorem 5.7. In particular, by Proposition 12.5 it respects $\textsc{fpe}$, since this is a sub-TS of $\textsc{st}$. So the theorem yields (1) immediately. It also yields (2) with the help of Corollary 13.10. ■

Thus we have ensured congruence of engaged bisimilarity in any hard abstract BRS $\textsc{Bbg}_{\mathsf{h}}(\mathcal{K})$ satisfying reasonable assumptions.

To conclude this section, we ask: What is the effect of working in $\acute{\textsc{B}}\textsc{bg}_{\mathsf{h}}$ and $\textsc{Bbg}_{\mathsf{h}}$ as we have done, rather than in $\acute{\textsc{B}}\textsc{bg}$ and $\textsc{Bbg}$? First, the theory is smoother because we avoid place graph elisions. On the other hand, the disadvantage is that the empty agent 1 is missing. In particular, the empty process $\textsc{nil}$ of the π-calculus must be encoded not by 1, but by a $\triangle$-atom, where $\triangle$ is an atomic control with zero arity (see Section 7). This inelegance is minor, because one can prove that $\triangle \,|\, a \sim a$ (provided that no redex contains $\triangle$). In Section 14 we shall improve on this; we shall find that for a subclass of the simple BRSs we can remove the inelegance by the forgetful functor from $\acute{\textsc{B}}\textsc{bg}_{\mathsf{h}}$ to $\acute{\textsc{B}}\textsc{bg}$ that replaces $\triangle$ by 1, thus turning it into a unit for prime parallel product.

14 Characterising basic BRSs

In this section we prepare for a wide range of applications of our theory to process calculi. We begin by defining the class of *basic* BRSs; for these we obtain a tractable characterisation of transitions, showing that both parts of a label-pair take a particularly simple form. We then show that in basic BRSs we can transfer the congruence for the engaged TS from hard concrete BRSs to soft abstract ones, which is where wish to deploy the theory in practice.

In Section 15 we shall deploy these results for a finite asynchronous π-calculus. Basic BRSs extend further than this, but they need refinement before handling full π-calculus or the ambient calculus. We are confident that this refinement is possible, but here we prefer to work in the simplest setting that allows an application to be treated fully.

Let us embark on defining basic BRSs. Most of the work has already been done. Recall Definition 13.1 of a *simple* BRS; every simple redex is free, prime, open and guarded, and satisfies structural conditions which, for a concrete BRS, ensure it is both epi and mono. A basic BRS will have two extra conditions.

Definition 14.1 (basic BRS) A redex is *flat* if no node has a node as parent. A redex is *basic* if it is flat and simple. A binding BRS is *basic* if it is definite (Definition 13.9) and all its redexes are basic. ■

The following proposition shows that basic redexes are easy to describe:

Proposition 14.2 (products of atoms and ions) *A redex is basic iff it is a non-empty prime product of free atoms and ions.*

We now seek to characterise uniformly, as exactly as possible, the FPE transitions in any basic BRS $\acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K},\mathsf{Reacts})$. For each *particular* such BRS this task may be relatively simple, but a general characterisation will avoid repeated work in particular cases. Throughout this section and the next we only consider FPE transitions; we write them as $a \xrightarrow{L} a'$ omitting the location index λ, because this is always zero.

The crucial property of a basic BRS that we shall exploit in order to characterise a transition $a \xrightarrow{L} a'$ is that the underlying redex r can be expressed as the parallel product of, essentially, the nodes shared with a, on the one hand, and the nodes disjoint from a, on the other. We call this property *pseudo-flatness*.

Definition 14.3 (pseudo-flat transition) A transition of an agent a is *pseudo-flat* if a and the underlying redex r can be expressed in the forms

$$a = /Z\,(\sigma\, r_0 \,|\, b) \quad \text{and} \quad r = \tau\, r_0 \,|\, r_1\ ,$$

where $|a| \cap |r_1| = \emptyset$. ■

Proposition 14.4 (basic ensures pseudo-flatness) *Every* FPE *transition arising from a basic redex is pseudo-flat.*

Thus, a BRS being basic is sufficient (though not necessary) for an FPE transition to be pseudo-flat; consequently we shall be content to characterise pseudo-flat FPE transitions.

First we need a couple of lemmas on IPOs of link graphs.

Lemma 14.5 *Let $\omega : X \to Y$ be a wiring and let $A : \epsilon \to X_0$ be a concrete link graph such that $X_0 \subseteq X$. Then the following square is pushout:*

$$\begin{tikzcd}
X \arrow[r, "\omega"] & Y \\
X \arrow[u, "A \mid \mathsf{id}_X"] \arrow[r, "\omega"'] & Y \arrow[u, "\omega\, A \mid \mathsf{id}_Y"'] .
\end{tikzcd}$$

Lemma 14.6 *Let $A_i : \epsilon \to X_i$ $(i = 0, 1)$ be concrete link graphs with disjoint support, and let $X \supseteq X_i$. Then the following square is an IPO:*

$$\begin{tikzcd}
X \arrow[r, "A_1 \mid \mathsf{id}_X"] & X \\
X \arrow[u, "A_0 \mid \mathsf{id}_X"] \arrow[r, "A_1 \mid \mathsf{id}_X"'] & X \arrow[u, "A_0 \mid \mathsf{id}_X"'] .
\end{tikzcd}$$

Recall that we use abbreviations like $\langle X \otimes Y \rangle$ for the prime interface $\langle 1, X \otimes Y \rangle$. We shall find it convenient in the following to extend this notation to wirings, so that for $\omega : X \to Y$ we denote by $\langle \omega \rangle$ the bigraph $\mathsf{id}_1 \otimes \omega : \langle X \rangle \to \langle Y \rangle$. Note that the placing of the angle brackets is somewhat arbitrary; for example $\langle X \otimes Y \rangle = \langle X \rangle \otimes Y$ and $\langle \mathsf{id}_{X \otimes Y} \rangle = \mathsf{id}_{\langle X \otimes Y \rangle} = \mathsf{id}_{\langle X \rangle} \otimes \mathsf{id}_Y$.

We now come to the characterisation theorem. Its proof relies on the notion of pseudo-flatness, which allows us to factor the redex r underlying a transition $a \xrightarrow{L} a'$ into the part shared with a, on the one hand, and the part disjoint from a, on the other. In general this factorisation is different from the decomposition of r into a parametric redex R and a parameter d, which in turn leads to the decomposition of the label L into a pair $(L^{\mathsf{red}}, L^{\mathsf{par}})$. It is convenient, therefore, to first prove the result for unstructured labels. Afterwards we shall then refine the characterisation for L to obtain L^{red} and L^{par} separately.

Theorem 14.7 (Characterising transitions in a basic BRS) *Let $a : \langle X \rangle$, and let $a \xrightarrow{L} a'$ be an FPE transition with underlying ground rule $(r, r' : \langle Y \rangle)$. Suppose the transition is pseudo-flat with expressions*

$$a = /Z\, (\sigma\, r_0 \mid b) \quad \textit{and} \quad r = \tau\, r_0 \mid r_1 \; ,$$

and let $Y_1 \subseteq Y$ be the names of r_1. Then L and a' are of the forms

$$\begin{aligned} L &= \langle \check{\tau} \rangle \mid \check{\sigma}\, r_1 : \langle X \rangle \to \langle X' \rangle \\ a' &= /Z\, (\hat{\sigma}\, r' \mid \hat{\tau}\, b) : \langle X' \rangle \; , \end{aligned}$$

where

$$\begin{tikzcd}
X \otimes Z \arrow[r, "\hat{\tau}"] & X' \otimes Z \\
V \arrow[u, "\sigma"] \arrow[r, "\tau"'] & Y \arrow[u, "\hat{\sigma}"']
\end{tikzcd}$$

is pushout and the substitutions satisfy $\hat{\sigma} \restriction Y_1 = \check{\sigma} \otimes Z$ *and* $\hat{\tau} = \check{\tau} \otimes \mathsf{id}_Z$.

Proof The transition $a \xrightarrow{L} a'$ has an underlying IPO as shown in (a) below, such that $a' = D \circ r'$. Thus the required expression for a' will follow by establishing

$$D = /Z\,(\langle\hat{\sigma}\rangle \,|\, \hat{\tau}\, b)\ .$$

Using the expressions for a and r we refine the diagram (a) as (b). Since r_0 is epi, the lower square in (b) is pushout, and hence the upper square is an IPO.

$$\begin{tikzcd}
\langle X \rangle \arrow[r, "L"] & {} \\
\epsilon \arrow[u, "a"] \arrow[r, "r"'] & \langle Y \rangle \arrow[u, "D"']
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
\langle X \rangle \arrow[r, "L"] & {} \\
\langle V \rangle \arrow[u, "/Z\,(\sigma\,|\,b)"'] \arrow[r, "\tau\,|\,r_1"'] & \langle Y \rangle \arrow[u, "D"'] \\
\epsilon \arrow[u, "r_0"'] \arrow[uu, bend left=60, "a"] \arrow[r, "r"'] & \langle Y \rangle \arrow[u, "\langle \mathsf{id}_Y \rangle"']
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
X \otimes Z \arrow[r, "\hat{\tau}"] & U \\
V \arrow[u, "\sigma"] \arrow[r, "\tau"'] & Y \arrow[u, "\hat{\sigma}"']
\end{tikzcd}$$

(c)

Now form a pushout of the substitutions σ and τ as shown in (c); this determines $\hat{\sigma}$ and $\hat{\tau}$ up to a bijection on U, and $\hat{\sigma}$ and $\hat{\tau}$ are themselves substitutions because they have empty support and are open. Suppose $\tau(v) = \tau(v')$ for some names $v, v' \in V$ such that $\sigma(v) \neq \sigma(v)$. Then by the consistency of the upper square in (b) the names v and v' must be open in $/Z\,(\sigma\,|\,b)$. Hence, we can choose $\hat{U} = X' \otimes Z$ and $\hat{\tau} = \check{\tau} \otimes \mathsf{id}_Z$ for some X' and $\check{\tau} : X \to X'$. Now suppose for some $v \in V$ that $\tau(v) \in Y_1$. Then $\tau(v) = r_1(p)$ for some port p of r_1, and since b and r_1 have disjoint support p is not shared with $/Z\,(\sigma\,|\,b)$. Using consistency again, v must then be open in $/Z\,(\sigma\,|\,b)$, and hence $\sigma(v) \notin Z$. This shows that $\hat{\sigma}(Y_1)$ and Z are disjoint, and it follows that $/Z\,\hat{\sigma}\,r_1 = \check{\sigma}\,r_1$, where we obtain $\check{\sigma}$ from $\hat{\sigma}$ by restricting the domain to Y_1.

We then refine the upper square in the diagram (b) above as follows:

$$\begin{tikzcd}
\langle X\rangle \arrow[r, "\langle\check\tau\rangle"'] \arrow[rr, bend left, "L"] & \langle X'\rangle \arrow[r, "\hat\sigma\, r_1 \mid \mathsf{id}_{X'}"'] & \langle X'\rangle \\
\langle X\otimes Z\rangle \arrow[u, "\langle \mathsf{id}_X \otimes /Z\rangle"'] \arrow[r, shift left, "\langle\check\tau\rangle\otimes\mathsf{id}_Z"] \arrow[r, shift right, "\langle\hat\tau\rangle"'] & \langle X'\otimes Z\rangle \arrow[r, "\hat\sigma\, r_1 \mid \mathsf{id}_{X'\otimes Z}"'] & \langle X'\otimes Z\rangle \arrow[u, "\langle \mathsf{id}_{X'}\otimes /Z\rangle"] \\
\langle X\otimes Z\rangle \arrow[u, "b\mid\mathsf{id}_{X\otimes Z}"'] \arrow[r, "\langle\hat\tau\rangle"] & \langle X'\otimes Z\rangle \arrow[u, "\hat\tau\, b\mid\mathsf{id}_{X'\otimes Z}"] \arrow[r, "\hat\sigma\, r_1\mid\mathsf{id}_{X'\otimes Z}"] & \langle X'\otimes Z\rangle \arrow[u, "\hat\tau\, b\mid\mathsf{id}_{X'\otimes Z}"] \\
\langle V\rangle \arrow[u, "\langle\sigma\rangle"'] \arrow[r, "\langle\tau\rangle"] \arrow[uuu, bend left, "/Z\,(\sigma\mid b)"] \arrow[rr, bend right, "\tau\mid r_1"'] & \langle Y\rangle \arrow[u, "\langle\hat\sigma\rangle"] \arrow[r, "\mathsf{id}_Y\mid r_1"] & \langle Y\rangle\ . \arrow[u, "\langle\hat\sigma\rangle"] \arrow[uuu, bend right, "D"']
\end{tikzcd}$$

The lower square on the left is pushout by the preceding construction; the squares above it and to its right are pushout by Lemma 14.5; the remaining small square is an IPO by Lemma 14.6; and the top rectangle is tensorial. The required expressions for L and D follow by calculation of the composite arrows. ■

Corollary 14.8 (Characterising parametric transitions) *In the above theorem, suppose $r = (\mathsf{id}_W \otimes R)\circ d$ for some redex R and discrete d. Then $r_1 = (\mathsf{id}_{W_1} \otimes R_1)\circ d_1$ for some $W_1 \subseteq W$, some R_1 with $|R_1| \subseteq |R|$, and some discrete d_1 with $|d_1| \subseteq |d|$, and L has the components*

$$L^{\mathsf{par}} = \langle \mathsf{id}_X\rangle \otimes d_1$$
$$L^{\mathsf{red}} = \check\tau \mid \check\sigma R_1$$

Proof Consider the IPO pair underlying the transition:

$$\begin{tikzcd}
\langle X\rangle \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
\epsilon \arrow[u, "a"] \arrow[r, "d"'] & {} \arrow[u, "D^{\mathsf{par}}"'] \arrow[r, "\mathsf{id}_W\otimes R"'] & \langle Y\rangle\ . \arrow[u, "D"']
\end{tikzcd}$$

Clearly, $|L^{\mathsf{par}}| = |d_1|$ and $|L^{\mathsf{red}}| = |R_1|$; moreover L^{par} is discrete, since d_1 is discrete, and hence L^{par} cannot contribute nontrivially to $\check\sigma$ and $\check\tau$. The factorisation of L follows as stated. ■

We now shift our attention to abstract BRSs. As mentioned repeatedly, we have worked in concrete BRSs because they have enough structure, i.e. enough RPOs and IPOs —and even pushouts— to apply the theory that ensures congruential behavioural equivalence. Moreover, we have eased our task by working in *hard* concrete BRSs $\acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K})$, where place graphs have pushouts.

It is now open to us to apply the quotient functor of Definition 9.12$^{\mathsf{b}}$

$$[\![\cdot]\!] : \acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K}) \to \mathrm{BBG}_{\mathsf{h}}(\mathcal{K})$$

in order to transfer FPE and the congruence theorem to hard abstract BRSs. In fact this transfer is justified by the crucial Theorem 5.7 that links concrete wide reactive systems to abstract ones. The most specific result is the congruence of engaged bisimilarity in $\mathrm{BBG}_{\mathsf{h}}(\mathcal{K})$ (Corollary 13.11).

This may be appropriate for some applications, but it fails for those where the reaction rules are destructive, in the sense that they may create bigraphs with empty regions — since these are inadmissible in hard bigraphs. Consider Example 5, illustrated in Figure 6; an empty region in the reactum is created. Recall our discussion at the end of Section 13; in hard bigraphs we would have to encode the empty agent NIL of the π-calculus by not by the unit 1 of parallel product, because 1 does not exist in hard bigraphs, but by a *place node*, i.e. a $\triangle$-node where $\triangle$ is an atomic control with zero arity. Then indeed we could expect to prove the bisimilarity $\triangle \mid a \sim a$.

But, just as we treat this equation in π-calculus as a *structural congruence* NIL $\mid P \equiv P$, so in bigraphs we would hope to treat it as an *identity* of bigraphs, not just a bisimilarity. So we would like to quotient by place equivalence, which is a static congruence. We therefore define $\Bumpeq_\triangle$ to be the smallest equivalence including both $\Bumpeq$ and $\equiv_\triangle$. (We might call it *soft lean-support equivalence*.) Then, following Definition 3.6, we have the $\Bumpeq_\triangle$-quotient functor

$$[\![\cdot]\!]_\triangle : \acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K}^\triangle) \to \mathrm{BBG}(\mathcal{K}) \ .$$

Now, to transfer our dynamic theory along this functor we must show that $\Bumpeq_\triangle$ respects FPE transitions, at least in basic BRSs. We know that $\Bumpeq$ does so; it therefore remains to show that $\equiv_\triangle$ does so.

Proposition 14.9 (place equivalence respects FPE) *In any basic BRS with all redexes $\triangle$-free, place equivalence ($\equiv_\triangle$) respects* FPE *transitions.*

The proof uses Corollary 14.8, and appears in detail in Appendix A.3; it exploits flatness, although it may well hold also under weaker conditions.

We are now ready to prove the corollary that will allow us to create a tractable and congruential TS in basic (soft) abstract BRSs. The following is an exact analogue of Corollary 13.11; it makes use of the preceding proposition to replace a hard abstract BRS by a soft one.

Corollary 14.10 (engaged congruence in soft abstract BRSs) *Let $\acute{\mathbf{A}} = \acute{\mathrm{B}}\mathrm{IG}_{\mathsf{h}}(\mathcal{K}^\triangle)$ be a hard concrete BRS that is definite and basic, with all redexes $\triangle$-free. Let*

$$[\![\cdot]\!]_\triangle : \acute{\mathrm{B}}\mathrm{BG}_{\mathsf{h}}(\mathcal{K}^\triangle) \to \mathrm{BBG}(\mathcal{K})$$

be the quotient functor by $\Bumpeq_{\triangle}$, *and let* $\mathbf{A} = \mathrm{B}\textsc{bg}(\mathcal{K})$. *Let* $\sim_{\textsc{fpe}}$ *denote bisimilarity both for* FPE *in* $\acute{\mathbf{A}}$ *and for the induced transition system* $[\![\textsc{fpe}]\!]_{\triangle}$ *in* $\mathbf{A}$. *Then*

(1) $a \sim_{\textsc{fpe}} b$ *in* $\acute{\mathbf{A}}$ *iff* $[\![a]\!]_{\triangle} \sim_{\textsc{fpe}} [\![b]\!]_{\triangle}$ *in* $\mathbf{A}$.

(2) Engaged bisimilarity $\sim_{\textsc{fpe}}$ *is a congruence in* $\mathbf{A}$.

Proof The functor $[\![\cdot]\!]_{\triangle}$ is the quotient by $\Bumpeq_{\triangle}$, the smallest equivalence that includes both $\Bumpeq$ and $\equiv_{\triangle}$. We know that engaged transitions respect $\Bumpeq$, and by Proposition 14.9 they also respect $\equiv_{\triangle}$; hence they respect $\Bumpeq_{\triangle}$ and thus of course the quotient functor $[\![\cdot]\!]_{\triangle}$. Therefore this functor and the transition system FPE fulfil the conditions of Theorem 5.7 which, with the help of Corollary 13.10, yields the required results. ■

The reader will find it helpful to compare Corollaries 12.6, 13.11 and 14.10. In each case we transfer a transition system from a concrete to an abstract BRS, and in each case we show that congruence of the associated bisimilarity is preserved. The third case has the advantage not only that it deals with an *engaged* transition system, which is more tractable, but also that it works in *soft* bigraphs, which is where we would often expect to work because it is inconvenient to avoid having empty regions.

To conclude this section we outline how we would normally expect to apply Corollary 14.10, and indeed how we shall apply it in Section 15. We assume that we have to hand an abstract BRS $\mathrm{B}\textsc{bg}(\mathcal{K}$,Reacts) which is basic; that is, all its reaction rules are simple and flat, and in addition it is definite, roughly meaning that no redex is properly included in another. We wish to equip this BRS with a suitable transition system for prime free agents, in such a way that bisimilarity is a congruence. We do this in three stages.

1. We first create $\acute{\mathrm{B}}\textsc{bg}_{\mathsf{h}}(\mathcal{K}^{\triangle}$,´Reacts), a preimage of $\mathrm{B}\textsc{bg}(\mathcal{K}$,Reacts) under the quotient functor $[\![\cdot]\!]_{\triangle}$, as follows. We choose a fresh nullary atomic control $\triangle$; then for the concrete reaction rules ´Reacts we take every lean $[\![\cdot]\!]_{\triangle}$-preimage of a rule in Reacts, and insert a $\triangle$-node into each empty region of its reactum.

2. Next we equip $\acute{\mathrm{B}}\textsc{bg}_{\mathsf{h}}(\mathcal{K}^{\triangle}$,´Reacts) with the engaged transition system FPE, knowing from previous results that is associated bisimilarity $\sim_{\textsc{fpe}}$ is a congruence.

3. Finally we equip $\mathrm{B}\textsc{bg}(\mathcal{K}$,Reacts) with the transition system $[\![\textsc{fpe}]\!]_{\triangle}$, and invoke Corollary 14.10 to ensure that the associated bisimilarity $\sim_{\textsc{fpe}}$ is a congruence.

This construction corresponds to Construction 9 in [20]; however, the details here are much simpler.

It is worth remarking that the passage to concrete BRSs has more than one purpose. Not only do concrete BRSs provide RPOs which are absent in abstract ones, but they also give meaning to ‘engaged’, a notion which is not so clear in abstract transition systems.

15 Finite asynchronous π-calculus

In this section we illustrate bigraph theory by applying it to the asynchronous π-calculus of Honda and Tokoro [19] and Boudol [3]. We restrict our attention here to the fragment without replication and summation; we refer to this calculus as Aπ. We encode Aπ as a BRS and investigate the TS and bisimilarity thereby induced on it.

Aπ has processes given by the abstract syntax

$$P \;::=\; \overline{x}y \;\big|\; x(z).P \;\big|\; 0 \;\big|\; P\,|\,Q \;\big|\; \nu z\, P\,,$$

denoting output, input, inaction, parallel composition, and restriction, respectively. Dynamics is given by the single reaction rule

$$\overline{x}y \,|\, x(z).P \to \{y/z\}P\;;$$

it indicates that y is communicated along x and substituted for z in P. This rule can be applied in any context except underneath an input prefix; moreover, the input and output terms may be 'brought together' by the application of *structural congruence* $\equiv$, which relates process terms differing only by syntactical detail, such as alpha-conversion of names and reordering of parallel components.

Dynamics is also given in terms of a TS, based on which several variations of bisimilarity are defined; in asynchronous calculi, as that considered here, several standard bisimilarities (including *early*, *late* and *open*) coincide and form a congruence. We shall refer to it simply as (π-)bisimilarity.

One variant of TS for π-calculus is the so-called *early* style. It can be summarised for Aπ by the following lemma, which characterises transitions according to the structure of processes up to structural congruence:

Lemma 15.1 *Let P be a process in* Aπ. *Then*

1. $P \xrightarrow{\overline{x}y} P'$ *iff* $P \equiv \overline{x}y \,|\, P'$.
2. $P \xrightarrow{\overline{x}(y)} P'$ *iff* $P \equiv \nu y\,(\overline{x}y \,|\, P')$.
3. $P \xrightarrow{xy} P'$ *iff there exist* P_0, P_1, z *and* Z *such that* $P \equiv \nu Z\,(x(z).P_0 \,|\, P_1)$ *and* $P' \equiv \nu Z\,(\{y/z\}P_0 \,|\, P_1)$.
4. $P \xrightarrow{\tau} P'$ *iff there exist* P_0, P_1, x, y, z *and* Z *such that* $P \equiv \nu Z\,(\overline{x}y \,|\, x(z).P_0 \,|\, P_1)$ *and* $P' \equiv \nu Z\,(\{y/z\}P_0 \,|\, P_1)$.

We now define a BRS for Aπ along the lines already anticipated in Example 2.

Definition 15.2 ($\textsc{Bbg}_{\mathbf{A}\pi}$) The BRS $\textsc{Bbg}_{\mathrm{A}\pi} = \textsc{Bbg}(\mathcal{K}_{\mathrm{A}\pi}, \mathcal{R}_{\mathrm{A}\pi})$ has signature $\mathcal{K}_{\mathrm{A}\pi}$ consisting of two controls,

$$\begin{array}{lll} \mathsf{send}: & 0 \to 2 & \text{(atomic)} \\ \mathsf{get}: & 1 \to 1 & \text{(non-atomic, passive)}\,. \end{array}$$

The rule set $\mathcal{R}_{\mathrm{A}\pi}$ consists of the single reaction rule (R, R', ϱ), where

$$\begin{aligned} R &= \mathsf{send}_{xy} \mid \mathsf{get}_x \\ R' &= x \mid \ulcorner y \urcorner \end{aligned}$$

with $x \neq y$, and the instantiation $\varrho : 1 \to 1$ is the identity. ■

We are now ready to translate $\mathrm{A}\pi$-processes into bigraphs. Another conventional abbreviation will help us. Recall that in composing a wiring we write ωG for $\omega \circ G$. We now adopt the same convention for ions and data, writing $\mathsf{get}_x G$ and $\ulcorner y \urcorner G$ for $\mathsf{get}_x \circ G$ and $\ulcorner y \urcorner \circ G$. With these conventions we model processes in $\mathrm{BBG}(\mathcal{K}_{\mathrm{A}\pi})$ as follows:[10]

$$\begin{aligned} [\![\overline{x}y]\!] &= \mathsf{send}_{xy} \\ [\![x(z).P]\!] &= \mathsf{get}_x\,(z)\,[\![P]\!] \\ [\![0]\!] &= 1 \\ [\![P \mid Q]\!] &= [\![P]\!] \mid [\![Q]\!] \\ [\![\nu z\, P]\!] &= /z\,[\![P]\!]\ . \end{aligned}$$

Thus, output and input, the 'operational' ingredients in $\mathrm{A}\pi$, are modelled by an atom and a molecule, respectively, built from controls introduced specifically for the purpose; parallel composition is modelled by prime product (justifying the overloading of the symbol); and restriction is modelled by name closure. Some basic properties of the encoding are immediate:

Lemma 15.3

1. *$[\![P]\!]$ is a prime, free, busy agent $a : \langle \mathsf{fn}(P) \rangle$.*
2. *$[\![\{x/y\}P]\!] = x/y[\![P]\!]$.*
3. *For all non-input contexts C of $\mathrm{A}\pi$ there is a bigraph D such that, for all processes P, $[\![C[P]]\!] = D \circ [\![P]\!]$.*
4. *$P \equiv Q$ implies $[\![P]\!] = [\![Q]\!]$.*

Proof (Outline.) The first three clauses are proved by induction on the structure of P. For the last clause one must check that each axiom of structural congruence is respected by the translation, and the result follows from compositionality of the translation. ■

The map $[\![\cdot]\!]$ is bijective on processes (up to $\equiv$), i.e., $[\![P]\!] = [\![Q]\!]$ iff $P \equiv Q$, provided that structural congruence is taken to include the *restriction-input* axiom

$$\nu z\, x(y).P \equiv x(y).\nu z\, P \qquad z \notin \{x, y\}\ .$$

This axiom is not usually included, because it is not necessary for defining reaction. We contend, however, that the axiom is entirely natural; it respects bisimilarity, so it does not change the behavioural theory of the calculus.

[10] Do not confuse this translation function $[\![\cdot]\!]$ with quotient functors used in earlier sections.

Reaction in $\textsc{Bbg}_{\mathrm{A}\pi}$ accurately models reaction in $\mathrm{A}\pi$. To state this formally we first deal with a slight complication. One might expect $P \to P'$ to imply $\llbracket P \rrbracket \longrightarrow\!\!\triangleright \llbracket P' \rrbracket$, but this is not true in general. As a simple counterexample, consider the $\mathrm{A}\pi$-reaction $\overline{x}y \mid x(z).P \to \{y/z\}P$, and suppose that x does not occur free in P (but z does). The corresponding reaction in $\textsc{Bbg}_{\mathrm{A}\pi}$ is

$$\llbracket \overline{x}y \mid x(z).P \rrbracket = \mathsf{send}_{xy} \mid (\mathsf{get}_x(z)\, \llbracket P \rrbracket) \longrightarrow\!\!\triangleright (y/z \llbracket P \rrbracket) \otimes x = \llbracket \{y/z\}P \rrbracket \otimes x \; .$$

The problem is that, while the $\mathrm{A}\pi$-reaction reduces the set of free names, bigraph interfaces are constant under reaction. To solve this we extend the mapping to agents with idle names by indexing it for each process P with a name set X containing all free names of P, as follows:

$$\llbracket P \rrbracket_X = \llbracket P \rrbracket \mid X \; .$$

Thus the bijection between processes and $\textsc{Bbg}_{\mathrm{A}\pi}$-agents is extended to cover *all* prime free agents (not just the busy ones); for such an agent a we shall denote the corresponding process —unique up to structural congruence— by a_π. We can then state the correspondence between reaction in $\mathrm{A}\pi$ and $\textsc{Bbg}_{\mathrm{A}\pi}$.

Theorem 15.4 (Dynamics correspondence) *For each process P and agent $a : \langle X \rangle$,*

$$\llbracket P \rrbracket_X \longrightarrow\!\!\triangleright a \quad \textit{iff} \quad P \to a_\pi \; .$$

Now that we have obtained an accurate bigraphical model of reaction in $\mathrm{A}\pi$, we are interested in what equivalence is induced on $\mathrm{A}\pi$ by bisimilarity in $\textsc{Bbg}_{\mathrm{A}\pi}$. A different, but related, question is how the TSs of $\mathrm{A}\pi$ and $\textsc{Bbg}_{\mathrm{A}\pi}$ relate. In order to investigate these issues one might choose either to work in $\mathrm{A}\pi$ (and reflect $\textsc{Bbg}_{\mathrm{A}\pi}$-bisimilarity and -transitions back into $\mathrm{A}\pi$), or to work in $\textsc{Bbg}_{\mathrm{A}\pi}$ (and analyse the images under $\llbracket - \rrbracket$ of the relevant $\mathrm{A}\pi$-relations). We choose the former approach, which enables us to use the well-developed theory of π-calculus as much as possible.

Definition 15.5 (induced bisimilarity) $\underset{\mathrm{ind}}{\sim}$ is the smallest relation on $\mathrm{A}\pi$-processes such that $a_\pi \underset{\mathrm{ind}}{\sim} b_\pi$ whenever $a \sim b$. ■

Theorem 15.6 (congruence) $\underset{\mathrm{ind}}{\sim}$ *is a congruence.*

Proof By Lemma 15.3(3) congruence of $\underset{\mathrm{ind}}{\sim}$ with respect to non-input contexts follows immediately from the general congruence property of bigraph bisimilarity (Theorem ref??). We treat input-contexts separately. Suppose $P \underset{\mathrm{ind}}{\sim} Q$ and let $C = x(z).[\cdot]$. Then $P = a_\pi$ and $Q = b_\pi$ for some bisimilar a and b. Let $a_1 = \mathsf{get}_x(z)\,(a \mid z)$ and $b_1 = \mathsf{get}_x(z)\,(b \mid z)$; these are bisimilar because their only engaged transitions are of the forms

$$\begin{aligned} a_1 &\longrightarrow\!\!\triangleright \mathsf{id} \mid \mathsf{send}_{xy} y/z(a \mid z) \\ b_1 &\longrightarrow\!\!\triangleright \mathsf{id} \mid \mathsf{send}_{xy} y/z(b \mid z) \; , \end{aligned}$$

and the right-hand sides are bisimilar by congruence of bigraph bismilarity. The result then follows, observing that $(a_1)_\pi = C[P]$ and $(b_1)_\pi = C[Q]$. ■

Lemma 15.7 *The following three statements are equivalent:*

1. $P \sim_{\text{ind}} Q$
2. $[\![P]\!]_X \sim [\![Q]\!]_X$ *for some* $X \supseteq \mathsf{fn}(P, Q)$
3. $[\![P]\!]_X \sim [\![Q]\!]_X$ *for all* $X \supseteq \mathsf{fn}(P, Q)$.

Proof The implication $(2) \Rightarrow (1)$ is immediate, noting that $([\![P]\!]_X)_\pi = P$ for any X. The implication $(3) \Rightarrow (2)$ is also immediate, since $\mathsf{fn}(P, Q)$ is finite and hence a proper subset of $\mathcal{N}$. For the implication $(1) \Rightarrow (3)$, suppose $P \sim_{\text{ind}} Q$. Then $P = a_\pi$ and $Q = b_\pi$ for some $a, b : \langle Y \rangle$ such that $a \sim b$ and $Y \supseteq \mathsf{fn}(P, Q)$. Let $a_0 = [\![P]\!]_Z$ and $b_0 = [\![Q]\!]_Z$, where $Z = \mathsf{fn}(P, Q)$. Then $a = a_0 \,|\, Y$ and $b = b_0 \,|\, Y$, and so by Proposition 13.8, $a_0 \,|\, X \sim b_0 \,|\, X$ for any X. If $X \supseteq \mathsf{fn}(P, Q)$ then $[\![P]\!]_X = a_0 \,|\, X$ and $[\![Q]\!]_X = b_0 \,|\, X$, and the result follows. ■

In order to relate $\sim_{\text{ind}}$ to Aπ-bisimilarity, we next analyse the TS induced on Aπ by the encoding. As we shall see, many transitions in the BRS correspond closely to ordinary Aπ-transitions. Certain transitions, however, seem alien, but can be easily eliminated. The problem is that BRS-transitions are defined only up to isomorphism; this means that whenever $a \xrightarrow{L} a'$ there is also a transition $a \xrightarrow{\iota L} \iota a'$. In particular, an arbitrary name substitution can be applied to a' by including it in the label. To avoid such arbitrary substitutions we introduce the concept of *straightness*:

Definition 15.8 (straight link graph) A link graph $A : X \to Y$ is *straight* if every outer name $y \in Y$ satisfies the following condition: if y is co-open (i.e. $y \in A(X)$) then $y \in X$ and $A(y) = y$, otherwise $y \notin X$. ■

Lemma 15.9 *For every link graph A there is an isomorphism ι such that ιA is straight.*

Proof For every outer name y of A, if it is co-open pick an inner name x such that $A(x) = y$, and let $\iota(y) \overset{\text{def}}{=} x$; if not, pick a fresh name w and let $\iota(y) \overset{\text{def}}{=} w$. Clearly this defines an iso, and its construction directly ensures straightness of ιA. ■

We say that a transition $a \xrightarrow{L} a'$ is straight if the label L is straight (i.e., has a straight link graph). An immediate consequence of the preceding lemma and Proposition 5.9 is the following:

Proposition 15.10 (adequacy of straight transitions) *In every BRS the straight transitions are adequate.*

Thus we are justified in limiting attention to the straight, engaged transitions in $\mathrm{B{\scriptstyle BG}}_{\mathrm{A}\pi}$. We now define an alternative TS on Aπ whose purpose is to reflect exactly these transitions.

Definition 15.11 (induced transitions) Define the *induced labels* for Aπ as

$$\alpha ::= \overline{x}(z)S \;\big|\; xy \;\big|\; {}^{x}\!/\!_{y} \;\big|\; \tau\ .$$

The *consigned output* label $\overline{x}{\scriptstyle (z)}S$ binds the name z within the process S; we require that no other name may have more than one free occurrence in S. Moreover, we consider such labels equal when they differ only by structural congruence on S and alpha-conversion on z.

For each induced label α we declare its *agent names* $\mathsf{an}(\alpha)$ and its *environment names* $\mathsf{en}(\alpha)$ to be as follows:

$$\begin{array}{rlrl}
\mathsf{an}(\overline{x}{\scriptstyle (z)}S) = & \{x\} \qquad & \mathsf{en}(\overline{x}{\scriptstyle (z)}S) = & \mathsf{fn}(S) \setminus \{z\} \\
\mathsf{an}(xy) = & \{x\} & \mathsf{en}(xy) = & \{y\} \\
\mathsf{an}({}^{x}\!/\!_{y}) = & \{x, y\} & \mathsf{en}({}^{x}\!/\!_{y}) = & \emptyset \\
\mathsf{an}(\tau) = & \emptyset & \mathsf{en}(\tau) = & \emptyset\ .
\end{array}$$

The mapping $[\![\alpha]\!]_X$ into $\mathrm{BBG}(\mathcal{K}_{\mathrm{A}\pi})$ is defined when $\mathsf{an}(\alpha) \subseteq X$ and $\mathsf{en}(\alpha) \cap X = \emptyset$, and is then given by

$$\begin{aligned}
[\![\overline{x}{\scriptstyle (z)}S]\!]_X &= \langle \mathsf{id}_X \rangle \mid \mathsf{get}_x\,{\scriptstyle (z)}\,[\![S]\!] \\
[\![xy]\!]_X &= \langle \mathsf{id}_X \rangle \mid \mathsf{send}_{xy} \\
[\![{}^{x}\!/\!_{y}]\!]_X &= \langle \mathsf{id}_{X \setminus y} \mid {}^{x}\!/\!_{y} \rangle \\
[\![\tau]\!]_X &= \langle \mathsf{id}_X \rangle\ .
\end{aligned}$$

For any process P and any name set $X \supseteq \mathsf{fn}(P)$ we write $P : X$ to denote the pair (P, X). Define the transition relation $\underset{\text{ind}}{\rightarrow}$ between such pairs, labelled with induced labels, to be the smallest such that $P : X \overset{\alpha}{\underset{\text{ind}}{\rightarrow}} P' : X'$ whenever $[\![P]\!]_X \xrightarrow{[\![\alpha]\!]_X} [\![P']\!]_{X'}$ is a straight engaged transition in $\mathrm{BBG}_{\mathrm{A}\pi}$. ■

Note that the definition ensures that, in a transition $P : X \overset{\alpha}{\underset{\text{ind}}{\rightarrow}} P' : X'$, the process P provides the free names of α and the environment provides the new names, which must be fresh. This reflects the IPO property of transitions in $\mathrm{BBG}_{\mathrm{A}\pi}$: in an IPO names are essentially only equated when this is required for commutativity.

We shall prove below that, indeed, the induced transition relation $\underset{\text{ind}}{\rightarrow}$ has induced bisimilarity $\underset{\text{ind}}{\sim}$ as its associated bisimilarity; in other words, $\underset{\text{ind}}{\rightarrow}$ provides us with a coinductive characterisation in $\mathrm{A}\pi$ of $\underset{\text{ind}}{\sim}$. First we give two intermediate results characterising, respectively, the straight, engaged transitions of $\mathrm{BBG}_{\mathrm{A}\pi}$, and the induced TS in $\mathrm{A}\pi$.

Lemma 15.12 *Let $a : \langle X \rangle$ in $\mathrm{BBG}_{\mathrm{A}\pi}$, and let $a \longrightarrow La'$ be a straight, engaged transition. Then a, L and a' are of the forms*

$$\begin{aligned}
a &= /Z\,(r_a \mid b) \\
L &= \langle \sigma \rangle \mid r_L : \langle X \rangle \rightarrow \langle X' \rangle \\
a' &= \sigma\, /Z\,(\ulcorner y \urcorner c \mid b) : \langle X' \rangle\ ,
\end{aligned}$$

where, up to a bijection of names, one of the following cases holds:

case	r_a	r_L	σ	*conditions*
(1)	send_{xy}	$\mathsf{get}_x\, c$	id	$x \in X,\ c$ *discrete* *with names not in* $X \cup Z$
(2)	$\mathsf{get}_x\, c$	send_{xy}	id	$x \in X,\ y \notin X \cup Z$
(3)	$\mathsf{send}_{x_0 y} \mid \mathsf{get}_{x_1}\, c$	1	$\mathsf{id} \mid {}^{x_i}/_{x_{\bar{\imath}}}$	$x_0, x_1 \in X$
(4)	$\mathsf{send}_{xy} \mid \mathsf{get}_x\, c$	1	id	

Proof In the rule $(r, r' : \langle Y \rangle)$ underlying the transition, r and r' have the forms

$$r = \mathsf{send}_{xy} \mid \mathsf{get}_x\, d \quad \text{and} \quad r' = \ulcorner y \urcorner\, d \mid x \ ,$$

where $d : \langle (1), W \rangle$ is discrete and $Y = W \otimes \{x, y\}$. Since $\mathrm{B{\small BG}}_{\mathrm{A}\pi}$ is pseudo-flat, we can apply Theorem 14.7 to characterise the transition. Hence r can be factored as $r = \sigma_r\, r_0 \mid r_1$ and

$$a = /Z\, (\sigma_a\, r_0 \mid b) \qquad L = \langle \check{\sigma}_r \rangle \mid \check{\sigma}_a\, r_1 \qquad a' = /Z\, (\hat{\sigma}_a\, r' \mid \hat{\sigma}_r\, b)\ ,$$

where

$$\begin{tikzcd}
X \otimes Z \arrow[r, "\hat{\sigma}_r"] & X' \otimes Z \\
V \arrow[u, "\sigma_a"] \arrow[r, "\sigma_r"'] & W \otimes x \otimes y \arrow[u, "\hat{\sigma}_a"']
\end{tikzcd} \tag{1}$$

is pushout and the substitutions satisfy $\check{\sigma}_a\, r_1 = /Z\, \hat{\sigma}_a\, r_1$ and $\hat{\sigma}_r = \check{\sigma}_r \otimes \mathsf{id}_Z$. By straightness and engagedness of the transition, $\check{\sigma}_r$ is straight and $|r_0|$ non-empty. We proceed by cases, according to the factorisation of r. In the following $\underline{u}$ denotes $\sigma_a(u)$ for $u \in V$.

Case $r_0 = \mathsf{send}_{uv}$ and $r_1 = \mathsf{get}_z\, d$. Then σ_r is an iso with $\sigma_r(u) = x$ and $\sigma_r(v) = y$. Then the pushout property implies that $\check{\sigma}_r = \mathsf{id}_X$ and that $\hat{\sigma}_a$ maps x to $\underline{u}$, y to $\underline{v}$, and W bijectively to some W' such that $X' = X \otimes W'$. Then

$$a = /Z\, (\mathsf{send}_{\underline{uv}} \mid b) \qquad L = \langle \mathsf{id}_X \rangle \mid \mathsf{get}_{\underline{u}}\, (\check{\sigma}_a\, d) \qquad a' = /Z\, (\ulcorner \underline{v} \urcorner\, (\hat{\sigma}_a\, d) \mid b)\ .$$

Let $c = \hat{\sigma}_a\, d$. Then c is discrete, since d is discrete and $\hat{\sigma}_a$ is bijective on W. Case (1) of the table follows.

Case $r_0 = \mathsf{get}_u\, e$ and $r_1 = \mathsf{send}_{xy}$. Then σ_r is an iso with $\sigma_r(u) = x$ and $\sigma_r\, e = d$. Then the pushout property implies that $\check{\sigma}_r = \mathsf{id}_X$ and that $\hat{\sigma}_a$ maps x to $\underline{u}$ and y to some y' such that $X' = X \otimes y'$, and moreover $\hat{\sigma}_a\, d = \sigma_a\, e$. Then

$$a = /Z\, (\mathsf{get}_{\underline{u}}\, (\sigma_a\, e) \mid b) \qquad L = \langle \mathsf{id}_X \rangle \mid \mathsf{send}_{\underline{u}y'} \qquad a' = /Z\, (\ulcorner \underline{v} \urcorner\, (\sigma_a\, e) \mid b)\ .$$

Let $c = \sigma_a\, e$; case (2) of the table follows.

Case $r_0 = \mathsf{send}_{uv} \mid \mathsf{get}_w\, e$ and $r_1 = 1$. Then $\sigma_r(u) = \sigma_r(w) = x$, $\sigma_r(v) = y$ and $\sigma_r\, e = d$. If $\underline{u} \neq \underline{w}$ then the pushout property implies that $\underline{u}, \underline{w} \in X$ and either

$X' = X \setminus \underline{w}$ and $\check{\sigma}_r = \mathsf{id}_{X'} \,|\, {}^{\underline{u}}\!/\!{}_{\underline{w}}$, or the symmetric case with $\underline{u}$ and $\underline{w}$ swapped. We assume the former. Then the pushout property further implies $\hat{\sigma}_a(x) = \underline{u}$, $\hat{\sigma}_a(y) = \underline{v}$, and $\hat{\sigma}_a\, d = {}^{\underline{u}}\!/\!{}_{\underline{w}}\, \sigma_a\, e$. Then

$$a = /Z\,(\mathsf{send}_{\underline{uv}} \,|\, \mathsf{get}_{\underline{w}}\,(\sigma_a\, e) \,|\, b) \qquad L = \langle \mathsf{id}_X \rangle$$
$$a' = /Z\,(\ulcorner \underline{v} \urcorner\,({}^{\underline{u}}\!/\!{}_{\underline{w}}\, \sigma_a\, e) \,|\, {}^{\underline{u}}\!/\!{}_{\underline{w}}\, b) = {}^{\underline{u}}\!/\!{}_{\underline{w}}\, /Z\,(\ulcorner \underline{v} \urcorner\,(\sigma_a\, e) \,|\, b)\ .$$

Let $c = \sigma_a\, e$; case (3) of the table follows.

If instead $\underline{u} = \underline{w}$ then the pushout property implies that $\check{\sigma}_r = \mathsf{id}_X$, that $\hat{\sigma}_a$ maps x to $\underline{u} = \underline{w}$ and y to $\underline{v}$, that $\hat{\sigma}_a\, d = \sigma_a\, e$, and that $X' = X$. Then

$$a = /Z\,(\mathsf{send}_{\underline{uv}} \,|\, \mathsf{get}_{\underline{w}}\,(\sigma_a\, e) \,|\, b) \qquad L = \langle \mathsf{id}_X \rangle \qquad a' = /Z\,(\ulcorner \underline{v} \urcorner\,(\sigma_a\, e) \,|\, b)\ .$$

Let $c = \sigma_a\, e$; case (4) of the table follows. ■

The following lemma follows straightforwardly from the preceding one.

Lemma 15.13 *If $P : X \xrightarrow[\text{ind}]{\alpha} P' : X'$ then $\mathsf{an}(\alpha) \subseteq \mathsf{fn}(P)$ and $\mathsf{en}(\alpha) = X' \setminus X$ and P, α and P' are of the form given by one of the following cases:*

case	P	α	P'
(1)	$\nu Z\,(\overline{x}y \,\|\, P_0)$	$\overline{x}(z)S$	$\nu Z\,(P_0 \,\|\, \{{}^{y}\!/\!{}_{z}\}S)$
(2)	$\nu Z\,(x(z).P_0 \,\|\, P_1)$	xy	$\nu Z\,(\{{}^{y}\!/\!{}_{z}\}P_0 \,\|\, P_1)$
(3)	$\nu Z\,(\overline{x_0}y \,\|\, x_1(z).P_0 \,\|\, P_1)$	${}^{x_i}\!/\!{}_{x_{\overline{\imath}}}$	$\{{}^{x_i}\!/\!{}_{x_{\overline{\imath}}}\}(\nu Z\,(\{{}^{y}\!/\!{}_{z}\}P_0 \,\|\, P_1))$
(4)	$\nu Z\,(\overline{x}y \,\|\, x(z).P_0 \,\|\, P_1)$	τ	$\nu Z\,(\{{}^{y}\!/\!{}_{z}\}P_0 \,\|\, P_1)$

We are now ready to prove that the transition relation $\underset{\text{ind}}{\rightarrow}$ characterises $\underset{\text{ind}}{\sim}$ in essentially the standard sense. The equivalence $\underset{\text{ind}}{\sim}$ relates processes, whereas $\underset{\text{ind}}{\rightarrow}$ is over pairs of processes and name sets. This mismatch is remedied simply by requiring transitions to involve pairs with name sets large enough to include all free names of both of the processes we are relating.

Lemma 15.14 *$P \underset{\text{ind}}{\sim} Q$ iff $(P : X)\ \mathcal{R}\ (Q : X)$ for some $X \supseteq \mathsf{fn}(P, Q)$ and some $\underset{\text{ind}}{\rightarrow}$-bisimulation $\mathcal{R}$.*

Proof ($\Rightarrow$) Let

$$\mathcal{R} = \{(P : X, Q : X) \mid P \underset{\text{ind}}{\sim} Q \text{ and } X \supseteq \mathsf{fn}(P, Q)\}\ ;$$

we show that this is a $\underset{\text{ind}}{\rightarrow}$-bisimulation. Suppose $P \underset{\text{ind}}{\sim} Q$ and $X \supseteq \mathsf{fn}(P, Q)$, and moreover $P : X \xrightarrow[\text{ind}]{\alpha} P' : X'$. Then there is a straight engaged transition $[\![P]\!]_X \xrightarrow{[\![\alpha]\!]_X} a' \overset{\text{def}}{=} [\![P']\!]_{X'}$ in $\text{B}\textsc{bg}_{\text{A}\pi}$. Note that $P' = a'_\pi$. By Lemma 15.7 $[\![P]\!]_X \sim [\![Q]\!]_X$, so there is a matching transition $[\![Q]\!]_X \xrightarrow{[\![\alpha]\!]_X} b' : \langle X' \rangle$ such that $a' \sim b'$. Hence $P' \underset{\text{ind}}{\sim} Q' \overset{\text{def}}{=} b'_\pi$. Note that $[\![Q']\!]_{X'} = b'$. Then $Q : X \xrightarrow[\text{ind}]{\alpha} Q' : X'$ and $(P' : X')\ \mathcal{R}\ (Q' : X')$ as required.

($\Leftarrow$) For any $\underset{\text{ind}}{\rightarrow}$-bisimulation $\mathcal{R}$ we show that

$$\mathcal{R}_\pi = \{([\![P]\!]_X, [\![Q]\!]_X) \mid (P : X)\ \mathcal{R}\ (Q : X)\}$$

is a bisimulation in $\textsc{Bbg}_{\mathrm{A}\pi}$. Suppose $(P:X)\;\mathcal{R}\;(Q:X)$ and $[\![P]\!] \longrightarrow\!\!\triangleright La' : \langle X' \rangle$. Let $\alpha = L_\pi$ and $P' = a'_\pi$; then $[\![\alpha]\!]_X = L$ and $[\![P']\!]_{X'} = a'$, and so $P:X \xrightarrow[\text{ind}]{\alpha} P':X'$. Then there is a matching transition $Q:X \xrightarrow[\text{ind}]{\alpha} Q':X'$ such that $(P':X')\;\mathcal{R}\;(Q':X')$. Then $[\![Q]\!]_X \longrightarrow\!\!\triangleright Lb' \overset{\text{def}}{=} [\![Q']\!]_{X'}$ and $(a',b') \in \mathcal{R}_\pi$. ■

With the coinductive characterisation of $\underset{\text{ind}}{\sim}$ we can now address the main question of this section: relating $\underset{\text{ind}}{\sim}$ to ordinary Aπ-bisimilarity.

Theorem 15.15 (characterising induced bisimilarity) *The induced bisimilarity $\underset{\text{ind}}{\sim}$ in* Aπ *coincides with standard bisimilarity $\sim$ and barbed congruence $\simeq$.*

Proof In Aπ bisimilarity and barbed congruence coincide. It will therefore suffice to establish the two inclusions

$$\sim\; \subseteq \;\underset{\text{ind}}{\sim}\; \subseteq \;\simeq\;.$$

The last inclusion is immediate, observing that $\underset{\text{ind}}{\sim}$ is a congruence, and that $\xrightarrow[\text{ind}]{}$-transitions characterise observations and reductions as follows:

1. $P:X \xrightarrow[\text{ind}]{\overline{x}}$ iff $P\downarrow_{\overline{x}}$
2. $P:X \xrightarrow[\text{ind}]{\tau} P':X$ iff $P \to P'$.

For the first inclusion, consider the relation $\mathcal{R}$ consisting of all pairs of the form

$$\big(\,\nu Z\,(P\,|\,R):X\;,\;\nu Z\,(Q\,|\,R):X\,\big)$$

such that $P \sim Q$ and $X \supseteq \mathsf{fn}(\nu Z\,(P\,|\,R), \nu Z\,(Q\,|\,R))$. We show that $\mathcal{R}$ is a $\xrightarrow[\text{ind}]{}$-bisimulation; this will suffice by taking $Z = \emptyset$ and $R = 0$. For any such pair suppose $\nu Z\,(P\,|\,R):X \xrightarrow[\text{ind}]{\alpha} P':X'$. We proceed by cases according to the label α.

Case $\alpha = \overline{x}(z)S$. Then by Lemma 15.13 we have

$$\begin{array}{rlllll} \text{either} & \text{(i)} & P = \nu y\,(\overline{x}y\,|\,P_0) & \text{and} & P' = \nu Z\,(\nu y\,(P_0\,|\,\{{}^{y}\!/\!{}_{z}\}S)\,|\,R) \\ \text{or} & \text{(ii)} & R = \nu y\,(\overline{x}y\,|\,R_0) & \text{and} & P' = \nu Z\,(P\,|\,\nu y\,(R_0\,|\,\{{}^{y}\!/\!{}_{z}\}S)) \end{array}$$

or similar cases without the restriction on y. We assume case (i) and omit the others, which are similar or simpler. By Lemma 15.1 $P \xrightarrow{\overline{x}(y)} P_0$. Then there is a matching transition $Q \xrightarrow{\overline{x}(y)} Q_0$ for some Q_0 such that $P_0 \sim Q_0$. Then, using the lemmas in reverse order, $Q = \nu y\,(\overline{x}y\,|\,Q_0)$, and hence $\nu Z\,(Q\,|\,R):X \xrightarrow[\text{ind}]{\alpha} Q':X'$, where $Q' \overset{\text{def}}{=} \nu Z\,(\nu y\,(Q_0\,|\,\{{}^{y}\!/\!{}_{z}\}S)\,|\,R)$. Without loss of generality we can assume $y \notin \mathsf{fn}(R)$; it follows that $P' = \nu Zy\,(P_0\,|\,\{{}^{y}\!/\!{}_{z}\}S\,|\,R)$ and $Q' = \nu Zy\,(Q_0\,|\,\{{}^{y}\!/\!{}_{z}\}S\,|\,R)$. Hence, $(P',X')\;\mathcal{R}\;(Q',X')$ as required.

Case $\alpha = xy$. Similar to previous case.

Case $\alpha = {}^{x}\!/\!{}_{u}$. Using Lemma 15.13 we get several subcases. One case is $P = \nu V\,(x(z).P_0\,|\,P_1)$ and $R = \nu y\,(\overline{u}y\,|\,R_0)$ with $y \notin \mathsf{fn}(P)$, and $P' = \{{}^{x}\!/\!{}_{u}\}(\nu Zy\,(P'_0\,|\,R_0))$, where $P'_0 = \nu V\,(\{{}^{y}\!/\!{}_{z}\}P_0\,|\,P_1)$. There are symmetrical cases with the roles of x and u swapped, and with the sender in P and the recipient in R; moreover, there are cases where both the sender and receiver is in either P or R; finally, all cases can be varied by dropping the restriction on y. We omit all these additional cases, which are similar to, or simpler than, the one stated. Without loss of generality we assume $y \notin \mathsf{fn}(Q)$. By Lemma 15.1 $P \xrightarrow{xy} P'_0$ and $R \xrightarrow{\overline{u}(y)} R_0$. Then there is a matching transition $Q \xrightarrow{xy} Q'_0$ for some Q'_0 such that $P'_0 \sim Q'_0$. Then, using the lemmas in reverse order, Q is of the form $\nu W\,(x(z).Q_0 \,|\, Q_1)$ and $Q'_0 = \nu W\,(x(z).Q_0 \,|\, Q_1)$, and hence $\nu Z\,(Q \,|\, R) : X \xrightarrow[\text{ind}]{\alpha} Q' : X'$, where $Q' \overset{\text{def}}{=} \{x/u\}\nu Z y\,(Q'_0 \,|\, |\, R_0)$. Hence, $(P', X')\;\mathcal{R}\;(Q', X')$ as required.

Case $\alpha = \tau$. Similar to previous case. ■

16 Further research directions

In this final section we examine some possible further developments. We consider three kinds of development: using the model, adapting and extending the model, and deepening the model theory. Of course, further uses of the model may well entail developments of the second and third kind.

Using the model

In the previous section we have shown how to model a substantial fragment of the π-calculus as a basic BRS. Work is in progress on extending our results to the full π-calculus; this involves several largely independent developments:

Synchronous output: The calculus considered in the previous section is asynchronous in the sense that outputs have no continuations. The full calculus has outputs of the form $\overline{x}y.P$; these can be modelled using a non-atomic form of the send-control. We expect this extension to be straightforward.

Replication: Some form of replication or recursion is necessary in order to express agents with infinite behaviour. The simplest form to handle is replicated input, an encoding of which is outlined in Example 2. A slight complication arises in the handling of restriction in the presence of replication, because we must now be careful about the location of a restriction. As an example, consider the processes $\nu z\,(!x(y).P)$ and $!x(y).\nu z\,P$; in the former, any free occurrence of z in P will be shared among copies of P, whereas in the latter each copy of P will have a private copy of z. Because of this we cannot, as we have done so far, handle all restrictions by name closure. A version of the !get-control that binds ports to be shared among copied agents seems to provide a simple solution.

Whereas replicated input amounts to an extra variant of reaction, it is standard in the π-calculus to introduce replication instead as a structural notion by including the axiom $!P \equiv P\,|\,!P$. This seems to elude a direct graphical modelling, but it might still be possible to work with a WRS obtained by quotienting the BRS by the equivalence induced by the axiom.

Summation: Example 3 outlines the encoding of summation of inputs. (An extension to cover also 'synchronous' outputs is immediate.) Such a reaction rule, however, departs from the conditions of basic BRSs, because the redex is not flat. Thus, some refinement of the theory is necessary in order to obtain the results; we believe that the flatness constraint can be weakened sufficiently by adopting a notion of sorting on bigraphs; summations and actions (inputs or outputs) will then have different sorts, and the redex of Example 3 will still be 'essentially flat', so that e.g. pseudo-flatness of transitions can still be established.

Encodings of other calculi than the π-calculus are of interest, too. One example is the ambient calculus, an encoding of which is outlined in Example 4. We intend to pursue this in detail, and to compare our resulting transitions and congruences with those that already exist, for example by Merro and Hennessy [24]. Note that the rule illustrated in our example violates the flatness constraint on redexes, and —unlike π-calculus with summation— the ambient calculus does not seem to be subject to an obvious notion of sorting. Thus, to encode ambient calculus we may need either to find other means of relaxing the flatness condition, or alternatively to work in a hard BRS. In any case one may note that the ambient calculus (in its purely migrationary form) is simpler than π-calculus in one important aspect: it does not employ binding.

As well as wishing to establish a firm link with existing process calculi, we also wish to explore beyond them. We may wish to combine existing calculi, or to set up new ones. In either case, both for analysis and for programming, the algebraic formulation of bigraphs is important, and the preliminary algebraic results of Section 10 provide a promising start. Combining these with the convenient notations suggested in our illustrations (Section 2), we propose to define a generic bigraphical programming language. It will allow systems designers to explore new design structures for mobile systems, thus providing an essential experimental tool for assessing the power and tractability of the model.

One particular line of modelling is already being explored by Cardelli [5]; this concerns the use of bigraphs to model biological processes. Building on an original model by Shapiro et al [38] that used the π-calculus for this purpose, Cardelli has shown that more direct modelling is possible using ambient-like reaction rules. Since the bigraph model embraces both the π-calculus and ambients, Cardelli is able to show how to map his rules into bigraphs without any extension to the latter.

But such experimental usage typically exposes the need to adapt or extend the bigraph model to accommodate real-world phenomena that lie beyond its present scope. One of these is a stochastic treatment of non-determinism; this was important in the cited paper of Shapiro et al, in order run simulations in the π-calculus model and check them against observed behaviour. Another real-world extension is to add the continuum, to allow continuous reactions. We comment on both these extensions below.

Adapting and extending the model

We have formulated bigraphs to admit a wide variety of dynamical systems, including existing process calculi. How much wider can we go? There are many directions to look, and our present model does not appear to block any of them.

In one direction, we may try to refine our locational structure. In particular, one can easily think of uses for a model whose locations —or regions— are not forced to be nested. For example, an agent may reside at a geographical location, say Cambridge, and may also be part of a national research network or a multinational business process; these two locations —one physical, one virtual— may overlap, neither lying within the other. To model this, our place graphs must become directed acyclic graphs, not forests. What effect does this have upon the theory? Difficulties could arise with RPOs (Section 7), with the algebra (Section 10) or with the programming language (suggested by examples in Section 2).

Another direction is in the form of our reaction rules. Why have we confined ourselves to transitions of *ground* bigraphs, i.e. those with domain ϵ? There are a number of inter-related issues here, which we chose not to tackle in this study. As Sewell [37] points out, we might consider reactions of non-ground contexts. Suppose for example that (R, R', ϱ) is a reaction rule, and $d = d_0 \otimes d_1$ a possible parameter for R. Then we may wish to allow all the following reactions:

$$R \circ d \longrightarrow\!\!\!\triangleright\, R' \circ d \qquad R \circ (d_0 \otimes \mathsf{id}) \longrightarrow\!\!\!\triangleright\, R' \circ (d_0 \otimes \mathsf{id}) \qquad R \longrightarrow\!\!\!\triangleright\, R' \ ,$$

although only the first is a ground reaction. We have avoided this because we wished to allow the reaction to replicate or discard parts of a parameter d. In our model this means that R' is not proper context but rather a schematic bigraph, since our contexts are *linear*, i.e. composition does not entail discard or replication. In fact, the result of the first reaction above is in general not $R' \circ d$ but $R' \circ \varrho(d)$, where ϱ performs the appropriate replications and discards. So, for all three of the above reactions to make sense, our theory must admit *non-linear* contexts, which is a non-trivial matter. It is not clear how composition of bigraphs would treat support, and it is not clear whether RPOs would exist. We leave this question for further research.

There is a strong challenge to represent real-time and hybrid systems, if we wish our model to embrace not only communication networks but also the physical devices to which they are connected (or within which they are embedded). Process calculi are moving in this direction. As far as real-time is concerned, there is already much research on timed transitions, we would hope to adopt similar approaches for bigraphs.

For hybrid systems, an approach very relevant to bigraphs is the Φ-calculus of Rounds *et al* [35], which combines the mobility of the π-calculus with differential equations for the behaviour of real (i.e. continuous) variables. Nothing in our formulation prevents a control signature from being denumerably infinite or even a continuum; for example, a family of controls indexed by the real numbers to represent distance. Then a differential equation —say relating several distances and their rates of change— can be modelled by a reaction rule representing infinitesimal variation. We could then represent the Φ-calculus as a BRS, which may then provide useful metatheory for the former. No doubt there are technical hurdles to overcome —not least in the handling of infinitesimals— but the approach seems worth investigation.

Finally, for many purposes of modelling, the non-determinism of the reaction relation (generated by the reaction rules of an arbitrary BRS) needs to be refined by a stochastic treatment. There is a considerable body of work on stochastic process calculi, and in particular the stochastic π-calculus by Priami [33]; this has already been exploited [34] in Shapiro's project to model biological processes. There are rich opportunities for modelling other real-life mobile processes, such as the applications on the Worldwide Web, using a stochastic treatment of bigraphs. Just as with biological processes, stochastics will provide the opportunity to compare bigraphical simulations with experiment, offering a way to validate a bigraphical model.

Deepening the model theory

The bigraph model is based on supported precategories, after much effort to find the mathematical medium best suited to express a uniform behavioural theory. It is still possible that other categorical structures satisfy our needs in a more standard way. Though supported precategories are well-behaved, they do not appear to be much used elsewhere. Sassone and Sobocinski have begun to investigate the use of 2-categories, in which the strict commutation of diagrams is relaxed by admitting second-order arrows. They note that if these arrows form a groupoid then bigraphs can be modelled, and RPOs turn into GRPOs (i.e. groupoid RPOs). Their work is consistent with ours, so may be useful in providing access to existing categorical results, while the precategorical approach may be retained for detailed theoretical analysis.

In this memorandum we have placed strong emphasis on transition systems and behavioural congruence. These notions have allowed us to form useful connections with existing process theory, but they are not suitable for every form of analysis. Also important are algebraic systems such as the CSP failures pre-order [18], which allow specification and implementation to be expressed and matched in the same medium. There is no reason why these models should not be adaptable to bigraphs; indeed Leifer [22] has already shown, in an abstract setting (the forerunner of our WRSs) that the failures preorder is a precongruence for RPO-derived transition systems, just as it is in CSP.

Process theory also has strong tradition of non-standard logics such as temporal logic or the modal μ-calculus; these allow incremental analysis of processes, because simple properties (as opposed to full specifications) of a system can be expressed and verified one by one. For bigraphs, the obvious challenge is to find a logic that is *spatial* as well as temporal. Indeed, work by Caires and Cardelli on spatial logics for mobile ambients [6] has already been under way for a few years, and provides a very promising starting point for a logic for bigraphs.

As a final direction for theoretical development, we may wish to refine the notion of *wide reactive system* (WRS). Recall from Section 4 that it was designed as an abstract framework in which transitions and behavioural congruence could be derived for systems with locality, allowing reaction to occur between remote components. But BRSs have many structural properties absent in WRSs, and they therefore enjoy a more refined theory; for example, RPOs can be shown to exist and IPOs can be fully characterised. At the same time, bigraphs as we have defined them are somewhat arbitrary. It is therefore worth asking whether we can impose axioms upon WRSs that are satisfied by bigraphs, but allow the theory to be derived via the axioms rather than in a fully specified model such as bigraphs. This axiomatic theory may come closer to the essence of mobile distributed systems.

Conclusion As we said at the outset, our model based on bigraphs is a pilot study. Here and there we have made arbitrary choices, with the aim not only to explore a topographical theory of mobile systems in its own right, but to see whether it might generalise existing process theories. We hope to have demonstrated some success. This work is best considered not as a definitive theory, but as a study of possible ingredients of such a theory, and as an incentive to develop it more thoroughly.

References

[1] Bergstra, J.A. and Klop, J.W. (1985), Algebra for communicating processes with abstraction. Theoretical Computer Science 37, pp77–121.

[2] Berry, G. and Boudol, G. (1992), The chemical abstract machine. Journal of Theoretical Computer Science, Vol 96, pp217–248.

[3] Boudol, G. (1992), Asynchrony and the π-calculus. Rapport de Recherche RR–1702, INRIA Sophia Antipolis.

[4] Brookes, S.D., Hoare, C.A.R. and Roscoe, A.W. (1984), A theory of communicating sequential processes. J. ACM 31, pp560–599.

[5] Cardelli, L. (2003) Bioware languages. In Computer Systems: Papers for Roger Needham, Springer Monographs in Computer Science, to appear, 6pp.

[6] Caires, L. and Cardelli, L. (2001), A spatial logic for concurrency (Part I). Proc. 4th International Symposium on Theoretical Aspects of Computer Software, LNCS 2215, Springer Verlag, pp1–37.

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

[8] Castellani, I. (2001), Process algebras with localities. *Handbook of Process Algebra*, eds Bergstra, Ponse and Smolka, Elsevier, pp947–1045.

[9] Cattani, G.L., Leifer, J.J. and Milner, R. (2000), Contexts and Embeddings for closed shallow action graphs. University of Cambridge Computer Laboratory, Technical Report 496. [Submitted for publication.] Available at `http://pauillac.inria.fr/˜leifer`.

[10] Drewes, F., Hoffmann, B. and Plump, D. (2000) Hierarchical graph transformation. In: Foundations of Software Science and Computation Structures, LNCS 1784, Springer Verlag.

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

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

[13] Fournet, C. and Gonthier, G. (1996), The reflexive Cham and the join calculus. Proc. 23rd Annual ACM Symposium on Principles of Programming Languages, Florida, pp372–385.

[14] Gadducci, F., Heckel, R. and Llabrés Segura, M. (1999), A bi-categorical axiomatisation of concurrent graph rewriting. Proc. 8th Conference on Category Theory in Computer Science (CTCS'99), Vol 29 of Electronic Notes in TCS, Elsevier Science.

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

[16] Gardner, P.A. and Wischik, L. (2000), Explicit fusions. Proc. MFCS 2000. LNCS 1893, pp373–383.

[17] Hirsch, D. and Montanari, U. (2001), Synchronised hyperedge replacement with name mobility. Proc. 12th International Conference on Concurrency Theory (CONCUR 2001), LNCS 2154, pp121–136.

[18] Hoare. C.A.R. (1985), *Communicating Sequential Processes.* Prentice Hall.

[19] Honda, K. and Tokoro, M. (1991). An object calculus for asynchronous communications. In ECOOP'91, Workshop on Object-based Concurrent Programming, LNCS 512.

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

[21] Lafont, Y. (1990) Interaction nets. Proc. 17th ACM Symposium on Principles of Programming Languages (POPL'90), pp95–108.

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

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

[24] Merro, M. and Hennessy, M. (2002), Bisimulation Congruences in Safe Ambients. Proc. 29th International Symposium on Principles of Programming Languages, Oregon, pp71–80.

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

[26] Milner, R. (1980) *A Calculus of Communicating Systems.* LNCS 92, Springer Verlag.

[27] Milner, R. (2001) Computational flux. Proc 28th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, pp220-221.

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

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

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

[31] Park, D.M.R. (1980), Concurrency and automata on infinite sequences. LNCS 104, Springer Verlag.

[32] Parrow, J. and Victor, B. (1998), The fusion calculus: expressiveness and symmetry in mobile processes. Proc. LICS'98, IEEE Computer Society Press.

[33] Priami, C. (1995), Stochastic π-calculus. Computer Journal, Vol 38 (7), pp578–589.

[34] Priami, C., Regev, A., Silverman, W. and Shapiro, E. (2001), Application of stochastic process algebras to bioinformatics of molecular processes. Information Processing Letters, Vol 80, pp 25–31.

[35] Rounds, W.H. and Song, H. (2003), The Φ-calculus - a language for distributed control of reconfigurable embedded systems. In Hybrid Systems: Computation and control, LNCS 2263, Springer-Verlag, pp435–449.

[36] Sassone, V. and Sobocinski, P. (2002), Deriving bisimulation congruences: a 2-categorical approach. Electronic Notes in Theoretical Computer Science, Vol 68 (2), 19pp.

[37] Sewell, P. (1998), From rewrite rules to bisimulation congruences. Proc CONCUR'98, LNCS 1466, pp269–284. Full version to appear in Theoretical Computer Science, Vol 272/1–2.

[38] Regev, A., Silverman, W. and Shapiro, E. (2001), Representation and simulation of biochemical processes using the π-calculus process algebra. Proc. Pacific Symposium of Biocomputing 2001 (PSB2001), Vol 6, pp459–470.

[39] Smith, H. and Fingar, P. (2002), Business Process Management: the third wave. Amazon.com.

[40] Wojciechowski, P.T. and Sewell, P. (1999), Nomadic Pict: Language and infrastructure design for mobile agents. Proc. ASA/MA '99, Palm Springs, California.

Appendix

A Proofs

A.1 Proofs for place graphs

We begin by justifying the RPO construction for place graphs, from Section 7. First we restate it, for convenience.

Construction 7.7 (RPOs in place graphs) We construct an RPO $(\vec{B}\colon \vec{m} \to \hat{m}, B\colon \hat{m} \to p)$ for a pair $\vec{A}\colon \ell \to \vec{m}$ of place graphs relative to a bound $\vec{D}\colon \vec{m} \to p$ in three stages.

nodes: If V_i are the nodes of A_i $(i = 0, 1)$ then the nodes of D_i are $V_{\bar{\imath}} \setminus V_2 \uplus V_3$ for some V_3. Define the nodes of B_i and B to be $V_{\bar{\imath}} \setminus V_2$ $(i = 0, 1)$ and V_3 respectively.

interface: Construct the shared codomain $\hat{m}$ of $\vec{B}$ as follows. First, define the roots in each m_i that must be mapped into $\hat{m}$:

$$m'_i \stackrel{\text{def}}{=} \{r \in m_i \mid D_i(r) \in V_3 \uplus p\} \ .$$

Next define, on the disjoint sum $m'_0 + m'_1$, the equivalence $\cong$ to be the smallest for which $(0, r_0) \cong (1, r_1)$ whenever $A_i(w) = r_i$ $(i = 0, 1)$ for some $w \in \ell \uplus V_2$. Then define the codomain up to isomorphism by

$$\hat{m} \stackrel{\text{def}}{=} (m'_0 + m'_1)/{\cong} \ .$$

For each $r \in m'_i$ we denote the $\cong$-equivalence class of (i, r) by $\widehat{i, r}$.

parents: Define B_0 to simulate D_0 as far as possible (B_1 is similar):

$$\begin{array}{lll}
\text{for } r \in m_0 : & B_0(r) \stackrel{\text{def}}{=} & \begin{cases} \widehat{0, r} & \text{if } r \in m'_0 \\ D_0(r) & \text{if } r \notin m'_0 \end{cases} \\
\text{for } v \in V_1 \setminus V_2 : & B_0(v) \stackrel{\text{def}}{=} & \begin{cases} \widehat{1, r} & \text{if } A_1(v) = r \in m_1 \\ D_0(v) & \text{if } A_1(v) \notin m_1 \ . \end{cases}
\end{array}$$

Finally define B, to simulate both D_0 and D_1:

$$\begin{array}{ll}
\text{for } \hat{r} = \widehat{i, r} \in \hat{m} : & B(\hat{r}) \stackrel{\text{def}}{=} D_i(r) \\
\text{for } v \in V_3 : & B(v) \stackrel{\text{def}}{=} D_i(v) \ .
\end{array}$$

■

Lemma A.1 *The definition in Construction 7.7 is sound.*

Proof The second clause defining $B_0(r)$ is sound, since if $r \notin m'_0$ then by definition $D_0(r) \in V_1 \setminus V_2$, which is indeed the node set of B_0. Similar reasoning applies to the second clause defining $B_0(v)$.

The first clause defining $B_0(v)$ is sound, since if $A_1(v) = r$ with $v \in V_1 \setminus V_2$ then we have $r \in m'_1$; for if not, then $D_1(r) \in V_0 \setminus V_2$, which is impossible since $D_1 \circ A_1 = D_0 \circ A_0$.

Finally, the clauses defining B are sound because the right-hand sides are independent of the choice of i and of r; this is seen by appeal to the definition of $\cong$ and the equation $D_1 \circ A_1 = D_0 \circ A_0$. ∎

Lemma A.2 *$(\vec{B}, B)$ is a candidate RPO for $\vec{A}$ relative to $\vec{D}$.*

Proof To prove $B_0 \circ A_0 = B_1 \circ A_1$, by symmetry it will be enough to consider cases for $w \in \ell \uplus V_0$, and for the value of $A_0(w)$.

Case $w \in V_0 \setminus V_2$, $A_0(w) = v \in V_0$. Then $(B_1 \circ A_1)(w) = B_1(w) = D_1(w) = (D_1 \circ A_1)(w) = (D_0 \circ A_0)(w) = A_0(w) = (B_0 \circ A_0)(w)$.

Case $w \in V_0 \setminus V_2$, $A_0(w) = r \in m_0$. Then $(B_1 \circ A_1)(w) = B_1(w) = \widehat{0,r} = B_0(r) = (B_0 \circ A_0)(w)$.

Case $w \in \ell \uplus V_2$, $A_0(w) = v \in V_0 \setminus V_2$. Then $(B_0 \circ A_0)(w) = A_0(w) = v$. Also $(D_1 \circ A_1)(w) = (D_0 \circ A_0)(w) = v$, so for some $r \in m_1$ we have $A_1(w) = r$ and $D_1(r) = v$, hence $r \notin m'_1$. Then $(B_1 \circ A_1)(w) = B_1(r) = D_1(r) = v$.

Case $w \in \ell \uplus V_2$, $A_0(w) = v \in V_2$. Then $(D_1 \circ A_1)(w) = (D_0 \circ A_0)(w) = v$, so also $A_1(w) = v$. Hence $(B_1 \circ A_1)(w) = v = (B_0 \circ A_0)(w)$.

Case $w \in \ell \uplus V_2$, $A_0(w) = r_0 \in m'_0$. Then $D_0(r_0) \in V_3 \uplus p$, and so $(D_1 \circ A_1)(w) = (D_0 \circ A_0)(w) \in V_3 \uplus p$; hence for some $r_1 \in m'_1$ we have $A_1(w) = r_1$ and $D_1(r_1) = D_0(r_0)$. Hence $(B_0 \circ A_0)(w) = B_0(r_0) = D_0(r_0) = D_1(r_1) = B_1(r_1) = (B_1 \circ A_1)(w)$.

Case $w \in \ell \uplus V_2$, $A_0(w) = r \in m_0 \setminus m'_0$. Then $D_0(r) = v \in V_1 \setminus V_2$; hence $(D_1 \circ A_1)(w) = (D_0 \circ A_0)(w) = v$, so $A_1(w) = v$. So $(B_1 \circ A_1)(w) = v = D_0(r) = B_0(r) = (B_0 \circ A_0)(w)$.

We now prove $B \circ B_0 = D_0$ by case analysis.

Case $r \in m'_0$. Then $(B \circ B_0)(r) = B(\widehat{0,r}) = D_0(r)$.

Case $r \in m_0 \setminus m'_0$. Then $B_0(r) = D_0(r) \in V_0 \setminus V_2$, hence $(B \circ B_0)(r) = D_0(r)$.

Case $v \in V_1 \setminus V_2$, $D_0(v) \in V_1 \setminus V_2$. Since $D_0 \circ A_0 = D_1 \circ A_1$ we have $A_1(v) \notin m_1$, so $B_0(v) = D_0(v) \in V_1 \setminus V_2$; hence $(B \circ B_0)(v) = B_0(v) = D_0(v)$.

Case $v \in V_1 \setminus V_2$, $D_0(v) \in V_3 \uplus p$. Since $D_0 \circ A_0 = D_1 \circ A_1$ there exists $r \in m_1$ with $A_1(v) = r$; moreover we readily deduce $r \in m'_1$, so $B_0(v) = \widehat{1,r}$. Hence $(B \circ B_0)(v) = B(\widehat{1,r}) = D_1(r) = (D_1 \circ A_1)(v) = (D_0 \circ A_0)(v) = D_0(v)$.

Case $v \in V_3$. Then $(B \circ B_0)(v) = B(v) = D_0(v)$. ∎

We are now ready to prove the theorem justifying our construction:

Theorem 7.8 (RPOs in place graphs) *In ´*PLG*, Whenever a pair $\vec{A}$ of place graphs has a bound $\vec{D}$, there exists an RPO $(\vec{B}, B)$ for $\vec{B}$ relative to $\vec{D}$, and Construction 7.7 yields such an RPO.*

Proof We have already proved that the triple $(\vec{B}, B)$ built in Construction 7.7 is an RPO candidate. Now consider any other candidate $(\vec{C}, C)$ with intervening interface n. C_i has nodes $V_{\bar{i}} \backslash V_2 \uplus V_4$ $(i = 0, 1)$ and C has nodes V_5, where $V_4 \uplus V_5 = V_3$. We have to construct a unique mediating arrow $\widehat{C}$, as shown in the diagram.

$$\begin{tikzcd}
 & & p & & \\
 & & n \arrow[u, "C"] & & \\
m_0 \arrow[uurr, "D_0"] \arrow[urr, "C_0"] \arrow[rr, "B_0"] & & \hat{m} \arrow[uu, bend left, "B"] \arrow[u, dashed, "\widehat{C}"'] & & m_1 \arrow[uull, "D_1"'] \arrow[ull, "C_1"'] \arrow[ll, "B_1"'] \\
 & & {} \arrow[ull, "A_0"] \arrow[urr, "A_1"'] & &
\end{tikzcd}$$

We define $\widehat{C}$ with nodes V_4 as follows:

$$\begin{array}{ll} \text{for } \hat{r} = \widehat{i, r} \in \hat{m} : & \widehat{C}(\hat{r}) \stackrel{\text{def}}{=} C_i(r) \\ \text{for } v \in V_4 : & \widehat{C}(v) \stackrel{\text{def}}{=} C_i(v) \ . \end{array}$$

Note that the equations $\widehat{C} \circ B_i = C_i$ $(i = 0, 1)$ determine $\widehat{C}$ uniquely, since they force the above definition. We now prove the equations (considering $i = 0$):

Case $r \in m'_0$. Then $(\widehat{C} \circ B_0)(r) = \widehat{C}(\widehat{0, r}) = C_0(r)$.

Case $r \in m_0 \backslash m'_0$. Then $\mathbf{D}_0(r) \in V_1 \backslash V_2$, so $B_0(r) = D_0(r)$, hence $(\widehat{C} \circ B_0)(r) = D_0(r)$. Also since $C \circ C_0 = D_0 \in V_1 \backslash V_2$ we have $C_0(r) = D_0(r)$.

Case $v \in V_1 \backslash V_2, D_0(v) \in V_1 \backslash V_2$. Since $D_0 \circ A_0 = D_1 \circ A_1$ we have $A_1(v) \notin m_1$, so $B_0(v) = D_0(v)$, hence $(\widehat{C} \circ B_0)(v) = D_0(v)$. Also $C_0(v) = (C \circ C_0)(v) = D_0(v)$.

Case $v \in V_1 \backslash V_2, D_0(v) \in V_3 \uplus p$. Then $A_1(v) = r \in m'_1$ with $D_1(r) = D_0(v)$, and $B_0(v) = \widehat{1, r}$. So $(\widehat{C} \circ B_0)(v) = \widehat{C}(\widehat{1, r}) = C_1(r) = (C_0 \circ A_0)(v) = C_0(v)$.

Case $v \in V_4$. Then $(\widehat{C} \circ B_0)(v) = \widehat{C}(v) = C_0(v)$.

It remains to prove that $C \circ \widehat{C} = B$. The following cases suffice:

Case $\hat{r} = \widehat{0, r} \in \hat{m}, B(\hat{r}) \in V_4$. Then $(C \circ \widehat{C})(\hat{r}) = \widehat{C}(\hat{r}) = C_0(r) = D_0(r) = B(\hat{r})$.

Case $\hat{r} = \widehat{0, r} \in \hat{m}$, $B(s) \in V_5 \uplus p$. Then $D_0(r) = B(\hat{r}) \in V_5 \uplus p$, so for some $s \in n$ we have $C_0(r) = s$ and $C(s) = B(\hat{r})$. But by definition $\widehat{C}(\hat{r}) = s$, so $(C \circ \widehat{C})(\hat{r}) = C(s) = (C \circ C_0)(r) = D_0(r) = B(\hat{r})$.

Case $v \in V_4, B(v) \in V_4$. Then $(C \circ \widehat{C})(v) = \widehat{C}(v) = C_0(v) = D_0(v) = B(v)$.

Case $\in V_4$, $B(v) \in V_5 \uplus p$. Then $B(v) = D_0(v) = C(t)$, where $C_0(v) = t \in n$, and by definition $\widehat{C}(v) = C_0(v)$, so $(C \circ \widehat{C})(v) = C(t) = B(v)$.

Case $v \in V_5$. Then $(C \circ \widehat{C})(v) = C(v) = D_0(v) = B(v)$.

Hence $\widehat{C}$ is the required unique mediator; so $(\vec{B}, B)$ is an RPO. ■

We now turn to the proofs on pushout variations at the end of Section 7. Again, we restate them for convenience.

Proposition 7.15 (first pushout variation) *Let $\vec{B}$ be a bound for $\vec{A}$ in ´$\mathrm{PLG}_{\mathrm{h}}(\mathcal{K}^{\triangle})$. 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 $(A_0^{\triangle}, A_1)$.*

(a)

$$\begin{tikzcd}
& {} & \\
& {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
& {} \arrow[ul, "A_0"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
& {} & \\
& {} \arrow[u, dashed, "\widehat{C}"] & \\
{} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1^{\triangle}"'] \arrow[ul, "B_1^{\triangle}"] \\
& {} \arrow[ul, "A_0^{\triangle}"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

Proof We refer to diagrams (a) and (b) for both directions of the proof. For the forward direction ($\Rightarrow$) we assume the pushout in diagram (a), then assume the uppermost arrows to be a bound in diagram (b) and finally prove them to be a bound also in (a); for the reverse direction ($\Leftarrow$) the reasoning goes the other way. For both directions, first note that $\triangle$ has a sibling, say w (a node or site), in $A_0^{\triangle}$.

($\Rightarrow$) Assume that $\vec{B}$ is a pushout for $\vec{A}$, and let $C_0, C_1^{\triangle}$ be an arbitrary bound in (b). To establish $(B_0, B_1^{\triangle})$ as a pushout out we must find a mediator $\widehat{C}$ in (b) as shown. (Uniqueness of a mediator is ensured since all arrows are epi.)

Clearly $C_1^{\triangle}$ contains $\triangle$, but C_0 does not. Now since $A_0^{\triangle}$ has a sibling for $\triangle$, this is also sibling for $\triangle$ in $C_0 \circ A_0^{\triangle} = C_1^{\triangle} \circ A_1$; hence $C_1^{\triangle}$ has a sibling for $\triangle$. We therefore obtain a well-formed hard place graph if we form C_1 from $C_1^{\triangle}$ by omitting $\triangle$. Then $\vec{C}$ is a bound in (a), and because $\vec{B}$ is a pushout there is a mediator $\widehat{C}$ in (a).

To show that $\widehat{C}$ is also a mediator in (b) it suffices to show that $(\widehat{C} \circ B_1^{\triangle})(\triangle) = C_1^{\triangle}(\triangle)$. We now consider two cases for the sibling w for $\triangle$ in $A_0^{\triangle}$:

Case 1 The sibling w is a node shared between $A_0^{\triangle}$ and $B_1^{\triangle}$. Then w is a sibling of $\triangle$ in both $B_1^{\triangle}$ and $C_1^{\triangle}$. So $(\widehat{C} \circ B_1^{\triangle})(\triangle) = (\widehat{C} \circ B_1^{\triangle})(w) = (\widehat{C} \circ B_1)(w) = C_1(w) = C_1^{\triangle}(w) = C_1^{\triangle}(\triangle)$.

Case 2 The sibling w is a node or site shared between $A_0^{\triangle}$ and A_1. Then $A_1(w) = i \in m_1$, where i is a sibling of $\triangle$ in both $B_1^{\triangle}$ and $C_1^{\triangle}$. Make the same calculation with i in place of w.

($\Leftarrow$) Assume that $(B_0, B_1^{\triangle})$ is a pushout for $(A_0^{\triangle}, A_1)$, and let $\vec{C}$ be an arbitrary bound in (a). We need a mediator $\widehat{C}$ in (a), as shown. Consider the sibling w of $\triangle$ in $A_0^{\triangle}$.

Case 1 The sibling w is a node shared between A_0 and B_1. Then w is a sibling of $\vartriangle$ in $B_1^\vartriangle$, and is also in C_1 since $\vec{C}$ is a bound. Extend C_1 to $C_1^\vartriangle$ by adding $\vartriangle$ as a sibling of the node w. Then $(C_0, C_1^\vartriangle)$ is a bound in (b), so a mediator $\widehat{C}$ exists in (b). We require $\widehat{C}$ also to be a mediator in (a), and for this it suffices to show that $\widehat{C} \circ B_1 = C_1$. But this follows directly from the fact that $\widehat{C} \circ B_1^\vartriangle = C_1^\vartriangle$, since B_1 and C_1 are obtained from $B_1^\vartriangle$ and $C_1^\vartriangle$ just by omitting $\vartriangle$.

Case 2 The sibling w is a node or site shared between A_0 and A_1. Then because $B_1^\vartriangle \circ A_1 = B_0 \circ A_0^\vartriangle$, we have $A_1(w) = i \in m_1$, where i is a sibling of $\vartriangle$ in $B_1^\vartriangle$. Extend C_1 to $C_1^\vartriangle$ by adding $\vartriangle$ as a sibling of the site i. Then again $(C_0, C_1^\vartriangle)$ is a bound in (b), and we proceed exactly as in the previous case. ■

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

(1) If $\vec{B}$ is a pushout for $\vec{A}$, it is also 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 $\vec{\mathbf{A}}_n$, it is also a pushout for $\vec{A}$.

(a)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
 {} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
 & {} \arrow[ul, "A_0"] \arrow[ur, "A_1"'] &
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
 & {} & \\
 & {} \arrow[u, dashed, "\widehat{C}"] & \\
 {} \arrow[uur, "C_0"] \arrow[ur, "B_0"'] & & {} \arrow[uul, "C_1"'] \arrow[ul, "B_1"] \\
 & {} \arrow[ul, "A_0^\vartriangle"] \arrow[ur, "A_1^\vartriangle"'] &
\end{tikzcd}$$

Proof Recall that to establish $\vec{B}$ as a pushout in either direction we need only exhibit a mediator for each arbitrary bound, since the epi property ensures unicity of a mediator.

(1) Assume that $\vec{B}$ is a pushout for $\vec{A}$, diagram (a). Let $\vec{C}$ be an arbitrary bound for $\vec{A}^\vartriangle$, diagram (b). Then, since $\vec{A}$ are formed by omitting $\vartriangle$ from both of **A**, $\vec{C}$ is also a bound for $\vec{A}$. Hence a mediator $\widehat{C}$ exists as shown in (a). Trivially, $\widehat{C}$ is also the required mediator in (b).

(2) Assume that $\vec{B}$ is a pushout for $\vec{A}^\vartriangle$, diagram (b). Let $\vec{C}$ be an arbitrary bound for $\vec{A}$, diagram (a). Using the shared sibling w of $\vartriangle$, we shall show that $\vec{C}$ is also a bound for $\vec{A}^\vartriangle$. It suffices to prove that $(C_0 \circ A_0^\vartriangle)(\vartriangle) = (C_1 \circ A_1^\vartriangle)(\vartriangle)$. We calculate:

$$\begin{aligned}(C_0 \circ A_0^\vartriangle)(\vartriangle) &= (C_0 \circ A_0^\vartriangle)(w) = (C_0 \circ A_0)(w) \\ &= (C_1 \circ A_1)(w) = (C_1 \circ A_1^\vartriangle)(w) = (C_1 \circ A_1^\vartriangle)(\vartriangle)\,.\end{aligned}$$

Now, since $\vec{B}$ is a pushout in (b), there is a mediator $\widehat{C}$ in (b) as shown. Trivially, $\widehat{C}$ is again a mediator in (a), ensuring that $\vec{B}$ is indeed a pushout in (a) as required. ■

A.2 Proofs for link graphs

We begin by justifying the construction of RPOs for link graphs given in Section 8. We first repeat the construction, for convenience.

Construction 8.8 (RPOs in link graphs) We construct an RPO $(\vec{B}\colon \vec{X} \to \hat{X}, B\colon \hat{X} \to Z)$ for a pair $\vec{A}\colon W \to \vec{X}$ of link graphs relative to a bound $\vec{D}\colon \vec{X} \to Z$ in three stages. Since RPOs are preserved by isomorphism, we assume X_0, X_1 disjoint. We use the notational conventions introduced above.

nodes and edges: If V_i are the nodes of A_i $(i = 0, 1)$ then the nodes of D_i are $V_{\bar{\imath}} \backslash V_2 \uplus V_3$ for some V_3. Define the nodes of B_i and B to be $V_{\bar{\imath}} \backslash V_2$ $(i = 0, 1)$ and V_3 respectively. Edges E_i are treated exactly analogously, and ports P_i inherit the analogous treatment from nodes.

interface: Construct the shared codomain $\hat{X}$ of $\vec{B}$ as follows. First, define the names in each X_i that must be mapped into $\hat{X}$:

$$X_i' \overset{\text{def}}{=} \{x \in X_i \mid D_i(x) \in P_3 \uplus Z\} \ .$$

Next, on the disjoint sum $X_0' + X_1'$, define $\cong$ to be the smallest equivalence for which $(0, x_0) \cong (1, x_1)$ whenever $A_i(p) = x_i$ $(i = 0, 1)$ for some point $p \in W \uplus P_2$. Then define the codomain up to isomorphism by

$$\hat{X} \overset{\text{def}}{=} (X_0' + X_1')/\cong \ .$$

For each $x \in X_i'$, denote the $\cong$-equivalence class of (i, x) by $\widehat{i, x}$.

parents: Define B_0 to simulate D_0 as far as possible (B_1 is similar):

$$\begin{array}{lll}
\text{for } x \in X_0 : & B_0(x) \overset{\text{def}}{=} & \begin{cases} \widehat{0, x} & \text{if } x \in X_0' \\ D_0(x) & \text{if } x \notin X_0' \end{cases} \\
\text{for } p \in P_1 \backslash P_2 : & B_0(p) \overset{\text{def}}{=} & \begin{cases} \widehat{1, x} & \text{if } A_1(p) = x \in X_1 \\ D_0(p) & \text{if } A_1(p) \notin X_1 \ . \end{cases}
\end{array}$$

Finally define B, to simulate both D_0 and D_1:

$$\begin{array}{ll}
\text{for } \hat{x} \in \hat{X} : & B(\hat{x}) \overset{\text{def}}{=} D_i(x) \ \text{ where } x \in X_i \text{ and } \widehat{i, x} = \hat{x} \\
\text{for } p \in P_3 : & B(p) \overset{\text{def}}{=} D_i(p) \ .
\end{array}$$

■

The soundness of this definition can be checked in the same way as for Construction 7.7 for place graph RPOs. Next, we show that

Lemma A.3 *$(\vec{B}, B)$ is a candidate RPO for $\vec{A}$ relative to $\vec{D}$.*

Proof To prove $B_0 \circ A_0 = B_1 \circ A_1$, by symmetry it will be enough to consider cases for $p \in W \uplus P_0$, and for the value of $A_0(p)$.

Case $p \in P_0 \backslash P_2$, $A_0(p) = e \in E_0$. Then $(B_1 \circ A_1)(p) = B_1(p) = D_1(p) = (D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) = A_0(p) = (B_0 \circ A_0)(p)$.

Case $p \in P_0 \backslash P_2$, $A_0(p) = x_0 \in X_0$. Then $(B_1 \circ A_1)(p) = B_1(p) = \widehat{x_0} = B_0(x_0) = (B_0 \circ A_0)(p)$.

Case $p \in \ell \uplus P_2$, $A_0(p) = e \in E_0 \backslash E_2$. Then $(B_0 \circ A_0)(p) = A_0(p) = e$. Also $(D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) = e$, so for some $x_1 \in X_1$ we have $A_1(p) = x_1$ and $D_1(x_1) = e$, hence $x_1 \notin X_1'$. Then $(B_1 \circ A_1)(p) = B_1(x_1) = D_1(x_1) = e$.

Case $p \in \ell \uplus P_2$, $A_0(p) = e \in E_2$. Then $(D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) = e$, so also $A_1(p) = e$. Hence $(B_1 \circ A_1)(p) = e = (B_0 \circ A_0)(p)$.

Case $p \in \ell \uplus P_2$, $A_0(p) = x_0 \in X_0'$. Then $D_0(x_0) \in E_3 \uplus Z$, and so $(D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) \in E_3 \uplus Z$; hence for some $x_1 \in X_1'$ we have $A_1(p) = x_1$ and $D_1(x_1) = D_0(x_0)$. Hence $(B_0 \circ A_0)(p) = B_0(x_0) = D_0(x_0) = D_1(x_1) = B_1(x_1) = (B_1 \circ A_1)(p)$.

Case $p \in W \uplus P_2$, $A_0(p) = x_0 \in X_0 \backslash X_0'$. Then $D_0(x_0) = e \in E_1 \backslash E_2$; hence $(D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) = e$, so $A_1(p) = e$. So $(B_1 \circ A_1)(p) = e = D_0(r_0) = B_0(x_0) = (B_0 \circ A_0)(p)$.

We now prove $B \circ B_0 = D_0$ by case analysis.

Case $x \in X_0'$. Then $(B \circ B_0)(x) = B(\widehat{0, x}) = D_0(x)$.

Case $x \in X_0 \backslash X_0'$. Then $B_0(x) = D_0(x) \in E_0 \backslash E_2$, hence $(B \circ B_0)(x) = D_0(x)$.

Case $p \in P_1 \backslash P_2$, $D_0(p) \in E_1 \backslash E_2$. Since $D_0 \circ A_0 = D_1 \circ A_1$ we have $A_1(p) \notin X_1$, so $B_0(p) = D_0(p) \in E_1 \backslash E_2$; hence $(B \circ B_0)(p) = B_0(p) = D_0(p)$.

Case $p \in P_1 \backslash P_2$, $D_0(p) \in E_3 \uplus Z$. Since $D_0 \circ A_0 = D_1 \circ A_1$ there exists $x \in X_1$ with $A_1(p) = x$; moreover we readily deduce $x \in X_1'$, so $B_0(p) = \widehat{1, x}$. Hence $(B \circ B_0)(p) = B(\widehat{1, x}) = D_1(x) = (D_1 \circ A_1)(p) = (D_0 \circ A_0)(p) = D_0(p)$.

Case $p \in P_3$. Then $(B \circ B_0)(p) = B(p) = D_0(p)$. ■

We are now ready to prove the theorem justifying our construction:

Theorem 8.9 (RPOs in link graphs) *In ´LIG, Whenever a pair $\vec{A}$ of link graphs has a bound $\vec{D}$, there exists an RPO $(\vec{B}, B)$ for $\vec{B}$ relative to $\vec{D}$, and Construction 8.8 yields such an RPO.*

Proof We have already proved that the triple $(\vec{B}, B)$ built in Construction 8.8 is an RPO candidate. Now consider any other candidate $(\vec{C}, C)$ with intervening interface Y. C_i has nodes $V_{\bar{i}} \backslash V_2 \uplus V_4$ $(i = 0, 1)$ and C has nodes V_5, where $V_4 \uplus V_5 = V_3$. We have to construct a unique mediating arrow $\widehat{C}$, as shown in the diagram.

$$\begin{tikzcd}
 & & Z & & \\
 & & Y \arrow[u, "C"'] & & \\
X_0 \arrow[uurr, "D_0"] \arrow[urr, "C_0"] \arrow[rr, "B_0"] & & \hat{X} \arrow[uu, bend left, "B"] \arrow[u, dashed, "\widehat{C}"'] & & X_1 \arrow[uull, "D_1"'] \arrow[ull, "C_1"'] \arrow[ll, "B_1"'] \\
 & & {} \arrow[uull, "A_0"] \arrow[uurr, "A_1"'] & &
\end{tikzcd}$$

We define $\widehat{C}$ with nodes V_4 as follows:

$$\begin{array}{ll} \text{for } \hat{x} = \widehat{i,x} \in \hat{X} : & \widehat{C}(\hat{x}) \stackrel{\text{def}}{=} C_i(x) \\ \text{for } p \in P_4 : & \widehat{C}(p) \stackrel{\text{def}}{=} C_i(p) \ . \end{array}$$

Note that the equations $\widehat{C} \circ B_i = C_i$ $(i = 0, 1)$ determine $\widehat{C}$ uniquely, since they force the above definition. We now prove the equations (considering $i = 0$):

Case $x \in X'_0$. Then $(\widehat{C} \circ B_0)(x) = \widehat{C}(\widehat{0,x}) = C_0(x)$.

Case $x \in X_0 \backslash X'_0$. Then $\mathbf{D}_0(x) \in E_1 \backslash E_2$, so $B_0(x) = D_0(x)$, hence $(\widehat{C} \circ B_0)(x) = D_0(x)$. Also since $C \circ C_0 = D_0 \in E_1 \backslash E_2$ we have $C_0(x) = D_0(x)$.

Case $p \in P_1 \backslash P_2$, $D_0(p) \in E_1 \backslash E_2$. Since $D_0 \circ A_0 = D_1 \circ A_1$ we have $A_1(p) \notin X_1$, so $B_0(p) = D_0(p)$, hence $(\widehat{C} \circ B_0)(p) = D_0(p)$. Also $C_0(p) = (C \circ C_0)(p) = D_0(p)$.

Case $p \in P_1 \backslash P_2$, $D_0(p) \in E_3 \uplus Z$. Then $A_1(v) = x \in X'_1$ with $D_1(x) = D_0(p)$, and $B_0(p) = \widehat{1,x}$. So $(\widehat{C} \circ B_0)(p) = \widehat{C}(\widehat{1,x}) = C_1(x) = (C_0 \circ A_0)(p) = C_0(p)$.

Case $p \in P_4$. Then $(\widehat{C} \circ B_0)(p) = \widehat{C}(p) = C_0(p)$.

It remains to prove that $C \circ \widehat{C} = B$. The following cases suffice:

Case $\hat{x} = \widehat{0,x} \in X$, $B(\hat{x}) \in E_4$. Then $(C \circ \widehat{C})(\hat{x}) = \widehat{C}(\hat{x}) = C_0(x) = D_0(x) = B(\hat{x})$.

Case $\hat{x} = \widehat{0,x} \in X$, $B(\hat{x}) \in E_5 \uplus Z$. Then $D_0(x) = B(\hat{x}) \in E_5 \uplus Z$, so for some $y \in Y$ we have $C_0(x) = y$ and $C(y) = B(\hat{x})$. But by definition $\widehat{C}(\hat{x}) = y$, so $(C \circ \widehat{C})(\hat{x}) = C(y) = (C \circ C_0)(x) = D_0(x) = B(\hat{x})$.

Case $p \in P_4$, $B(v) \in E_4$. Then $(C \circ \widehat{C})(p) = \widehat{C}(p) = C_0(p) = D_0(p) = B(p)$.

Case $p \in P_4$, $B(p) \in E_5 \uplus Z$. Then $B(p) = D_0(p) = C(y)$, where $C_0(p) = y \in Y$, and by definition $\widehat{C}(p) = C_0(p)$, so $(C \circ \widehat{C})(p) = C(y) = B(p)$.

Case $p \in P_5$. Then $(C \circ \widehat{C})(p) = C(p) = D_0(p) = B(p)$.

Hence $\widehat{C}$ is the required unique mediator; so $(\vec{B}, B)$ is an RPO. ■

A.3 Proofs for a basic BRS

. We give here the proof of Proposition 14.9, that place equivalence respects the transition system FPE in a basic BRS. We begin with a technical lemma that shows how the pushout-pair underlying a transition of an agent a is affected by the addition or removal of a single place node in a. This lemma invokes Propositions 7.15 and 7.16, which are proved in Appendix A.1.

In the following we write $G >_{\triangle} F$ to mean that G is formed from F by adding a single place node.

Lemma A.4 *Let $\overline{a} >_{\triangle} a$. Then*

(1) If diagram (a) *underlies an L-transition of a, then $\overline{a}$ has an L-transition based upon either diagram* (b) *with $\overline{d} >_{\triangle} d$, or diagram* (c) *with $\overline{D} >_{\triangle} D$.*

(2) For every L-transition of $\overline{a}$, the underlying pushout pair takes the form of either diagram (b) *or diagram* (c)*, such that there is a corresponding L-transition of a based upon diagram* (a) *where either $\overline{d} >_{\triangle} d$, or respectively $\overline{D} >_{\triangle} D$.*

(a)

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "a"] \arrow[r, "d"'] & I \arrow[u, "D^{\mathsf{par}}"'] \arrow[r, "\mathsf{id}_W \otimes R"'] & J \arrow[u, "D"']
\end{tikzcd}$$

(b)

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "\overline{a}"] \arrow[r, "\overline{d}"'] & I \arrow[u, "D^{\mathsf{par}}"'] \arrow[r, "\mathsf{id}_W \otimes R"'] & J \arrow[u, "D"']
\end{tikzcd}$$

(c)

$$\begin{tikzcd}
{} \arrow[r, "L^{\mathsf{par}}"] & {} \arrow[r, "L^{\mathsf{red}}"] & {} \\
{} \arrow[u, "\overline{a}"] \arrow[r, "d"'] & I \arrow[u, "\overline{D^{\mathsf{par}}}"'] \arrow[r, "\mathsf{id}_W \otimes R"'] & J \arrow[u, "\overline{D}"']
\end{tikzcd}$$

Proof In either case, let $\triangle_u$ be the place node added to a to form $\overline{a}$. Then $\triangle_u$ has a sibling node, say u, in $\overline{a}$.

(1) Assume an L-transition of a based upon diagram (a). There are three cases according to where else u occurs in the diagram.

Case 1 The node u occurs in the parameter d. Then form $\overline{d} >_{\triangle} d$ by setting $\overline{d}(\triangle_u) \overset{\mathrm{def}}{=} d(u)$. Clearly diagram (b) commutes, and by Proposition 7.16(1) the new left-hand square is a pushout; so the diagram underlies an L-transition of $\overline{a}$.

Case 2 The node u occurs in both D^{par} and the redex R. Then because R is flat, we have $R(u) = j$, a site in J. Form $\overline{D^{\mathsf{par}}} >_{\triangle} D^{\mathsf{par}}$ and $\overline{D} >_{\triangle} D$ by setting $\overline{D^{\mathsf{par}}}(\triangle_u) \overset{\mathrm{def}}{=} D^{\mathsf{par}}(u)$ and $\overline{D}(\triangle_u) \overset{\mathrm{def}}{=} D(j)$. Then clearly diagram (c) commutes, and by Proposition 7.15 each of the new squares is a pushout; so the diagram underlies an L-transition of $\overline{a}$.

Case 3 The node u occurs in both D^{par} and D. Then form $\overline{D^{\mathsf{par}}} >_{\vartriangle} D^{\mathsf{par}}$ and $\overline{D} >_{\vartriangle} D$ by setting $\overline{D^{\mathsf{par}}}(\vartriangle_u) \overset{\text{def}}{=} D^{\mathsf{par}}(u)$ and $\overline{D}(\vartriangle_u) \overset{\text{def}}{=} D(u)$, and argue as in the previous case that diagram (c) underlies an L-transition of $\overline{a}$.

(2) Assume an L-transition of $\overline{a}$, which contains $\vartriangle_u$. In the underlying pushout pair $\vartriangle_u$ cannot occur in the redex R; so it must occur either in the parameter only, or in both of the two vertical arrows. Diagrams (b) and (c) represent these two cases, which we now analyse separately.

Case 1 The transition is based upon (b), with $\vartriangle_u$ in $\overline{d}$. Then since $\vartriangle_u$ has a sibling u in $\overline{a}$, by commutation it must also have u as a sibling in $\overline{d}$. So by omitting $\vartriangle_u$ from $\overline{d}$ we obtain a well-formed parameter d, and we also obtain a commuting diagram (a).

This diagram differs from (b) only in the left-hand square. Since u is a shared sibling of $\vartriangle_u$ in $\overline{a}$ and $\overline{d}$, we invoke Proposition 7.16(2), showing that this new square is a pushout, so diagram (a) underlies an L-transition of a.

Case 2 The transition is based upon (c), with $\vartriangle_u$ in both $\overline{D^{\mathsf{par}}}$ and $\overline{D}$. Then, since $\vartriangle_u$ has a sibling u in $\overline{a}$, by commutation it must also have a sibling in $\overline{D^{\mathsf{par}}}$ and in $\overline{D}$. (This sibling may be u, or it may be a site.) So by omitting $\vartriangle_u$ from $\overline{D^{\mathsf{par}}}$ and $\overline{D}$ we obtain a well-formed arrows D^{par} and D, and we also obtain a commuting diagram (a).

We now invoke Proposition 7.15 for each square in turn, showing that this diagram is a pushout-pair, so diagram (a) underlies an Ltransition of a. ■

Note that this lemma made use of flatness.

We can now prove the property that justifies our taking the quotient of hard place graphs by place equivalence.

Proposition 14.9 (place equivalence respects FPE**)** *In any basic BRS with all redexes $\vartriangle$-free, place equivalence respects* FPE *transitions.*

Proof Let $a \xrightarrow{L} a'$ be an FPE transition. We have three things to prove:

(1) We must show that if $a \equiv_{\vartriangle} b$ and $L \circ b$ is defined, then for some $b' \equiv_{\vartriangle} a'$ we have $b \xrightarrow{L} b'$. It will be enough to prove this for the cases $a >_{\vartriangle} b$ and $b >_{\vartriangle} a$. Now Lemma A.4 assures us of a transition $b \xrightarrow{L} b'$ whose underlying pushout pair has arrows agreeing with those for a only by place equivalence. Moreover, for any instantiation ϱ we can easily verify that $d \equiv_{\vartriangle} e \Rightarrow \varrho(d) \equiv_{\vartriangle} \varrho(e)$. Putting these two together, we find that $b' \equiv_{\vartriangle} a'$ as required.

(2) Now let M be another label such that $L \equiv_{\vartriangle} M$ — i.e. $L^{\mathsf{par}} \equiv_{\vartriangle} M^{\mathsf{par}}$ and $L^{\mathsf{red}} = M^{\mathsf{red}}$ — and $M \circ a$ is defined. We have to show that for some $a'' \equiv_{\vartriangle} a'$ we have $a \xrightarrow{M} a''$. It will suffice to prove this for the simple case when $L^{\mathsf{par}} >_{\vartriangle} M^{\mathsf{par}}$ or $L^{\mathsf{par}} <_{\vartriangle} M^{\mathsf{par}}$.

Now by Corollary 14.8 we have $L^{\mathsf{par}} = \mathsf{id} \otimes d_1$, where d_1 is the tensor product of some prime factors of d, the parameter underlying the transition of a. Then M^{par} can only be $\mathsf{id}_H \otimes e_1$ where $d_1 >_{\vartriangle} e_1$ or $d_1 <_{\vartriangle} e_1$, and if we form a new discrete parameter e from d by replacing (the factors) d_1 by the new factors e_1, then by Proposition 7.15 (in one or other direction), together with the property of instantiations already noted, we indeed obtain a pushout pair underlying a transition $a \xrightarrow{M}\!\!\triangleright a''$ with $a'' \equiv_{\vartriangle} a'$, as required.

(3) Finally, we must check that if $a \xrightarrow{L}\!\!\triangleright a'$ is an *engaged* transition, then so is the new transition $b \xrightarrow{M}\!\!\triangleright b'$ generated in cases (1) and (2) by adding or subtracting place nodes in a and L respectively. We need only note that any node shared between a and the redex R cannot be a place node (since R has none), and that the construction changes a by place nodes only, leaving R unchanged. This completes the proof. ■