On the Construction of Sorted Reactive Systems*

Lars Birkedal, Søren Debois, and Thomas Hildebrandt

Programming, Logic and Semantics Group
IT University of Copenhagen
{birkedal,debois,hilde}@itu.dk

Abstract. We develop a theory of *sorted bigraphical reactive systems*. Every application of bigraphs in the literature has required an extension, a sorting, of pure bigraphs. In turn, every such application has required a redevelopment of the theory of pure bigraphical reactive systems for the sorting at hand. Here we present a general construction of sortings. The constructed sortings always sustain the behavioural theory of pure bigraphs (in a precise sense), thus obviating the need to redevelop that theory for each new application. As an example, we recover Milner's local bigraphs as a sorting on pure bigraphs.
Technically, we give our construction for ordinary reactive systems, then lift it to bigraphical reactive systems. As such, we give also a construction of sortings for ordinary reactive systems. This construction is an improvement over previous attempts in that it produces smaller and much more natural sortings, as witnessed by our recovery of local bigraphs as a sorting.

1 Introduction

Bigraphical reactive systems is a framework proposed by Milner and others [1–4] as a unifying theory of process models and as a tool for reasoning about ubiquitous computing. For process models, it has been shown that Petri-nets [5], CCS [1], various π-calculi [4, 6, 7], the fusion calculus [8], mobile ambients [6], and Homer [9] can all be understood as bigraphical reactive systems (although transition semantics are usually captured only approximately). Moreover, Milner recently used bigraphs as a vehicle for studying confluence, using the λ-calculus as an example [3, 10]. For ubiquitous computing, bigraphical models were investigated in [11].

A bigraphical reactive system consists of a category of bigraphs and a reaction relation on those bigraphs; we can think of the bigraphs as terms modulo structural congruence and the reaction relation as term rewrite rules. The benefit of working within bigraphical reactive systems comes from their rich behavioural

* This work is funded in part by the Danish Research Agency (grants no.: 2059-03-0031 and no.: 274-06-0415) and the IT University of Copenhagen (the LaCoMoCo, BPL and CosmoBiz projects). Authors are listed alphabetically.

theory [1, 4, 2, 6, 12–14] which (a) induces a labelled transition system automatically for any reaction relation and (b) guarantees that bisimulation on that transition system is a congruence.

A category of bigraphs is formed according to a single-sorted signature, which defines the kinds of nodes found in bigraphs of that category. Single-sorted signatures are usually insufficient when we define programming languages or algebraic models: We need to constrain the combination of operators, so we need richer notions of sorting. Indeed, *every one* of [1–7, 9–11] construct a richer sorting to fit the framework of bigraphs to the problem at hand.

Alas, the behavioural theory of bigraphs applies only to single-sorted (or pure) bigraphs, not to arbitrarily-sorted extensions. Hence, *also* every one of [1–7, 9–11] must re-develop substantial parts of the behavioural theory. Worse, although some sortings are easy to construct [1, 2, 6], others require either hard work [11] or ingenuity [4, 3, 10] to achieve conceptually simple effects.

Up until now, it has been an open question what kinds of extensions would admit such a redevelopment. In this paper we provide a large class of extensions for which such a redevelopment is possible. Moreover, we give a method for automatically constructing sortings for such extensions.

The key observation is that most sortings in the literature exists solely to get rid of bigraphs that are meaningless for the application at hand. That is, most sortings exists solely to impose a predicate on the morphisms in the category of pure bigraphs. We give a method to automatically construct a well-behaved sorting for any decomposable such predicate. Here, a predicate P is decomposable iff it is true at every identity and $P(g \circ f)$ implies $P(g)$ and $P(f)$; all but one of the above-mentioned applications fall into this class. In particular, we prove that Milner's local bigraphs [10] arise as a sorting of pure bigraphs.

Thus, by identifying a large class of predicates for which we can construct well-behaved sortings, we make it easier to work with bigraphical reactive systems and we push back the limit for what we can hope to achieve with them.

Overview of the technical development We ask and answer the following two questions:

1. Which sortings sustain the behavioural theory of bigraphs?
2. How do we construct such a sorting for a given problem domain?

We answer Question 1 by giving a sufficient condition for a sorting of a reactive system to sustain the behavioural theory of the well-sorted parts of the original system. We then lift both this result and previous work on sortings for reactive systems [15, 6] to the present setting of bigraphical reactive systems. We answer Question 2 by giving a new family of sortings, closure sortings, all of which sustain the behavioural theory. In particular, we show how Milner's local bigraphs [10, 3] arise as a special case of the closure sorting.

In more detail: Question 1. Jensen [6] found a sufficient condition, safety, for a small class of sortings for bigraphs to preserve congruence properties. In [15] we moved that condition to general sortings of reactive systems.

In the present paper we complement that result with a sufficient condition for a sorting to preserve and reflect reaction and transition semantics for any well-sort of reactive system. We say that sorting with that property "has semantic correspondence." Altogether, in the setting of reactive systems we now have sufficient conditions for a sorting to reflect congruence properties and operational semantics; this is what we mean by "sustaining the behavioural theory". We then proceed to lift these conditions to bigraphical reactive systems which, despite the name, are not an instance of ordinary reactive systems. Moreover, we argue that in general, to construct a well-behaved sorting of a bigraphical reactive system, it is sufficient to construct a well-behaved sorting of the underlying reactive system.

In more detail: Question 2. As part of the safety condition mentioned above, it is required that if a decomposable context has a sorting, then the sorting can be decomposed correspondingly. In particular, the "has a sorting" predicate P on the pure category is decomposable, i.e., $P(f \circ g)$ implies $P(f)$ and $P(g)$. Thinking in terms of sorted algebra or programming languages this is a very natural condition — a refinement of a sorting should not constrain the way a well-sorted term can be decomposed.

In [15] we discovered that banning bigraphs containing particular "bad" sub-bigraphs corresponds exactly to giving a decomposable predicate. This insight gave rise to an answer to Question 2, albeit only for reactive systems: We gave, for any predicate P on the morphisms of a category, a sorting called the predicate sorting. The predicate sorting sustains the theory of reactive systems.

In practice, however, the predicate sorting turns out to provide far more sorts than did the sortings found in an ad hoc way for the applications in [1–7, 9–11]. In the present paper, we construct a new family of sortings, the closure sortings. Like the predicate sortings, closure sortings sustain the behavioural theory of both bigraphs and reactive systems. Unlike the predicate sortings, closure sortings give sorts much closer to what we find in the literature. As a (spectacular!) example, we show that Milner's Local bigraphs [10, 3] are recovered in a closure sorting, by taking Milner's scoping condition as a predicate on bigraphs.

Outline. In Section 2 we revisit Leifer and Milners' classic Reactive Systems [12, 13], a precursor to bigraphs. In Section 3, we recall the definition of a sorting of a reactive system and recall our previous generalisation of Jensen's safety condition. In Section 4, we give a general condition for a sorting of a reactive system to preserve dynamics up to a predicate (Theorem 1), partially answering Question 1. In Section 5, we give the closure sorting (Definition 12), partially answering Question 2 above (Theorem 2). In Section 6, we remark on lifting these partial answers to the setting of bigraphical reactive systems, thus arriving at complete answers to both questions. Finally, in Section 7, as an extended example, we recover local bigraphs as a full sub-sorting of a closure sorting (Theorem 3).

For want of space, proofs have been omitted from this extended abstract; they can be found in [16].

2 Reactive Systems

In this section, we recall Milner and Leifer's *Reactive Systems* [17, 13, 12]. These systems form the conceptual basis of bigraphical reactive systems. Except for the running example, this section contains no original work.

Let C be a category, and let $\mathcal{R}_\epsilon$ be an object of C. We think of morphisms with domain $\mathcal{R}_\epsilon$ as *agents* or *processes* and all other morphisms as *contexts*. A *reaction rule* (l, r) is a cospan of agents with common domain $\mathcal{R}_\epsilon$; intuitively, l and r are the left- and right-hand sides of a rewrite rule. A set $\mathcal{R}$ of reaction rules induces a *reaction relation*, $\longrightarrow$, obtained by closing reaction rules under contexts:

$$a \longrightarrow b \text{ iff } \exists f \in C, \exists (l, r) \in \mathcal{R}.\ a = f \circ l, b = f \circ r. \tag{1}$$

Altogether, these components constitute a reactive system.

Definition 1 (Reactive system). *A* reactive system *over a category C comprises a distinguished object $\mathcal{R}_\epsilon$ and a set $\mathcal{R}$ of reaction rules; the reaction rules give rise to a* reaction relation *by (1) above. We identify a reactive system with its reaction rules, writing $\mathcal{R}$ for both.*

In this definition we have omitted the notion of activity usually associated with reactive systems. Activity can be recovered as a sorting both in the case of reactive systems [16] and in the case of bigraphical reactive systems [6].

Example 1. Here is a small process language.

$$P, Q ::= \mathsf{0} \mid \mathsf{a} \mid \mathsf{s} \mid (P|Q) \tag{2}$$

These are the nil process $\mathsf{0}$, atomic processes a and s, and parallel composition of processes. As usual, we consider processes up to a structural congruence comprising the commutative monoid laws for $|$ and $\mathsf{0}$. Clearly, the set of processes is isomorphic to the free, commutative monoid over $\{\mathsf{a}, \mathsf{s}\}$; that is, a category with a single object, terms up to structural congruence as morphisms, composition $f \circ g = f|g$, and identity 0. (In this case, there is no distinction between agents and contexts: all morphisms are both.) Call this category $\mathcal{C}$. We intend a to model a normal process and s to model two processes in a synchronized state. Here are the reaction rules:

$$(\mathsf{a}|\mathsf{a},\ \mathsf{s}\ ) \qquad \text{“two processes synchronize,”} \tag{3a}$$

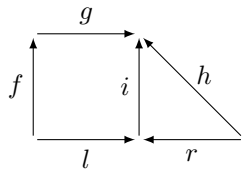
$$(\mathsf{s}\ ,\ \mathsf{a}|\mathsf{a}) \qquad \text{“processes drop synchronization.”} \tag{3b}$$

Here are two reactions, using first rule 3a, then 3b: $\mathsf{a}|\mathsf{a}|\mathsf{s} \longrightarrow \mathsf{s}|\mathsf{s} \longrightarrow \mathsf{s}|\mathsf{a}|\mathsf{a}$.

Leifer and Milner give a method for *deriving* labeled transitions for any reactive system. If the underlying category has sufficient relative pushouts (RPOs), then the bisimulation on those labeled transitions is a congruence. To construct labeled transitions, we take as labels minimal contexts enabling reaction. The notion of idem-pushout (IPOs) captures minimality[1]

[1] For brevity, we omit the definitions of both RPOs and IPOs.

Definition 2. *For a reactive system $\mathcal{R}$ over a category C, we define the* transition relation *by $f \xrightarrow{g} h$ iff there exists a context i and a reaction rule $(l, r) \in \mathcal{R}$ s.t. the following diagram commutes, and the square is an IPO.*



(4)

Example 2. $\mathcal{C}$ is isomorphic to the category of multisets over $\{\mathsf{a}, \mathsf{s}\}$ (with multiset union as composition). Thus $\mathcal{C}$ has pushouts, given by multiset subtraction, and thus RPOs. The pushout of multisets simply adds what is missing: The pushout of a and $\mathsf{a}|\mathsf{a}$ is a and $\mathsf{0}$, the pushout of a and s is s and a. Because IPOs are precisely the pushouts in this category, we find transitions for an agent a by taking the pushout of a and either left-hand side of the two rules.

$$\mathsf{a} \xrightarrow{\mathsf{a}} \mathsf{s} \qquad \text{by rule (3a)} \tag{5}$$

$$\mathsf{a} \xrightarrow{\mathsf{s}} \mathsf{a}|\mathsf{a}|\mathsf{a} \qquad \text{by rule (3b)} \tag{6}$$

There are no transitions from a with label $\mathsf{a}|\mathsf{a}$. A label can only add what is missing, and the agent a is only one "a" short of the left-hand side $\mathsf{a}|\mathsf{a}$ of rule (3a).

As mentioned, the bisimulation on such derived transition systems is a congruence whenever the underlying category has RPOs [13, 12].

Proposition 1 ([12]). *Let $\mathcal{R}$ be a reactive system on a category C. If C has RPOs, then the bisimulation on the derived transitions is a congruence.*

3 Sortings

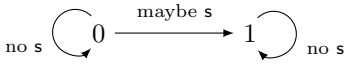
In this section, we recall the notion of *sorting* [15] for categories. As in the previous section, this section contains nothing new but the running example.

Definition 3 (Sorting). *A* sorting *of a category C is a functor $F : X \to C$ that is faithful and surjective on objects.*

We shall consistently confuse a sorting functor F with its domain: We write $F \to C$, and we speak interchangeably of F as a category and as a functor.

Example 3. Suppose we want to restrict our process language such that at most two processes are synchronized at any time, i.e., no process can contain a subterm $\mathsf{s}|\mathsf{s}$. We cannot just stipulate that there are no such terms, because the composition of s and s would then be undefined. Instead, we define a sorting $\mathcal{F} \to \mathcal{C}$ that refines the single homset of $\mathcal{C}$ into three.

The category $\mathcal{F}$ has two objects, 0 and 1. The homsets $\mathcal{F}(0,0)$ and $\mathcal{F}(1,1)$ are identical, comprising morphisms of $\mathcal{C}$ that *do not* contain an s; $\mathcal{F}(0,1)$ comprises morphisms with *at most one* s. Here is a sketch of $\mathcal{F}$.



Composition is defined as in $\mathcal{C}$; it is easy to check that this composition is well-defined on our refined homsets.

Usually when we construct a sorting $F \to C$, we will want to apply Proposition 1 to the category F. Hence, we want sortings that allow us to infer the existence of RPOs in F from the existence of RPOs in C. The following notion of *transfer* helps us do that.

Definition 4 (Transfer of RPOs). *A sorting $F \to C$* transfers RPOs *iff whenever the image of a square s in F has an RPO, then that RPO has an F-preimage that is an RPO for s.*

Jensen gives a sufficient condition, *safety*, for a sorting to transfer RPOs [6]; we generalized that condition in [15] (see also [16]; again for brevity, we omit the definition here).

Proposition 2 ([6, 15]). *Let $F \to C$ be a safe sorting. Then F transfers RPOs, and, if C has RPOs then so does F.*

4 Semantic Correspondence

In Example 3 above we used the sorting $H \to C$ to get rid of morphisms not satisfying the predicate "contain at most one s". Thus, that sorting is a *realization of a predicate* on the morphisms of C. Of the sortings in [1–7, 9–11], only the one in [7] is not a realization of a predicate.

However, not every sorting realizing a predicate is equally interesting; we must require also that the sorted category supports the same reactive systems as the original one, at least when we restrict our attention to the "good" morphisms of the original category. This semantic correspondence is part of what we mean by "sustaining the behavioural theory". In this section, towards answering Question 1, we give a sufficient condition for a sorting of a reactive system to admit such semantic correspondence.

This result generalizes our previous [15], where we proved that a particular sorting has semantic correspondence[2]. As in that paper, we will consider only predicates P that are decomposable, that is, that are true at every identity and have $P(f \circ g)$ implies $P(f)$ and $P(g)$. This restriction is not very severe; for free structures, decomposable predicates are precisely those that prohibit terms containing some given set of subterms [15].

[2] Alhough in that paper, we used the somewhat inaccurate term "preserves semantics" rather than the present "has semantic correspondence".

To formalize the notion of "semantic correspondence", we need the notion of reflection of a reactive system.

Definition 5 (Reflection of a reactive system). *Let $F \to C$ be a sorting, and let $\mathcal{R}$ be a reactive system on C. For a preimage $\hat{\mathcal{R}}_\epsilon$ of $\mathcal{R}_\epsilon$, the* reflection of $\mathcal{R}$ at $\hat{\mathcal{R}}_\epsilon$ *is $\hat{\mathcal{R}}$, where*

$$\hat{\mathcal{R}} = \{(f, g) \mid \exists x. f, g : \hat{\mathcal{R}}_\epsilon \to x \wedge (F(f), F(g)) \in \mathcal{R}\}. \tag{7}$$

We can now define semantic correspondence precisely.

Definition 6 (Correspondence of reactions, transitions). *Let $F \to C$ be a sorting, let $\mathcal{R}, \mathcal{S}$ be reactive systems on F, C, respectively, and let P be a decomposable predicate on C. The sorting $F \to C$ has* correspondence of P-transitions for $\mathcal{R}, \mathcal{S}$ *iff whenever f, g, h are morphisms of C with $P(g \circ f)$ and $P(h)$, then*

$$\begin{aligned} f \xrightarrow{\ g\ } h \quad \textit{iff} \quad \exists \hat{f}, \hat{g}, \hat{h}.\ \hat{f} \xrightarrow{\ \hat{g}\ } \hat{h} \ \textit{where} \\ F(\hat{f}) = f, F(\hat{g}) = g, F(\hat{h}) = h. \end{aligned} \tag{8}$$

We define correspondence of P-reactions similarly.

Note that despite F faithful, $\hat{f}, \hat{g}, \hat{h}$ are not necessarily unique.

Before we can define the general notion of semantic correspondence, we need first a notion of a reactive system respecting a predicate.

Definition 7 (P-respecting reactive system). *Let $\mathcal{R}$ be a reactive system and let P be a predicate on $\mathcal{R}$'s underlying category. We say that $\mathcal{R}$ is a P-*respecting reactive system *iff every rule $(l, r) \in \mathcal{R}$ has both $P(l)$ and $P(r)$.*

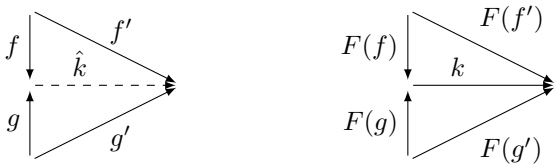
We lift the correspondence from a property of reactive systems to a property of sortings: A sorting $F \to C$ has semantic correspondence iff any P-respecting reactive system on C has a reflection in F with which it is in semantic correspondence.

Definition 8 (Semantic correspondence). *We say that a sorting $F \to C$ has* semantic correspondence up to P *iff for any P-respecting reactive system $\mathcal{R}$ on C, there exists a reflection $\hat{\mathcal{R}}$ of $\mathcal{R}$ at some $\hat{\mathcal{R}}_\epsilon$ such that F has correspondence of P-reactions and P-transitions for $\hat{\mathcal{R}}, \mathcal{R}$.*

Theorem 1 below gives a sufficient condition for a sorting to have semantic correspondence. This condition depends on the notion of "weak joint opfibration", which was introduced in [15]. Intuitively, a weak joint opfibration is a functor which has most general lifts of every cospan in its domain. The following notion of "jointly opcartesian" captures such most general lifts.

Definition 9 (Jointly opcartesian). *Let $F \to C$ be a functor. A cospan f, g in C is said to be* jointly opcartesian *iff whenever f', g' is a cospan, f, f' is a span, and g, g' is a span (see the diagram below, left side) with $F(f') = k \circ F(f)$*

and $F(g') = k \circ F(g)$ *(see the diagram below, right side), then there exists a unique lift* $\hat{k}$ *of* k *s.t.* $f' = \hat{k} \circ f$ *and* $g' = \hat{k} \circ f'$.



Example 4. In the sorting $\mathcal{F} \to \mathcal{C}$ of Example 3, the cospan a, a has a jointly opcartesian lift $\mathsf{a} : 0 \longrightarrow 0 \longleftarrow 0 : \mathsf{a}$; the cospan s, a has a jointly opcartesian lift $\mathsf{s} : 0 \longrightarrow 1 \longleftarrow 0 : \mathsf{a}$.

Armed with jointly opcartesian lifts, we define weak joint opfibrations.

Definition 10 (Weak joint opfibration [15]). *A functor* $F : X \to C$ *is a* weak joint opfibration *iff whenever* $F(f), F(g)$ *form a cospan in* C*, then there exists a jointly opcartesian pair* $\hat{f}, \hat{g}$ *with* $F(\hat{f}) = F(f)$ *and* $F(\hat{g}) = F(g)$.

Finally, getting back to our sufficient condition for a sorting to have semantic respondents, we now need only the following auxiliary definition.

Definition 11. *Let* $F \to C$ *be a sorting, let* x *be an object of* F*, and let* P *be a predicate on* C*. We say* F reflects P at x *if every morphism* $f : F(x) \to c$ *with* $P(f)$ *has a lift at* x.

Theorem 1. *Let* $F \to C$ *be a sorting, let* P *be a decomposable predicate on* C*, let* $\mathcal{R}$ *be a reactive system on* C*, and let* $\hat{\mathcal{R}}_\epsilon$ *be an* F*-preimage of* $\mathcal{R}_\epsilon$*. Then (a) the reactions of the reflection of* $\mathcal{R}$ *at* $\hat{\mathcal{R}}_\epsilon$ *correspond to the* P*-reactions of* $\mathcal{R}$ *if (i) the image of* F *is* P*, (ii)* F *reflects* P *at* $\hat{\mathcal{R}}_\epsilon$*, and (iii)* F *is a weak joint opfibration. Morever, (b) the transitions of the reflection of* $\mathcal{R}$ *at* $\hat{\mathcal{R}}_\epsilon$ *corresponds to the* P*-transitions of* $\mathcal{R}$ *if also* F *transfers and preserves RPOs. In this case,* F *has semantic correspondence up to* P.

Between them, Theorem 1 above and Proposition 2 recalled in the preceding section answer Question 1. The former gives us a sufficient condition for a sorting to admits the necessary operational semantics; the latter gives as a sufficient condition for a sorting to reflect congruence properties. Technically, this answer applies only to sortings a reactive systems, however, it is straightforward to lift the answer to the case of bigraphical reactive systems; more on such in Section 6.

5 Closure Sortings

In this section, we answer Question 2: We define, for each decomposable predicate P, a *Closure sorting* realizing P. Every closure sorting transfers RPOs and

has semantic correspondence up to P. We define closure sortings in terms of categories and reactive systems here, but in Section 6, we remark on lifting them to bigraphs.

Suppose we want to construct a sorting realizing a decomposable predicate P on the morphisms of a category C. The basic problem here is that we may have morphisms $f : x \to y$ and $g : y \to z$ which satisfy P individually but not when composed, i.e., $P(f)$ and $P(g)$, but $\neg P(g \circ f)$. At each preimage of y, we must choose whether to admit f or g. We make this choice explicit in the closure sorting by taking as pre-images for an object y pairs (F, G) of sets of morphisms such that every $g \in G$ has domain y and can be composed with every $f \in F$, that is, f has codomain y and $P(g \circ f)$. This approach leads to too many objects in the sorted category, so we further insist that (F, G) be *maximal*, that is, that adding morphisms to either F or G would violate $f \in F, g \in G \implies P(g \circ f)$.

To formalize maximality, first define $g \perp f$ iff $P(g \circ f)$. Then define, for any $c \in C$, operators $\triangle$ and $\triangledown$ by

$$\begin{aligned} \triangle F &= \{g : c \to x \mid g \perp F, \text{ any } x \in C\} \\ \triangledown G &= \{f : y \to c \mid G \perp f, \text{ any } y \in C\} , \end{aligned} \tag{9}$$

where, e.g., $g \perp F$ is lifted pointwise[3]. We can now define that $(F, G)_c$ is *maximal* iff $\triangle F = G$ and $\triangledown G = F$.

Definition 12 (Closure sorting). *Let C be a category, and let P be a decomposable predicate on C. The* closure sorting $\mathfrak{C}(P) \to C$ *has* objects $(F, G)_c$ *where c is an object of C and F, G are sets of morphisms of C s.t. every $f \in F$ and $g \in G$ has* $\mathrm{cod}(f) = c = \mathrm{dom}(g)$ *and $P(g \circ f)$. Moreover, $(F, G)_c$ must be maximal. $\mathfrak{C}(P)$ has* morphisms $k : (F, G)_c \to (H, J)_d$ *those $k : c \to d$ in C satisfying*

$$f \in F \implies k \circ f \in H \text{ and } j \in J \implies j \circ k \in G \, . \tag{10}$$

It is fairly easy to establish that $\triangle$ and $\triangledown$ form a Galois connection: $\triangle F \supseteq G$ iff $F \subseteq \triangledown G$. Thus $\triangle\triangledown$ and $\triangledown\triangle$ are indeed closure operators: $\triangle\triangledown\triangle F = \triangle F$ and $\triangledown\triangle\triangledown G = \triangledown G$. (Hence the name "closure sorting".) We can use these closure operators to "fill up" a pair $(F, G)_c$ that is not maximal, taking either $(\triangledown G, \triangle\triangledown G)_c$ or $(\triangledown\triangle F, \triangle F)_c$. This realization is crucial in establishing that the fibres of a closure sorting are lattices and, in turn, that every closure sorting satisfies the premises of Proposition 2 and Theorem 1.

Lemma 1. *Let P be a decomposable predicate on C. Then $\mathfrak{C}(P)$ (1) is safe, (2) is a weak joint opfibration, and (3) lifts P-agents at every object c of C.*

By Proposition 2 and Theorem 1, every closure sorting transfers RPOs and has semantic correspondence.

Theorem 2. *Let C be a category with RPOs, and let P be a decomposable predicate on C. Then $\mathfrak{C}(P)$ has RPOs, transfers RPOs, and has semantic correspondence up to P.*

[3] The operators $\triangle, \triangledown$ are related to the BiLog [18] adjuncts to composition.

Example 5. The sorting $\mathcal{F} \to \mathcal{C}$ of Example 3 is indeed a closure sorting for the predicate "contains at most one $\mathtt{s}$"; we recover the objects 0 and 1 as $0 = (\nabla\triangle\{\mathtt{0}\}, \triangle\{\mathtt{0}\})$ and $1 = (\nabla\{\mathtt{0}\}, \triangle\nabla\{\mathtt{0}\})$. By Theorem 2, $\mathcal{F}$ has RPOs and has semantic correspondence up to that predicate.

The closure sorting answers Question 2, "how do we construct a sorting given some problem domain" by virtue of Theorem 2. Technically, the answer applies only to reactive system, but again, it is straightforwardly lifted to the setting bigraphical reactive systems; we discussed this lifting in more detail in the next section.

6 Bigraphical Reactive Systems

As mentioned in the introduction, bigraphical reactive systems are not instances of ordinary reactive systems. Because categories of bigraphs do not contain relative pushouts, Milner, Leifer, and Jensen introduced supported pre-categories [13, 4, 1].

Intuitively, the supported pre-category of pure bigraphs adds a notion of identity of sub-terms; the totality of identities in some term is called its support, hence "supported". For the notion of support to make sense, composition of bigraphs with overlapping support is left undefined, hence "pre-category".

For each supported pre-category S we obtain a category $[S]$ by considering equal morphisms which differ only in the particular identities chosen for their support. The support quotient functor $\eta_S : S \to [S]$ takes each supported term to such a support equivalence class. The category of abstract bigraphs arise as the quotient $[S]$ of the supported pre-category S, the concrete bigraphs.

This solution, using supported pre-categories, is widely regarded as being overly ad hoc. Following a suggestion by Leifer [13], Sassone and Sobocinski [19, 20] investigated using instead first two-categories and later bi-categories as foundations for reactive systems. Unfortunately, comfortable as their results may be, no one has as yet formalised bigraphs in one of these more general settings. Hence, we stick with the present formalisation in supported pre-categories, at least for the present paper.

Fortunately, lifting Jensen's previous work and the results of the preceding sections to the setting of supported pre-categories pose no real difficulties, it only requires a lot of footwork. For want of space, we omit that footwork here; the interested reader is referred to [16]. We do, however, make the following comments.

A sorting on pure abstract bigraphs induces a sorting on the corresponding concrete bigraphs, found by taking the pullback of that sorting along the support quotient.

$$\begin{array}{ccc} F^\star & \longrightarrow & S \\ \downarrow & \lrcorner & \downarrow{\scriptstyle \eta_S} \\ F & \xrightarrow[F]{} & [S] \end{array} \tag{11}$$

Here we have concrete bigraphs S, abstract bigraphs $[S]$; $\eta_S : S \to [S]$ is the support quotient. We are given a sorting $F \to [S]$, and we find the corresponding sorting $F^\star \to S$ simply by taking the pullback (in the category of supported pre-categories) of $F \to [S]$ along η_S.

It turns out that if the original sorting functor satisfies the sufficient conditions we have seen so far, then the induced sorting functor will sustain the behavioural theory. That is, in this case, the induced sorting will preserve congruence properties and enjoy semantic correspondence. Moreover, there is essentially a bijection between the sortings of a supported pre-category (S in the above diagram) and the sortings of its support quotient ($[S]$ in the above diagram). Hence, for practical work with bigraphical models, it is sufficient to consider sortings of abstract bigraphs.

Readers who find all this a bit hand-wavy are again referred to [16], where they will find formalisation. In particular, Proposition 2, Theorem 1, and Theorem 2 of the present paper are stated and proved in the bigraphical setting there as [16, Theorem 5.26, Theorem 5.29, and Theorem 5.31]. The above-mentioned bijection theorem is [16, Theorem 5.21].

Altogether, by lifting in this manner the result of the preceding sections, we fully answer the two questions posed in the introduction: we have sufficient conditions for a sorting to preserve congruence properties and enjoy semantic correspondence, and we have a construction, the closure sorting, for constructing sortings satisfying these conditions.

We proceed to demonstrate the viability of the closure sortings by recovering Milner's local bigraphs as a particular closure sorting.

7 Local Bigraphs and Closure Sortings

In the setting of bigraphical reactive systems we have binding bigraphs [4] as a natural extension of pure bigraphs [1, 4]; moreover, we have local bigraphs [10, 3] as a natural extension of binding bigraphs. It has been suspected [21] that local bigraphs represent the end of this evolutionary ladder. In this section, we clarify what this evolution is and demonstrate that the closure sorting for a predicate derived from the scope rule is its natural endpoint. In the process, we prove that local bigraphs can, in a sense to be made precise, be replaced by the closure sorting for this predicate.

We do not reiterate the definition of pure and local bigraphs here. Refer to one of [1, 4] for the definition of pure bigraphs; to one of [18, 11] for intuition and examples of pure bigraphs; to [4] for the definition of binding bigraphs; and to [10, 3] for the definition of local bigraphs.

Binding bigraphs partition the ports of a signature into the binding ports and the free ports. The intuition of binding ports is that [4, p.68]:

> "all points linked to a binding port of a node u lie inside u."

Although Milner and Jensen use the word "points" and later clarify that these may be names as well as ports, binding bigraphs are surely but a means of re-

stricting pure bigraphs to those that only peer a binding port with ports beneath it. This condition, called the scope condition, is easily seen to be decomposable.

Definition 13 (Scope predicate). *Let Σ be a binding signature. The* Scope predicate for Σ *is a predicate $\mathbf{P}_\Sigma$ on the morphisms of $\mathbf{B}(U(\Sigma))$, that is, on the pure bigraphs over $U(\Sigma)$. For a bigraph f of $\mathbf{B}(U(\Sigma))$, we define $\mathbf{P}_\Sigma(f)$ iff whenever a binding port p on a node n is in a link, then any other port p' in that link is on a node that has n as an ancestor.*

Binding ports are intended to model binders in term languages, e.g., the binder k in the input prefix $x(k).P$ of the π-calculus [4, 6]. (The introduction of binding ports also enables a more expressive definition of parametric reaction rules. We shall not discuss this application here.)

The progression from binding to local bigraphs is one including more and more ways to decompose bigraphs satisfying the scope predicate P_Σ given above. The closure sorting for P_Σ is an endpoint of this progression, as it contains by definition every possible such decomposition. We discuss this progression in some detail because so reveals, we believe, the essence of binding and local bigraph. The discussion will be somewhat technical, however, so feel free to skip ahead to the paragraph titled "Replacability".

To make sure that the scope condition is preserved by composition, binding bigraphs augment objects with locations of names. Each name x in an object (m, X) is either global or located at a specific place $i \in m$. A binding bigraph is then permitted to link only appropriately located ports or inner names to located outer names or binding ports. (When we say that something is "linked to a binding port" we actually mean peered with that binding port. Because no link can contain two binding ports, it is sound to identify an edge with a binding port linked to that edge.)

Ascribing only a single place to each located name does not account for all possible decompositions of a bigraph satisfying the scope condition. For instance, the following two pure bigraphs both satisfy the scope condition, as does their composition. However, we can assign no location to x that will make f a valid binding bigraph.

$$f : (2, \{x\}) \to (1, \emptyset) = /x.\mathsf{k}_{(x)}(-_0 \mid -_1)$$
$$g : (0, \emptyset) \to (2, \{x\}) = \mathsf{h}_x \mid \mathsf{h}_x$$

To remedy this deficiency, [3] introduces local bigraphs. These assign to each name x of an interface (m, X) a subset $m' \subset m$ of places, instead of just a single place $i \in m$. The global names of binding bigraphs correspond to everywhere located names.

Local bigraphs still do not capture every possible decomposition of pure bigraphs satisfying the scope condition. To wit, consider the following two bigraphs.

$$f : \ (1, \{x, y\}) \to (1, \emptyset) = \ /x, y.-_0 \tag{12}$$
$$g : \ (1, \emptyset) \to (1, \{x, y\}) = \ \mathsf{k}_{(x)}(\mathsf{h}_y) \tag{13}$$

The interfaces of local bigraphs are simply not capable of expressing that in g, the name x can be free, as long as any context can only link it to names within the scope of the control $\mathsf{k}_{(x)}$, such as the name y in this example.

The closure sorting for the scope condition includes exactly this kind of additional interfaces, and thus allows this decomposition. For the application of local bigraphs in [10], encoding of the lambda calculus as a bigraphical reactive system, this extra flexibility is not exploited: For any term or reaction rule containing binders, the bound names will not appear in the interface. However, as we will make precise below, the extra flexibility is not harmful.

Replacability The closure sorting for the scope predicate is also a viable substitute for local bigraphs in the following precise sense. There exists a full embedding $\iota : \mathbf{B}(\Sigma) \to \mathfrak{C}(\mathbf{P}_\Sigma)$ of local bigraphs into the closure sorting for $\mathbf{P}_\Sigma$. This embedding witnesses $\mathbf{B}(\Sigma)$ being a sub-sorting of $\mathfrak{C}(\mathbf{P}_\Sigma)$ in the sense that it makes the following diagram commute.

$$\begin{array}{ccc} \mathbf{B}(\Sigma) & \xrightarrow{\ \iota\ } & \mathfrak{C}(\mathbf{P}_\Sigma) \\ & \searrow \quad \swarrow & \\ & \mathbf{B}(U(\Sigma)) & \end{array} \tag{14}$$

Moreover, this embedding both preserves and reflects bisimilarity for any reactive system $\mathcal{R}$ on $\mathbf{B}(\Sigma)$ and its image $\mathcal{R}$ in $\mathfrak{C}(\mathbf{P}_\Sigma)$. Indeed, ι preserves and reflects transitions, and the morphisms in the image of ι has no transitions outside of that image. In this sense, the closure sorting $\mathfrak{C}(\mathbf{P}_\Sigma)$ is a reasonable substitute for local bigraphs.

Theorem 3. *Let Σ be a binding signature, let $\mathbf{P}_\Sigma$ be the Scope predicate on $\mathbf{B}(\Sigma)$, and let $\mathcal{R}$ be a reactive system on $\mathfrak{C}(\mathbf{P}_\Sigma)$. Suppose f is an agent of $\mathcal{R}$ in $\mathbf{B}(\Sigma)$. There is a transition $\iota(f) \xrightarrow{g'} h'$ in $\mathfrak{C}(\mathbf{P}_\Sigma)$ if and only if there is a transition $f \xrightarrow{g} h$ in $\mathbf{B}(\Sigma)$ and both $\iota(g) = g'$ and $\iota(h) = h'$. It follows that for agents f, g of $\mathbf{B}(\Sigma)$ we have $f \sim g$ if and only if $\iota(f) \sim \iota(g)$.*

It follows that local bigraphs sustain the behavioural theory of pure bigraphs. Milner proved as much by hand [10, 3]; now, we get the same result for free.

Corollary 1. *Let Σ be a binding signature. For any supported reactive system on $\mathbf{B}(\Sigma)$, bisimilarity on the supported transitions is a congruence.*

However, we also *expand* on Milner's results, because Theorem 1 gives us that local bigraphs have semantic correspondence up to P_Σ.

Corollary 2. *Let Σ be a binding signature. Then the sorting $\mathbf{B}(\Sigma) \to \mathbf{B}(U(\Sigma))$ respects supported $\mathbf{P}_\Sigma$-reactions and -transitions.*

The proof of Theorem 3 hinges on the following characterisation of $\mathfrak{C}(\mathbf{P}_\Sigma)$, which is interesting in its own right in so far as it tells us exactly what is missing from the interfaces of local bigraphs to allow all decompositions.

Lemma 2. *Let Σ be a binding signature, let $\mathfrak{C}(\mathbf{P}_\Sigma)$ be the closure sorting for the scope predicate $\mathbf{P}_\Sigma$, and let (m, X) be an object of $\mathbf{B}(\Sigma)$. Let $\Gamma(m, X) = \mathcal{P}(\{0, \ldots, m-1\}) + \mathcal{P}(X)$ be the set comprising subsets of places and subsets of names of the object (m, X). We call maps $\rho : X \to \Gamma(m, X)$ s.t. $\rho(x) \in \mathcal{P}(X)$ implies $x \in \rho(x)$* generating maps *for (m, X).*

The fibre of $\mathfrak{C}(\mathbf{P}_\Sigma) \to \mathbf{B}(U(\Sigma))$ over $(0, X)$ is isomorphic to the partial order that has only one object. The fibre of $\mathfrak{C}(\mathbf{P}_\Sigma)$ over (m, X) for $m > 0$ is isomorphic to the partial order over generating maps for (m, X), ordered pointwise by $\rho(x) \sqsubseteq \varrho(x)$ if and only if either (a) $\rho(x) \subseteq \varrho(x) \subseteq \mathcal{P}(\{0, \ldots, m-1)$, (b) $\varrho(x) \subseteq \rho(x) \subseteq \mathcal{P}(X)$, or (c) $\rho(x) \subseteq \mathcal{P}(\{0, \ldots, m-1\})$ and $\varrho(x) \subseteq \mathcal{P}(X)$.

The generating maps $\rho : X \to \Gamma(m, X)$ cover exactly the decomposition in (13), with the intuition that if $\rho(x) \in \mathcal{P}(X)$ then x is a name that occurs in a binder that may be safely linked to names in $\rho(x)$.

We recover the interfaces of local bigraphs as the generating maps $\rho : X \to \mathcal{P}(\{0, \ldots, m-1\})$ that take every name to a left inject, that is, that assigns locations to names.

We have now shown how local bigraphs arise as a special case closure sorting. In [16] it was shown how Milner's homomorphic sorting [1] also arise as a special case on the closure sorting.

8 Conclusion and Future Work

First, we have given a sufficient condition for a sorting to reflect reactive and transition semantics of well-sorted terms. Second, we have extended the theory of sortings for reactive systems with a new construction of sortings for decomposable predicates, the closure sorting. Third, we have sketched a generalisation of the theory of sortings for reactive systems to the setting of supported precategories. Finally, we proved that local bigraphs arise naturally as a sub-sorting of the closure sorting obtained from the scope condition. Besides alleviating the need for redeveloping the behavioural theory for local bigraphs, it supports local bigraphs as the natural extension of bigraphs with local names. We conjecture that the sortings [1–6, 9–11] can all be obtained as closure sortings. (Of the sortings mentioned in the introduction, this list leaves out only the edge-sortings of [7], which does not appear to approximate predicates.)

We see two main avenues of future work. One is to investigate sortings in other frameworks, in particular within graph rewriting [22, 23] and the 2-categorical approach to reactive systems [20, 19]. Another is to investigate the algebraic properties of sortings and if the closure sorting is somehow universal among sortings that capture a decomposable predicate and respect the behavioural theory.

References

1. Milner, R.: Pure bigraphs: Structure and dynamics. Information and Computation **204**(1) (2006) 60–122

2. Milner, R., Leifer, J.J.: Transition systems, link graphs and Petri nets. Technical Report 598, U. of Cambridge Computer Laboratory (2004)
3. Milner, R.: Bigraphs whose names have multiple locality. Technical Report 603, U. of Cambridge Computer Laboratory (2004)
4. Jensen, O.H., Milner, R.: Bigraphs and mobile processes (revised). Technical Report 580, U. of Cambridge Computer Laboratory (2004)
5. Milner, R.: Bigraphs for petri nets. In: Lectures on Concurrency and Petri Nets. Volume 3098 of LNCS. (2003) 686–701
6. Jensen, O.H.: Mobile Processes in Bigraphs. PhD thesis, U. of Aalborg (2008) Forthcoming.
7. Bundgaard, M., Sassone, V.: Typed polyadic pi-calculus in bigraphs. In: PPDP '06. (2006) 1–12
8. Grohmann, D., Miculan, M.: Reactive systems over directed bigraphs. In: CONCUR '07. Volume 4703 of LNCS. (2007) 380–394
9. Bundgaard, M., Hildebrandt, T.: Bigraphical semantics of higher-order mobile embedded resources with local names. In: GT-VC '05. Volume 154 of ENTCS. (2006) 7–29
10. Milner, R.: Local bigraphs and confluence: Two conjectures. In: EXPRESS '06. (2006) 42–50
11. Birkedal, L., Debois, S., Elsborg, E., Hildebrandt, T., Niss, H.: Bigraphical Models of Context-aware Systems. In: FOSSACS '06. Volume 3921 of LNCS. (2006)
12. Leifer, J.J., Milner, R.: Deriving bisimulation congruences for reactive systems. In: CONCUR '00. (2000) 243–258
13. Leifer, J.J.: Operational Congruences for Reactive Systems. PhD thesis, U. of Cambridge Computer Laboratory and Trinity College (2001)
14. Sewell, P.: From rewrite rules to bisimulation congruences. Theoretical Computer Science **274**(1–2) (2002) 183–230
15. Birkedal, L., Debois, S., Hildebrandt, T.: Sortings for reactive systems. In: CONCUR '06. Volume 4137 of LNCS., Springer (2006) 248–262
16. Debois, S.: Sortings and Bigraphs. PhD thesis, IT University of Copenhagen (2008) `http://www.itu.dk/people/debois/pubs/thesis.pdf`.
17. Leifer, J.J.: Synthesising labelled transitions and operational congruences in reactive systems, part 2. Technical Report RR-4395, INRIA (2002)
18. Conforti, G., Macedonio, D., Sassone, V.: Spatial logics for bigraphs. In: ICALP '05. Volume 3580 of LNCS. (2005) 766–778
19. Sassone, V., Sobocinski, P.: Deriving bisimulation congruences: 2-categories vs. precategories. In: FOSSACS '03. Volume 2620 of LNCS. (2003) 409–424
20. Sassone, V., Sobocinski, P.: Reactive systems over cospans. In: LICS '05, IEEE (2005) 311–320
21. Milner, R. Personal communication (2006)
22. Bonchi, F., Gadducci, F., König, B.: Process bisimulation via a graphical encoding. In: ICGT '06. Volume 4178 of LNCS. (2006) 168–183
23. Ehrig, H., König, B.: Deriving bisimulation congruences in the DPO approach to graph rewriting with borrowed contexts. Mathematical Structures in Computer Science **16**(6) (2006) 1133–1163