Reactive systems over cospans

Vladimiro Sassone
University of Sussex

Paweł Sobociński
Università di Pisa*

Abstract

The theory of reactive systems, introduced by Leifer and Milner and previously extended by the authors, allows the derivation of well-behaved labelled transition systems (LTS) for semantic models with an underlying reduction semantics. The derivation procedure requires the presence of certain colimits (or, more usually and generally, bicolimits) which need to be constructed separately within each model. In this paper, we offer a general construction of such bicolimits in a class of bicategories of cospans. The construction sheds light on as well as extends Ehrig and König's rewriting via borrowed contexts and opens the way to a unified treatment of several applications.

Introduction

Some of the most important operational techniques, developed through research on concurrency, involve the use of labelled transition systems (LTS) and the accompanying notions of operational preorders and equivalences, bisimulation being chief among these, in order to reason about process equivalence. Some basic concepts, common to many applications where such operational techniques apply, include the notions of *term*, *context* and *reaction*.

Such notions began to be studied in the seminal work of Berry and Boudol [3] on the *chemical abstract machine*. More recently, Leifer and Milner [12] introduced the framework of *reactive systems*, aimed at providing a general setting where such notions can be studied. Reactive systems can be seen as a generalisation of ground term-rewriting systems, where a collection of ground rewrite rules is closed under a set of "reactive" contexts to obtain the rewrite relation. Contexts are, themselves, organised as the arrows of a category $\mathbb{C}$.

Leifer and Milner's chief contribution was to utilise a universal categorical construction, dubbed *relative pushout* (RPO), in order to equip each reactive system with an LTS whose labels can be characterised as the smallest contexts that allow reactions to occur – an idea originally due to Sewell [19]. Such LTSs are very well-behaved; in particular, bisimilarity is a congruence with respect to all contexts, provided that $\mathbb{C}$ has "enough" RPOs. In [16], the authors proposed an approach based on a 2-dimensional generalisation of RPOs, the *groupoidal relative pushout* (GRPO), which has been shown in [15] to encompass previous approaches addressing these issues. Of course, one retains the congruence theorems (bisimilarity, trace and failures equivalences) in the more general setting.

Several constructions of GRPOs have been proposed in the literature for *particular* categories of models. For instance, Milner explicitly constructed such bicolimits for bigraphs [13] in full detail,[1] but there has been little work on understanding how such constructions may be performed in other models. Indeed, the main applications of the theory of reactive systems have, thus far, been relatively simple [16, 15].

In the paper we address this by giving a construction of GRPOs for a wide range of models – those which can be expressed as certain cospans over *adhesive* categories. The class of adhesive categories, introduced in [10], includes many categories where the objects are variants of graphs – for instance, the category **Graph** of graphs and graph homomorphisms is adhesive.

An important application of our results is Ehrig and König's *rewriting via borrowed contexts*, an extension of the double-pushout (DPO) graph transformation approach with a contextual semantics that allows transfer of concepts and techniques from the field of process algebra to graphs. The labels of their LTS are the so-called "borrowed-contexts" – certain contexts which allow a DPO transformation to be performed.

Other applications include a partial treatment of Milner's bigraphs [13], a model introduced to model various fun-

*Research partly supported by EU FET-GC 'MyThS: Models and Types for Security in Mobile Distributed Systems', IST-2001-32617 and 'Agile: Architectures for Mobility', IST-2001-32747; EC RTN 2-2001-00346 SegraVis; and BRICS, Basic Research in Computer Science (www.brics.dk), funded by the Danish National Research Foundation, University of Aarhus.

[1] Milner actually constructs ordinary RPOs in a *precategory* – this is equivalent to constructing GRPOs in the bicategory which results from taking support translations as 2-cells (see [15]).

damental notions arising from the study of process calculi such as the pi-calculus, as well as a Petri net-based model of open systems [17], related to open nets [1].

Before a formal presentation of the technical details of our results, we shall first allow ourselves a certain amount of informality in order to illustrate the intuition and the motivation behind the construction introduced in this paper. A cospan in an adhesive category is for us a generalised "graphical context" between "interfaces;" starting with objects I, J and C of an adhesive category $\mathbb{A}$ (the reader could take $\mathbb{A}$ = **Graph** for concreteness), we can treat C as a context with inner interface I and outer interface J by providing homomorphisms $I \xrightarrow{\iota_C} C \xleftarrow{o_C} J$. This resulting cospan is treated as an *arrow* of a new category Cospan($\mathbb{A}$) with the same objects as $\mathbb{A}$ – indeed such cospans may be composed; intuitively, by "glueing together along the common interface;" concretely, by constructing a pushout. Due to the nature of pushouts, the composition is, in general, not associative on the nose, but rather, up to a unique coherent isomorphism – which means that Cospan($\mathbb{A}$) is, in general, a *bicategory* [2].

Starting with any category $\mathbb{A}$ with pushouts, the process $\mathbb{A} \rightsquigarrow$ Cospan($\mathbb{A}$) can be thought of as a kind of a canonical "contextualisation" operation, turning objects of a one category into the arrows of another (bi)category; the reader should keep in mind that this is an informal description, since a cospan consists of not just an object of $\mathbb{A}$, but also of the two maps to it, and these are of vital importance.

The work of Gadducci et al. (see for instance [6, 7]) on cospans and their relationship with various aspects of the theory of graph transformation systems is closely related to our approach; the idea of using cospans as a notion of generalised contexts is implicit in their work. Cospans as well as spans (the dual notion) over graphs have also been studied by Katis et al. [9] – roughly, they argue that spans over graphs can be seen as generalised automata.

For an idealised example of cospans, consider the simple model of a coffee vending machine, illustrated by the leftmost diagram of Fig. 1. It has an outer interface consisting of two nodes, \$ and C, which one can think of as a money slot and the coffee out-tray. These are the parts of the coffee machine accessible to the environment, the internal components, represented by S, are invisible. The middle diagram represents a coffee drinker. He expects to see a money slot and a coffee out-tray, which are his inner interfaces. As the outer interface of the coffee machine and the inner interface of the coffee drinker match, one may compose them and obtain the system pictured in the rightmost diagram.

There are several examples of cospans relevant for computer science. For instance bigraphs are cospans in an adhesive category of place-link graphs with outer interface maps mono, and graph contexts in the sense of Ehrig and König [5] are cospans in **Graph** with both interface maps mono. Examples of *reactive systems* over such cospan categories include, therefore, the theory of *bigraphical reactive systems* of Milner [13] and the theory of *double-pushout* (DPO) graph transformation [4].

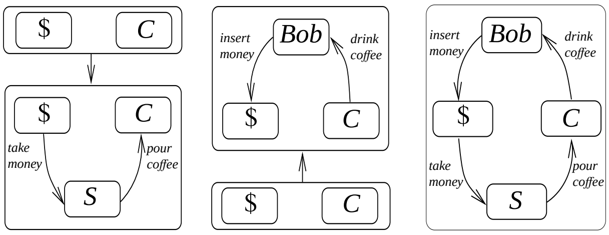


Figure 1. Example of a contextual system.

The main result of the paper is the construction of GRPOs in a class of cospan bicategories over adhesive categories, which in turn allows the derivation of LTSs for *all* reactive systems based on such bicategories. Specifically, we require that the *inner* interface map of every cospan is *mono*. Although technical in nature, the linearity condition does have an intuitive account. As alluded to in the coffee drinker example, one can consider a cospan as a "black box," with an inner interface and an outer interface. The environment cannot see the internals of the system and only interacts with it through the outer interface. The fact that the outer interface need not be linear means that the system is free to connect the outer interface arbitrarily to its internal representation. For example, the coffee machine could have two extra buttons in its outer interface; the "café latte" button and the "cappuccino" button. The machine internals could connect both these buttons to the same internal trigger for coffee with milk; the point is that the system controls its outer interface and is able to equate parts of it. On the other hand, the system cannot control what is plugged into one of its slots. Thus, an assumption of input-linearity is essentially saying that the system does not have the right to assume that two components coming in through the inner interface can be confused.

Applying the construction to mono cospans in **Graph** yields Ehrig and König's [5] construction of rewriting via borrowed contexts – which confirms that their approach is an instance of the general methodology of reactive systems; we shall show that their labels actually arise from GRPOs and thus satisfy a universal property. As a consequence, their congruence theorem follows automatically from the theory of reactive systems. Finally, because we require only the left leg of the cospan to be mono, we are able to generalise their results.

Since Milner's bigraphs can be seen as output-linear cospans, the construction does not apply. However, we provide LTS semantics for a different class of bigraphs: the input-linear cospans over the same category. The construction has also been applied in order to derive LTS semantics for a compositional model of Petri nets [17].

Structure of the paper. The paper begins with two sections summarising the background theory necessary for our construction. First, in §1 we recall Leifer and Milner's theory of reactive systems and relative pushouts together with the generalisation of the theory previously worked out by the authors. Secondly, in §2 we recall about adhesive categories and cospan bicategories. Our main construction appears in §3. In §4, we provide a detailed exposition of the relevance to Ehrig and König's [5] and mention other applications.

Preliminaries. Throughout the paper we assume a basic knowledge of category theory, as well as basic acquaintance with the concepts of 2-categories and bicategories. Here we mention only that a *bicategory* [2] can be described, roughly, as a 2-category where associativity and identity laws of horizontal composition hold up to isomorphisms. We shall denote all associativity isomorphisms by a, as for example, $a : h(gf) \Rightarrow (hg)f$. The isomorphisms are required to respect well-known coherence axioms. We use id_X to denote the identity at X, while 1_f and $\bullet$ stand, respectively, for identity and vertical composition of 2-cells.

1 Reactive Systems and GRPOs

Here we briefly recall an extension of Leifer and Milner's notion of reactive system to categories with 2-dimensional structure, as previously introduced by the authors [16]. The basic idea of a reactive system is to bring together a collection of contexts, which form the arrows of a category, with a set of reaction rules on terms. From the reaction rules one generates a reaction relation by contextual closure. Using certain pushouts, one equips a reactive system with an LTS on which bisimilarity is a congruence with respect to all contexts.

Apart from trivial examples, it is the norm that the contexts of a reactive system have some notion of isomorphism – often referred to as structural congruence in the field of process calculi. It turns out that ignoring such structure is often problematic (cf. [16]) and one solution is to include the isomorphism/structural congruence as part of the structure; i.e., to consider *2-categories* of contexts instead of ordinary categories. Guided by examples, we usually consider only invertible 2-cells. In order to generate an LTS and extend the congruence results still, one then needs to consider bipushouts (rather than pushouts) [16].

In this paper we consider cospans as contexts; since the composition is associative up to isomorphism, we shall need to work in bicategories instead of 2-categories. Apart from changing the definition of reactive system, this generalisation poses no difficulty – bicolimits are naturally a bicategorical notion.

Definition 1.1 (Reactive System). A *reactive system* **C** consists of

1. a bicategory $\mathbb{B}$ with all 2-cells invertible;
2. a collection $\mathbb{D}$ of arrows of $\mathbb{B}$ called the *reactive contexts*; it is required to be closed w.r.t. 2-isomorphisms and be composition-reflecting (see below);
3. a distinguished object $0 \in \mathbb{B}$; arrows with domain 0 are often referred to as *terms*;
4. a set of pairs of terms, the *reaction rules* $\mathcal{R}$. The terms l, r of any given pair $\langle l, r\rangle \in \mathcal{R}$ have the same codomain as arrows of $\mathbb{B}$.

The reactive contexts are those inside which evaluation may occur. Composition-reflecting means that $dd' \in \mathbb{D}$ implies d and $d' \in \mathbb{D}$, while the closure property means that given $d \in \mathbb{D}$ and an isomorphism $\rho\colon d \Rightarrow d'$ in $\mathbb{B}$ implies $d' \in \mathbb{D}$.

The reaction relation $\longrightarrow\!\!\!\!\triangleright$ is defined by taking $a \longrightarrow\!\!\!\!\triangleright\, a'$ if there is $\langle l, r\rangle \in \mathcal{R}$, $d \in \mathbb{D}$, $\alpha\colon dl \Rightarrow a$ and $\alpha'\colon a' \Rightarrow dr$. This represents that, up to structural congruence α, a is the left-hand side l of a reduction rule in a reactive context d, while a' is, up to structural congruence α', the right-hand side r in the same context.

Leifer and Milner [12] developed the derivation of a canonical LTS associated to any given reactive system. The derivation uses a universal construction, dubbed relative-pushout (RPO), which is a pushout in a slice category. Bisimulation on the resulting LTS is a congruence, provided that the underlying category of the reactive system has enough RPOs; we shall make the meaning of "enough" precise below.

A groupoidal-relative pushout (GRPO) is the generalisation of an RPO to categories with 2-dimensional structure. We recall the definition below,[2] and due to lack of space refer the reader to [16] for a full presentation. Note that although GRPOs are introduced there in the setting of G-categories (2-categories with invertible 2-cells), the development is easily transferred to bicategories with invertible 2-cells – it suffices to introduce the coherent associativity isomorphisms where necessary. In order to increase readability, we leave these out within the definition below.

Definition 1.2 (GRPO). Within the scope of this definition we shall refer to the diagrams of Fig. 2. A *candidate* for square (*i*) is a tuple $\langle I_5, e, f, g, \beta, \gamma, \delta\rangle$ such that $\delta b \bullet g\beta \bullet \gamma a = \alpha$. In other words, the 2-cells γ, β and δ, illustrated in diagram (*ii*), paste together to give α.

A *groupoidal-relative-pushout* (GRPO) is a candidate which satisfies a universal property, namely, for any other

[2] For category theorists, a GRPO can be described concisely as a bi-pushout in a pseudo-slice category.

$$\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, "c"] \arrow[rr, phantom, "\overset{\alpha}{\Longrightarrow}"] & & I_3 \arrow[ul, "d"'] \\
 & I_1 \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, bend left, "c"{name=C}] \arrow[r, "e"] & I_5 \arrow[u, "g"] \arrow[Rightarrow, to=D, "\delta"] & I_3 \arrow[ul, bend right, "d"'{name=D}] \arrow[l, "f"'] \\
 & I_1 \arrow[ul, "a"{name=A}] \arrow[ur, "b"'{name=B}] &
\arrow[Rightarrow, from=C, to=2-2, "\gamma"]
\arrow[Rightarrow, from=A, to=B, "\beta"']
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I_6 & \\
I_2 \arrow[ur, bend left, "e'"{name=E}] \arrow[r, "e"] & I_5 \arrow[u, "h"] \arrow[Rightarrow, to=F, "\psi"] & I_3 \arrow[ul, bend right, "f'"'{name=F}] \arrow[l, "f"'] \\
\arrow[Rightarrow, from=E, to=2-2, "\varphi"]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I_4 \\
I_6 \arrow[ur, bend left, "g'"{name=G1}] & I_5 \arrow[l, "h"] \arrow[u, "g"'{name=G2}] \\
\arrow[Rightarrow, from=G1, to=G2, "\tau"]
\end{tikzcd}$$

(i) (ii) (iii) (iv)

Figure 2. Components of a GRPO.

$$\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, "f"] \arrow[rr, phantom, "\overset{\alpha}{\Longrightarrow}"] & & I_3 \arrow[ul, "d"'] \\
 & 0 \arrow[ul, "a"] \arrow[ur, "l"'] &
\end{tikzcd}$$

Figure 3. Redex square.

candidate $\langle I_6, e', f', g', \beta', \gamma', \delta' \rangle$ there exists a *mediating morphism*, that is a quadruple

$$\langle h : I_5 \to I_6, \varphi : e' \Rightarrow he, \psi : hf \Rightarrow f', \tau : g'h' \Rightarrow g \rangle$$

illustrated in diagrams (iii) and (iv). The equations that need to be satisfied are: 1. $\tau e \bullet g'\varphi \bullet \gamma' = \gamma$; 2. $\delta' \bullet g'\psi \bullet \tau^{-1} f = \delta$; and 3. $\psi b \bullet h\beta \bullet \varphi a = \beta'$.

Such a mediating morphism must be *essentially unique*, namely, for any other mediating morphism $\langle h', \varphi', \psi', \tau' \rangle$ there must exist a unique 2-cell ξ: $h \Rightarrow h'$ which makes the two mediating morphisms compatible, i.e.: 1. $\xi e \bullet \varphi = \varphi'$; 2. $\psi \bullet \xi^{-1} f = \psi'$; and 3. $\tau' \bullet g'\xi = \tau$.

The reader should note that, in general, the GRPO of diagram (i) depends not only on a and b, but also on c, d and α; that is, given arrows a and b, there may be many different ways of obtaining a "smallest context"; for a simple example, consider the two GRPOs illustrated in Fig. 7 and discussed on page 6.

GRPOs are used to classify the smallest contexts necessary for a reaction to be performed. The redex squares which satisfy this property are known as groupoidal-idem-pushouts, or GIPOs.

Definition 1.3 (GIPO). Diagram (i) of Fig. 2 is a G-idem-pushout (GIPO) if $\left\langle I_4, c, d, \mathrm{id}_{I_4}, \alpha, 1_c, 1_d \right\rangle$ is its GRPO.

We are now in a position to define the LTS which has such smallest contexts as labels.

Definition 1.4 (LTS). For $\mathbf{C}$ a reactive system and $\mathbb{B}$ its underlying bicategory, define GLTS($\mathbf{C}$) as follows:

- the states of GLTS($\mathbf{C}$) are terms;
- there is a transition $a \xrightarrow{f} a'$ if there exists a 2-cell α, a rule $\langle l, r \rangle \in \mathcal{R}$, and a reactive context $d \in \mathbb{D}$ such that the diagram in Fig. 3 is a GIPO and a' is isomorphic to dr.

$$\begin{tikzcd}
 & D & \\
A \arrow[ur, "g"] & & B \arrow[ul, "n"'] \\
 & C \arrow[ul, "m"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Figure 4. Pushout diagram.

In previous work [16, 15], we have considered an *abstract* LTS where the terms and transitions are quotiented by isomorphism; this is the norm in specific calculi, for instance in CCS or the pi-calculus the states are processes, i.e., terms modulo structural congruence. For simplicity, we leave the terms and transitions unquotiented here, that is, we deal with the *concrete* transition system, using the terminology of [20]. Both (concrete and abstract) versions have the same congruence properties.

Definition 1.5. A reactive system $\mathbf{C}$ is said to have *redex-GRPOs* when every redex square (Fig. 3), with a and f arbitrary, d a reactive context and l a part of a reaction rule $\langle l, r \rangle \in \mathcal{R}$, has a GRPO.

The following is a slight extension of a theorem for 2-categories which can be found in [16].

Theorem 1.6. *Let* $\mathbf{C}$ *be a reactive system with redex-GRPOs. Then bisimilarity on* GLTS($\mathbf{C}$) *is a congruence with respect to all the contexts of* $\mathbf{C}$.

2 Adhesive categories and cospans

Adhesive categories. In order to construct GRPOs in cospan bicategories we shall need the notion of adhesive categories, we refer the reader to [10] for an introduction; here we shall only mention some properties of adhesivity which shall be used in the later exposition. In §3, we assume that the underlying category of the cospan bicategory is adhesive and use the structure of adhesive categories repeatedly in the construction of GRPOs.

Adhesive categories can be described by a slogan: pushouts along monomorphisms exist and are well-behaved. The class of adhesive categories includes **Set**, **Graph**, as well as any elementary topos. It is closed under product, slice, coslice and functor category. As explained in [10], many graphical structures relevant to computer science form adhesive categories.

We shall need the following properties of adhesive categories for our constructions. The proof of the following lemma can be found in [10].

Lemma 2.1. *Let* $\mathbb{A}$ *be an adhesive category.*

1. *Monos are stable under pushout in $\mathbb{A}$. In other words, in the diagram of Fig. 4, if m is mono then n is mono.*
2. *A pushout in $\mathbb{A}$ (Fig. 4) is also a pullback diagram, if m is mono.*
3. *If it exists, a pushout complement*[3] *of $\langle m, g\rangle$, with m mono, is unique up to a compatible isomorphism; more precisely, if f, n and $f' : C \to B', n' : B' \to D$ are pushout complements, then there exists an isomorphism $\varphi : B \to B'$ such that $\varphi f = f'$ and $n'\varphi = n$.*

Recall that, given an object X, a subobject $[\mu : Y \to X]$ is an equivalence class of monomorphisms into X, where the equivalence relation is generated from the canonical preorder on monomorphisms into X: $\mu \leq \mu'$ if there exists $k : Y \to Y'$ such that $\mu' k = \mu$. Subobject *intersection* and *union* refers to a meet and join of subobjects (if it exists) in this preorder. Any category with pullbacks has binary subobject intersection. Binary subobject unions in adhesive categories are calculated by pushing out along their intersection.

Cospan bicategories. We shall assume that $\mathbb{C}$ is a category with *chosen pushouts*. That is, for arrows $m : C \to A$ and $f : C \to B$, there exists unique "chosen" object $A +_C B$ and arrows $A \to A +_C B$ and $B \to A +_C B$ such that the resulting square is a pushout.

The bicategory of cospans Cospan($\mathbb{C}$) has the same objects as $\mathbb{C}$, but arrows from I_1 to I_2 are cospans. $I_1 \xrightarrow{\iota_C} C \xleftarrow{o_C} I_2$. We will denote such cospans $C^{o_C}_{\iota_C} : I_1 \to I_2$, and omit ι_C (resp. o_C) when I_1 (resp. I_2) is initial. We shall sometimes refer to I_1 and I_2 as, respectively, the inner and outer interfaces of $C^{o_C}_{\iota_C}$. Intuitively, we can think of a cospan as a generalised context, where C is the internals, (the image via o_C of) I_2 represents the public view of C, and (the image via ι_C of) I_1 the view of C afforded to the "holes" in it.

A 2-cell $h\colon C^{o_C}_{\iota_C} \Rightarrow C'^{o_{C'}}_{\iota_{C'}} : I_1 \to I_2$ in Cospan($\mathbb{C}$) is an arrow $h\colon C \to C'$ in $\mathbb{C}$ satisfying $h\iota_C = \iota_{C'}$ and $ho_C = o_{C'}$. The invertible 2-cells provide a canonical notion of "structural congruence." We shall denote the bicategory of cospans which has the 2-cells limited to isomorphisms by Cospan$^{\cong}$($\mathbb{C}$). Given cospans $C^{o_C}_{\iota_C} : I_1 \to I_2$ and $D^{o_{C'}}_{\iota_{C'}} : I_2 \to I_3$, their composition is the cospan obtained by taking the pushout of o_C and ι_D, as illustrated by Fig. 5. Note that in the resulting composition, I_2 is "forgotten." Composition is associative up to a unique isomorphism. It is easy to check that the associativity isomorphisms satisfy the coherence axioms, and thus yield a bicategory.

To perform our construction we need a certain linearity restriction. In particular, the notion of input-linear cospan.

$$\begin{tikzcd}
 & & C +_{I_2} D & & \\
 & C \arrow[ur] & & D \arrow[ul] & \\
I_1 \arrow[ur, "\iota_C"] & & I_2 \arrow[ul, "o_C"'] \arrow[ur, "\iota_D"'] & & I_3 \arrow[ul, "o_D"']
\end{tikzcd}$$

Figure 5. Composition of cospans.

Definition 2.2 (Linearity). A cospan $C^{o_C}_{\iota_C}$ is said to be *input-linear* when ι_C is a mono, *output-linear* when o_C is mono, and *linear* when it is both input-linear and output-linear.

A simple corollary of the first part of Lemma 2.1 is that the composition of two input-linear cospans over an adhesive category yields an input-linear cospan. Similarly, composition preserves output-linearity and linearity.

Definition 2.3. Assuming that $\mathbb{A}$ is adhesive, let ILC($\mathbb{A}$) be the bicategory consisting of input-linear cospans and invertible 2-cells. Similarly, let LC($\mathbb{A}$) be the bicategory of linear cospans and invertible 2-cells.

3 GRPOs for cospans

In this section we present a general construction of GRPOs for a class of reactive systems over cospan bicategories.

Construction. The following result is the main original contribution of this paper. Several implications of Theorem 3.1 and of the construction which constitutes its proof are considered in §4.

Theorem 3.1. *If $\mathbb{A}$ is adhesive,* ILC($\mathbb{A}$) *has GRPOs.*

The remainder of this section presents a proof of the above theorem. We shall first outline an algorithm for the construction of the desired minimal candidate. Because of space constraints, we have omitted most technical details; these can be found in the second author's Ph.D. thesis [20] and in [18].

Throughout this section, we shall refer to the diagrams within Fig. 6. An arbitrary square in ILC($\mathbb{A}$), as illustrated in diagram (*i*), amounts to a commutative diagram (*ii*) in $\mathbb{A}$, with α an isomorphism.

Definition 3.2 (Frame of reference). Given a diagram (*i*) in ILC(**C**), by a *frame of reference* we mean an arbitrary object X equipped with isomorphisms $\omega_l : A +_{I_2} C \to X$ and $\omega_r : B +_{I_3} D \to X$ such that $\omega_r^{-1}\omega_l = \alpha$. Let $\omega_A = \omega_l i_{A \to A+C} : A \to X$, $\omega_C = \omega_l i_{C \to A+C} : C \to X$, $\omega_B = \omega_r i_{B \to B+D} : B \to X$ and $\omega_D = \omega_r i_{D \to B+D} : D \to X$.

Clearly, a frame of reference for (*i*) always exists. For instance, one can let X be $B +_{I_3} D$, ω_r be the identity and

[3] Given $m : C \to A$ and $g : A \to D$, we say that B is a *pushout complement* of $\langle m, g\rangle$ when there exist $f : C \to B$ and $n : B \to D$ such that the resulting diagram (Fig. 4) is a pushout diagram.

$$\begin{tikzcd}
& I_4 & \\
I_2 \arrow[ur, "C"] \arrow[rr, Rightarrow, "\alpha"] & & I_3 \arrow[ul, "D"'] \\
& I_1 \arrow[ul, "A"] \arrow[ur, "B"'] &
\end{tikzcd}$$

(i)

$$\begin{tikzcd}
& C \arrow[d, "i"] & I_4 \arrow[l, "o_C"'] \arrow[r, "o_D"] & D \arrow[d, "i"'] & \\
I_2 \arrow[ur, tail, "\iota_C"] \arrow[dr, "o_A"'] & A+_{I_2}C \arrow[rr, "\alpha"] & & B+_{I_3}D & I_3 \arrow[ul, tail, "\iota_D"'] \arrow[dl, "o_B"] \\
& A \arrow[u, "i"] & I_1 \arrow[l, tail, "\iota_A"] \arrow[r, tail, "\iota_B"'] & B \arrow[u, "i"'] &
\end{tikzcd}$$

(ii)

$$\begin{tikzcd}
& C \arrow[dr, "\omega_C"'] & I_4 \arrow[l, "o_C"'] \arrow[r, "o_D"] & D \arrow[dl, "\omega_D"] & \\
I_2 \arrow[ur, "\iota_C"] \arrow[dr, "o_A"'] & & X & & I_3 \arrow[ul, "\iota_D"'] \arrow[dl, "o_B"] \\
& A \arrow[ur, "\omega_A"] & I_1 \arrow[l, "\iota_A"] \arrow[r, "\iota_B"'] & B \arrow[ul, "\omega_B"'] &
\end{tikzcd}$$

(iii)

$$\begin{tikzcd}
& X & \\
A \arrow[ur, bend left, "\omega_A"] \arrow[r, "\epsilon_1"] & Y \arrow[u, tail, "\mu"] & B \arrow[ul, bend right, "\omega_B"'] \arrow[l, "\epsilon_2"'] \\
& I_1 \arrow[ul, "\iota_A"] \arrow[ur, "\iota_B"'] &
\end{tikzcd}$$

(iv)

$$\begin{tikzcd}
& & I_5 \arrow[dll, "o_E"'] \arrow[dr, "\iota_G"] \arrow[ddd, "o'_F"] & \\
E \arrow[dr, "\rho_1"] \arrow[ddd, "\sigma_1"'] & & & G \arrow[dll, "\pi_1"'] \arrow[ddd, "\pi_2"] \\
& C \arrow[ddd, "\omega_C"] & & \\
& & F \arrow[dll, "\sigma_2"'] \arrow[dr, "\rho_2"] & \\
Y \arrow[dr, "\mu"'] & & & D \arrow[dll, "\omega_D"] \\
& X & &
\end{tikzcd}$$

(v)

$$\begin{tikzcd}
& I_4 & \\
I_2 \arrow[ur, bend left, "C"] \arrow[r, "E"] \arrow[rr, Rightarrow, bend right=20, "\beta"'] \arrow[ur, Rightarrow, phantom, "\gamma"] & I_5 \arrow[u, "G"] & I_3 \arrow[ul, bend right, "D"'] \arrow[l, "F"'] \arrow[ul, Rightarrow, phantom, "\delta"'] \\
& I_1 \arrow[ul, "A"] \arrow[ur, "B"'] &
\end{tikzcd}$$

(vi)

Figure 6. Construction of GRPOs

ω_I be α. It follows easily that diagram (iii) is commutative. Notice that both of the squares in diagram (iii) are actually pushouts.

Since ω_A and ω_B are readily seen to be mono, being pushouts of monos in an adhesive category (Lemma 2.1), we are able to obtain the subobject union – an object $Y = A \cup B$ and monomorphisms $\mu : Y \rightarrow X$, $\epsilon_1 : A \rightarrow Y$ and $\epsilon_2 : B \rightarrow Y$ satisfying $\mu\epsilon_1 = \omega_A$ and $\mu\epsilon_2 = \omega_B$.

Notice that since μ is mono and $\mu\epsilon_1\iota_A = \omega_A\iota_A = \omega_B\iota_B = \mu\epsilon_2\iota_B$ we have that $\epsilon_1\iota_A = \epsilon_2\iota_B$. We obtain the commutative diagram (iv).

Algorithm 3.3 (GRPO Construction in ILC($\mathbb{A}$)**).** The construction of the components of the minimal candidate is outlined below. They are illustrated in diagrams (v) and (vi). We obtain:

1. G as the pullback of $\omega_C : C \rightarrow X$ and $\omega_D : D \rightarrow X$;
2. E as the pullback of $\mu : Y \rightarrow X$ and $\omega_C : C \rightarrow X$;
3. F as the pullback of $\mu : Y \rightarrow X$ and $\omega_D : D \rightarrow X$;
4. I_5 as the pullback of $\rho_2 \colon F \rightarrow D$ and $\pi_2 \colon G \rightarrow D$; notice that due to the properties of pullbacks, we obtain a morphism $o_E : I_5 \rightarrow E$ such that all the faces of (v) are pullbacks.

$$\begin{tikzcd}
& 0 & \\
1 \arrow[ur, "2"] \arrow[rr, Rightarrow, "1"] & & 1 \arrow[ul, "2"'] \\
& 0 \arrow[ul, "1"] \arrow[ur, "1"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& 0 & \\
1 \arrow[ur, "2"] \arrow[rr, Rightarrow, "\gamma"] & & 1 \arrow[ul, "2"'] \\
& 0 \arrow[ul, "1"] \arrow[ur, "1"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
& 0 & \\
1 \arrow[ur, bend left, "2"] \arrow[r, "\mathrm{id}"] \arrow[rr, Rightarrow, bend right=20, "1"'] \arrow[ur, Rightarrow, phantom, "1"] & 1 \arrow[u, "2"] & 1 \arrow[ul, bend right, "2"'] \arrow[l, "\mathrm{id}"'] \arrow[ul, Rightarrow, phantom, "1"'] \\
& 0 \arrow[ul, "1"] \arrow[ur, "1"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& 0 & \\
1 \arrow[ur, bend left, "2"] \arrow[r, "2"] \arrow[rr, Rightarrow, bend right=20, "\gamma"'] \arrow[ur, Rightarrow, phantom, "1"] & 2 \arrow[u, "2"] & 1 \arrow[ul, bend right, "2"'] \arrow[l, "2'"'] \arrow[ul, Rightarrow, phantom, "1"'] \\
& 0 \arrow[ul, "1"] \arrow[ur, "1"'] &
\end{tikzcd}$$

Figure 7. GRPOs in ILC(**Set**)**.**

The following two lemmas show that the construction of Algorithm 3.3 indeed results in a GRPO, and complete the proof of Theorem 3.1.

Lemma 3.4. The construction as outlined in Algorithm 3.3 provides a candidate for square (i).

Proof. In order to prove the lemma, we are required to:

1. demonstrate the existence of appropriate inner and outer interfaces for each of E, F and G;
2. show that there exist isomorphisms $\beta : A +_{I_2} E \rightarrow B +_{I_3} F$, $\gamma : C \rightarrow E +_{I_5} G$ and $\delta : F +_{I_5} G \rightarrow D$;
3. show that these isomorphisms paste together in way which results in a 2-cell equal to $\alpha : A +_{I_2} C \rightarrow B +_{I_3} D$.

We omit the details, and refer the reader to [18, 20]. □

Lemma 3.5. The candidate constructed using Algorithm 3.3 satisfies the universal property of a GRPO.

Proof. Omitted, see [18, 20]. □

Examples. First, with the aid of a simple example, we shall demonstrate that it is *critical* that one fixes a particular isomorphism $\alpha : A +_{I_2} C \rightarrow B +_{I_3} D$. Indeed, consider the two diagrams in ILC(**Set**) in the first row of Fig. 7; the arrow $0 \xrightarrow{1} 1$ denotes the cospan $0 \xrightarrow{!} 1 \xleftarrow{\mathrm{id}} 1$, while $1 \xrightarrow{2} 0$ denotes the cospan $1 \xrightarrow{0} 2 \xleftarrow{!} 0$, and $\gamma : 2 \rightarrow 2$ is the bijection which swaps the two elements. We illustrate the two resulting candidates in the second row of Fig. 7: $1 \xrightarrow{\mathrm{id}} 1$ is the identity cospan $1 \xrightarrow{\mathrm{id}} 1 \xleftarrow{\mathrm{id}} 1$, whereas $1 \xrightarrow{2} 2$ is the cospan $1 \xrightarrow{0} 2 \xleftarrow{\mathrm{id}} 2$ and $1 \xrightarrow{2'} 2$ is the cospan $1 \xrightarrow{0} 2 \xleftarrow{\gamma} 2$.

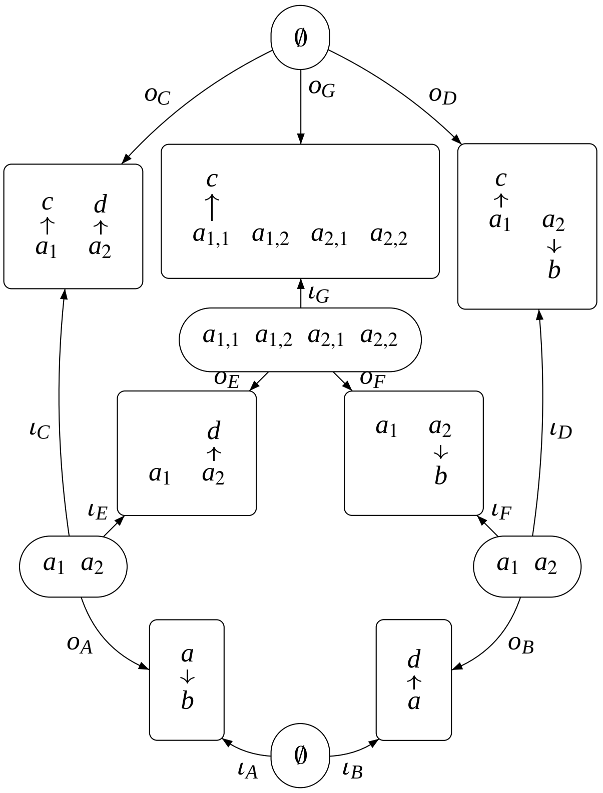
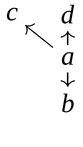


Figure 8. GRPO in ILC(**Graph**)

Intuitively, in the first case, the two lower arrows correspond to the same element. Thus the context contains redundant information which can be factored out resulting in a candidate which is just the identity context. Conversely, they are different in the second case, meaning that the context has to be kept in the minimal candidate.

We shall give one more example with ILC(**Graph**) as our category of contexts in order to illustrate some of the effects of allowing non-mono outer interfaces. In the accompanying Fig. 8, we label the nodes of the graphs in order to clarify the action of various graph morphisms, which we leave unlabelled. We also do not draw the 2-cells as the labelling on the nodes makes these clear. The exterior maps of Fig. 8 correspond to two different ways of decomposing the graph illustrated below.



The GRPO is illustrated in the interior of the diagram. One can compare this "blowing up" of the outer interface in the GRPO with the phenomenon of "forking" as described by Leifer [11, Fig. 1.4].

$$\begin{tikzcd}
L & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed, "g"] & R \arrow[d, "h"] \\
C \arrow[u, leftarrow, "f"'] & E \arrow[l, "v"] \arrow[r, "w"'] & D
\end{tikzcd}$$

Figure 9. Rewrite rule application (DPO).

4 Applications

In this sections we shall examine some of the applications of Theorem 3.1 and Algorithm 3.3. Due to space restrictions, we shall concentrate on the application Ehrig and König's rewriting via borrowed contexts and briefly summarise applications to bigraphs and Petri nets.

4.1 Rewriting via borrowed contexts

Our results both shed light on and extend Ehrig and König's rewriting via borrowed contexts [5]. First, we shall show that borrowed contexts correspond to GIPOs, and therefore satisfy a universal property. Consequently, the theory of rewriting via borrowed contexts falls within the framework of reactive systems [12, 16, 15] and therefore the various congruence properties and constructions carry over. In particular, Ehrig and König's congruence theorem can be seen as an application of the congruence of bisimilarity theorem for reactive systems. Finally, due to the generality of Theorem 3.1, we relax some of the technical conditions imposed by Ehrig and König and thereby introduce the notion of extended borrowed contexts (Definition 4.7).

We begin with a brief account of a variant of double-pushout (DPO) graph transformation [4], working at the level of an arbitrary adhesive category $\mathbb{A}$.

Definition 4.1 (Rewrite Rule). A rewrite rule p is a span

$$L \xleftarrow{l} K \xrightarrow{r} R \tag{1}$$

in $\mathbb{A}$. No assumptions are made about either l or r.

Here L and R represent respectively the left and right-hand side of the rule, while K is information which remains unaffected by the rewrite. A redex in an object C is identified by matching a rule's left-hand side, which is done via a morphism $f \colon L \to C$.

Definition 4.2 (Grammar). A grammar G is a pair $\langle \mathbb{C}, \mathbf{P} \rangle$ where $\mathbb{C}$ is a category and $\mathbf{P}$ is a set of rewrite rules. A grammar is said to be *adhesive* when $\mathbb{C}$ is adhesive.

Definition 4.3 (Rewrite rule application). Object C rewrites to D with rule p, in symbols $C \longrightarrow_{p,f} D$, if there exist an object E and morphisms so that the two squares in Fig. 9 are pushouts. We shall write $C \longrightarrow D$ if there exist $p \in P$ and $f \colon L \to C$ such that $C \longrightarrow_{p,f} D$.

Using the conclusion of the third part of Lemma 2.1, one can show that in adhesive grammars, the application of left-linear rewrite rules is functional, in the sense that if $C \mathrel{-\!\!\blacktriangleright}_{p,f} D$ and $C \mathrel{-\!\!\blacktriangleright}_{p,f} D'$, then $D \cong D'$ (cf. [10]).

Any grammar $G = \langle \mathbb{C}, \mathbf{P} \rangle$ can be seen translated into a reactive system on Cospan($\mathbb{C}$). Let 0 denote the empty graph, and the set $\mathcal{R}$ contain for each rewrite rule p (1) a pair

$$\left\langle 0 \to L \xleftarrow{l} K, 0 \to R \xleftarrow{r} K \right\rangle.$$

We choose all arrows of Cospan($\mathbb{C}$) to be reactive. Let $\mathrel{-\!\!\triangleright}$ denote the resulting reaction relation. The translation preserves semantics, as demonstrated by the following lemma, which is similar to a previously published result [6] and can be considered folklore. We use the shorthand $C \mathrel{-\!\!\triangleright} D$ to mean that C and D are cospans with empty inner and outer interfaces. It is crucial for us because it serves as a foundation for relating the theory of DPO transformation systems and reactive systems over cospans.

Lemma 4.4. *$C \mathrel{-\!\!\blacktriangleright} D$ iff $C \mathrel{-\!\!\triangleright} D$.*

In order to apply Theorem 3.1, we need restrict to graph transformation systems which correspond to reactive systems over input-linear cospans.

Definition 4.5 (Input-Linear Rewrite Application). Object C rewrites to D input-linearly with rule p, in symbols $C \mathrel{\succ\!\!-\!\!\blacktriangleright}_{p,f} D$, if $C \mathrel{-\!\!\blacktriangleright}_{p,f} D$ and in addition f, g and h of Fig. 9 are mono.

It follows that starting with an adhesive grammar $G = \langle \mathbb{A}, \mathbf{P} \rangle$ and an input-linear rewrite relation $\mathrel{\succ\!\!-\!\!\blacktriangleright}$, we may construct a corresponding reactive system $\mathbf{A}$ over the bicategory of input-linear cospans ILC($\mathbb{A}$). Clearly, Lemma 4.4 specialises, and thus semantics is preserved. Now, using the construction of GRPOs of Theorem 3.1, we obtain an LTS; let LTS(G) = GLTS($\mathbf{A}$). Using the congruence theorem for bisimilarity from [16] on such an LTS, we obtain that bisimilarity on LTS(G) is a congruence.

A rewrite rule (1) is called *linear* when both l and r are mono. If we restrict our view to DPO rewriting systems with linear rewrite rules and input-linear rewrites then we are in a position to compare the resulting LTS with rewriting via borrowed contexts, which requires these extra assumptions.

Precisely, given an adhesive grammar $G = \langle \mathbb{A}, \mathbf{P} \rangle$, where $\mathbf{P}$ consists of linear rewrite rules, let RBC(G) be the LTS derived via rewriting with borrowed contexts as defined in [5]. Recall that the states of RBC(G) are graphs with a mono outer interface. The transitions are labelled by mono cospans of graphs; there is a transition $G^{o_G} \xrightarrow{F^{o_F}_{\iota_F}} H^{\theta_2 o_C}$ precisely when there exists a diagram as illustrated in Fig. 10 with all arrows mono, $L \xleftarrow{l} I \xrightarrow{r} R \in \mathbf{P}$, regions (1–4) pushouts and region (5) a pullback.

$$\begin{tikzcd}
D \arrow[r, tail] \arrow[d, tail] \arrow[dr, phantom, "(1)"] & L \arrow[d, tail, "\epsilon_2"] \arrow[dr, phantom, "(2)"] & I \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, tail, "\iota_C"] \arrow[dr, phantom, "(3)"] & R \arrow[d, tail, "\theta_1"] \\
G \arrow[r, tail, "\epsilon_1"] & G^+ & C \arrow[l, "\sigma_2"'] \arrow[r, "\theta_2"] & H \\
J \arrow[u, "o_G"] \arrow[r, tail, "\iota_F"'] \arrow[ur, phantom, "(4)"] & F \arrow[u, "\sigma_1"] \arrow[ur, phantom, "(5)"] & K \arrow[l, "o_F"] \arrow[u, "o_C"'] &
\end{tikzcd}$$

Figure 10. Borrowed context.

Theorem 4.6. LTS(G) = RBC(G).

Proof. The proof consists of showing that borrowed contexts correspond to contexts obtained from GIPOs. Throughout the proof, we shall refer to the diagrams within Fig. 11.

From borrowed contexts to GIPOs. Given a borrowed context diagram as illustrated in Fig. 10, we shall show that one may construct a GIPO, as illustrated in diagram (i).

Assume without loss of generality that $L +_I C = G^+$. Let $\alpha : F +_J G \to G^+$ be the unique isomorphism such that $\alpha i_1 = \sigma_1 : F \to G^+$ and $\alpha i_2 = \epsilon_1 : G \to G^+$.

Since square (1) of Fig. 10 is a pushout of monos and we are in an adhesive category, it is also a pullback (Lemma 2.1). This implies that $L \cup G = G^+$. The cube which results from an application of Algorithm 3.3 is illustrated in (ii), in the construction we use only the fact that square (5) of Fig. 10 is a pullback. Thus, the GRPO of redex square (i) is $\langle K, F, C, \mathrm{id}, \mathrm{id}, \mathrm{id}, \alpha \rangle$, meaning that it is a GIPO.

From GIPOs to borrowed contexts. Suppose that $G^{o_G} \xrightarrow{F^{o_F}_{\iota_F}} H^{o_H}$ is a transition in LTS(G). then $F^{o_F}_{\iota_F}$ must be a part of a GIPO diagram. Since every GIPO can be constructed as a GRPO, we have a redex diagram as the outside of diagram (iii), with $\langle L^{o_L}, R^{o_R} \rangle$ being a reaction rule, corresponding via the translation of Lemma 4.4 to the rewrite rule $L \xleftarrow{o_L} I_3 \xrightarrow{o_R} R$. The candidate $\langle I_5, F, C, G, \beta, \gamma, \delta \rangle$ illustrated is the GRPO obtained via the construction of Algorithm 3.3.

We also have $H^{o_H} \cong C^{o_C}_{i_C} \circ R^{o_R}$, which in other words means that diagram (iv) is a pushout with $o_H : I_5 \to H$ equal to $\theta_2 \circ o_C$.

Recall that from the construction, we have that diagram (v) is a pushout, diagram (vi) is a pullback, diagrams (vii) and (v) are pushouts. Diagrams (iv) through to ($viii$) paste together to give the required borrowed context. □

We end this section with an extension of borrowed contexts suggested by the linearity conditions imposed in the construction of Algorithm 3.3.

$$\begin{tikzcd}
 & K & \\
J \arrow[ur, "F"] \arrow[rr, Rightarrow, "\alpha"] & & I \arrow[ul, "C"'] \\
 & 0 \arrow[ul, "G"] \arrow[ur, "L"'] &
\end{tikzcd}$$

(*i*)

$$\begin{tikzcd}
 & & K \arrow[dll, "o_F"'] \arrow[ddd, "o_C"] \arrow[dr] & \\
F \arrow[dr] \arrow[ddd, "\sigma_1"'] & & & K \arrow[dll, "o_F"] \arrow[ddd, "o_C"] \\
 & F \arrow[ddd, "\sigma_1"] & & \\
 & & C \arrow[dll, "\sigma_2"] \arrow[dr] & \\
G^+ \arrow[dr] & & & C \arrow[dll, "\sigma_2"] \\
 & G^+ & &
\end{tikzcd}$$

(*ii*)

$$\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, bend left, "F'"] \arrow[r, dashed, "F"] & I_5 \arrow[u, dashed, "G"] & I_3 \arrow[ul, bend right, "C'"'] \arrow[l, dashed, "C"'] \\
 & 0 \arrow[ul, "G"] \arrow[ur, "L"'] &
\end{tikzcd}$$

γ δ β

(*iii*)

$$\begin{tikzcd}
I_3 \arrow[r, "o_R"] \arrow[d, tail, "i_C"'] & R \arrow[d, tail, "\theta_1"] \\
C \arrow[r, "\theta_2"'] & H
\end{tikzcd}$$

(*iv*)

$$\begin{tikzcd}
G \cap L \arrow[r, tail] \arrow[d, tail] & G \arrow[d, tail, "\epsilon_1"] \\
L \arrow[r, tail, "\epsilon_2"'] & Y
\end{tikzcd}
\quad
\begin{tikzcd}
I_5 \arrow[r, "o_F"] \arrow[d, "o_C"'] & F \arrow[d, "\sigma_1"] \\
C \arrow[r, "\sigma_2"'] & Y
\end{tikzcd}
\quad
\begin{tikzcd}
I_2 \arrow[r, tail, "i_F"] \arrow[d, "o_G"'] & F \arrow[d, "\sigma_1"] \\
G \arrow[r, tail, "\epsilon_1"'] & Y
\end{tikzcd}
\quad
\begin{tikzcd}
I_3 \arrow[r, tail, "i_C"] \arrow[d, "o_L"'] & C \arrow[d, "\sigma_2"] \\
L \arrow[r, tail, "\epsilon_2"'] & Y
\end{tikzcd}$$

(*v*) (*vi*) (*vii*) (*viii*)

Figure 11. GIPOs and Borrowed Contexts

Definition 4.7 (Extended Borrowed Contexts). Given an adhesive grammar $G = \langle \mathbf{C}, \mathbf{P} \rangle$ with an *arbitrary* set of rules $\mathbf{P}$, we shall construct a labelled transition system with:

- States: Graphs with outer interfaces, $J \xrightarrow{o_G} G$ where o_G is arbitrary;
- Transitions: Cospans of graphs $J \xrightarrow{\iota_F} F \xleftarrow{o_F} K$ where ι_F is mono and o_F is arbitrary.

We derive a transition $G^{o_G} \xrightarrow{\overset{o_F}{F_{\iota_F}}} H^{o_H}$ if there exists a commutative diagram as in Fig. 10, where (*1*), (*2*), (*3*) and (*4*) are pushouts, while square (*5*) is a pullback. The indicated morphisms are assumed to be mono, the others are arbitrary.

As a corollary of the translation between borrowed contexts and GIPOs of Theorem 4.6 and the congruence we have that bisimilarity on the labelled transition system resulting from Definition 4.7 is a congruence with respect to arbitrary input-linear graph contexts.

4.2 Bigraphs and Petri nets

Bigraphs are a graphical formalism originally introduced by Milner [13] in order to model dynamic systems with independent locality and connectivity structures [8]. They can be seen as 'output-linear' cospans over a certain adhesive category, as we shall explain below.

It follows, that by considering an input-linear variant – a model which still incorporates many of bigraphs' fundamental features – we meet the requirements of our framework. Using Algorithm 3.3, we are then able to derive an LTS for any reactive system on such bigraphical structures. Such an LTS coincides with Milner's on the bigraphs in the intersection of the respective definitions. Here we offer only a brief discussion, the reader is referred to [18, 20] for details. We introduce the adhesive category of place-link graphs, which can be considered as "bigraphs without interfaces." Interfaces will be handled by considering the bicategory of cospans over this category.

For an alphabet of controls Σ, we define a *place-graph* G to be a directed graph with nodes labelled over Σ. Here edges represent (a generalisation of) the nesting of locations. The connectivity of each $\mathsf{K} \in \Sigma$ is determined by an arity function $ar \colon \Sigma \to \mathbb{N}$, so that K has ports numbered $1, \ldots, ar(\mathsf{K})$. By "*set of ports*" of G we indicate the disjoint union of all the ports associated to all the nodes of G.

Definition 4.8 (Place-Link Graphs). A place-link (pl) graph is a place-graph G together with a *link* map $l \colon P \to S$, where P is the set of ports of G and S is a finite set of "connections". The role of l and S is to jointly describe the connections between ports; two ports $p, p' \in P$ are said to be *connected* iff $l(p) = l(p')$.

A place-link (pl) morphism is a graph morphism together with a function between the sets of connections so that the port connections are preserved in the obvious way. Let $\mathbf{PLGraph}_\Sigma$ be the category of pl-graphs and pl-graph morphisms over Σ.

It is easy to construct a category $\mathbb{X}_\Sigma$ so that $\mathbf{PLGraph}_\Sigma$ is a presheaf category $\mathbf{Set}^{\mathbb{X}_\Sigma}$ and, as such, is adhesive. We can then conclude as follows.

Corollary 4.9. ILC($\mathbf{PLGraph}_\Sigma$) *has GRPOs, calculated using Algorithm 3.3.*

There are two aspects of our bicategory of pl-graphs that generalise the theory of bigraphs. Firstly, interfaces here need not be "discrete," i.e., specify just nodes to be merged and ports to be connected, but allow for more complex forms of composition. Secondly, place graphs are usually forests of trees, and their input (resp. output) interfaces reach only leaves (resp. roots), while we allow all directed graphs. One can, however, prove that Corollary 4.9 remains valid when the arrows of ILC($\mathbf{PLGraph}_\Sigma$) are restricted to such bigraphs.

The main difference with bigraphs remains the input- vs output-linearity issue, which has the interesting effect of banning name aliasing in respectively the inner and the outer interface. We leave a deeper analysis of this dichotomy as future work.

Petri nets provide another application of our framework, described in [17]. The model consists of adding an elementary notion of interface to Petri nets, similarly to other proposals in the literature, as e.g. [1, 14]. Specifically, we equip each net with two ordered subsets of places: the *inner* and the *outer interface*. This, together with morphisms that respect net topologies and interfaces, yields a bicategory **INet**, which can easily be recast in terms of cospans over an adhesive category. Formally, let **MNet** be the category of marked Petri nets and structure-preserving homomorphisms. As **MNet** is adhesive, we have the following result, where GRPOs are calculated via Algorithm 3.3.

Corollary 4.10. ILC(**MNet**) *has GRPOs,*

INet consists of the linear cospans with discrete interfaces in ILC(**MNet**) and, once again, the existence of GRPOs is not affected by this restriction. Then, considering the usual Petri "token game" as a reactive system over **INet**, we automatically derive an LTS for Petri nets with interfaces over which bisimulation is congruence.

As it turns out, such a bisimulation admits a very simple characterisation consisting of only three kinds of transitions: $p \xrightarrow{\tau} p'$, if p' results from p through an internal firing; $p \xrightarrow{+i} p'$ if p' results from p by adding a token at its ith outer place; and $p \xrightarrow{-i} p'$ if p' results from p by removing a token from its ith outer place. We remark that this is the same bisimulation as derived by Milner using a bigraph-based representation of Petri nets with interfaces [14].

5 Conclusion

We have constructed groupoidal relative pushouts (GRPOs) in a general framework of generalised contexts and interfaces, represented by cospan bicategories over adhesive categories. This allows us to *systematically* derive a compositional semantics for *each* reactive system in the framework, in the form of a LTS determining a bisimulation congruence. We have shown in detail how Ehrig and König's rewriting via borrowed contexts [5] falls within the framework and can be obtained as an application of our theory, and mentioned two others, a variant of Milner's bigraphs and a compositional approach to Petri nets [17].

As future work, we plan to investigate conditions under which a general construction of GRPOs exists for output-linear cospan bicategories.

Acknowledgements. The authors would like to thank Vincent Danos and the anonymous referees for their helpful comments.

References

[1] P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional modeling of reactive systems using open nets. In *CONCUR '01*, volume 2154 of *LNCS*. Springer, 2001.

[2] J. Bénabou. Introduction to bicategories. In *Midwest Category Seminar*, volume 42 of *LNM*, pages 1–77. Springer, 1967.

[3] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.

[4] H. Ehrig, G. Engels, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors. *Graph Grammars and Computing by Graph Transformation, Vol. 1-3.* World Scientific, 1999.

[5] H. Ehrig and B. König. Deriving bisimulation congruences in the DPO approach to graph rewriting. In *FOSSACS '04*, volume 2987 of *LNCS*, pages 151–166. Springer, 2004.

[6] F. Gadducci and R. Heckel. An inductive view of graph transformation. In *WADT '97*, pages 223–237, 1997.

[7] F. Gadducci, R. Heckel, and M. Llabrés. A bi-categorical axiomatisation of concurrent graph rewriting. In *CTCS '99*, volume 29 of *ENTCS*. Elsevier, 1999.

[8] O. H. Jensen and R. Milner. Bigraphs and mobile processes. Technical Report 570, University of Cambridge, 2003.

[9] P. Katis, N. Sabadini, and R. F. C. Walters. Bicategories of processes. *J. Pure Appl. Algebra*, 115:141–178, 1997.

[10] S. Lack and P. Sobociński. Adhesive categories. In *FOSSACS '04*, volume 2987 of *LNCS*. Springer, 2004.

[11] J. Leifer. *Operational congruences for reactive systems*. Phd thesis, University of Cambridge, 2001.

[12] J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *CONCUR '00*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000.

[13] R. Milner. Bigraphical reactive systems. In *Concur '01*, volume 2154 of *LNCS*, pages 16–35. Springer, 2001.

[14] R. Milner. Bigraphs for Petri nets. In *Lectures on Concurrency and Petri Nets 2003*, volume 3098 of *LNCS*, pages 686–701. Springer, 2004.

[15] V. Sassone and P. Sobociński. Deriving bisimulation congruences: 2-categories vs precategories. In *FOSSACS '03*, volume 2620 of *LNCS*. Springer, 2003. Full version to appear in *Theor. Comp. Sci.*

[16] V. Sassone and P. Sobociński. Deriving bisimulation congruences using 2-categories. *Nord. J. Comput.*, 10(2):163–183, 2003.

[17] V. Sassone and P. Sobociński. A congruence for Petri nets. In *Workshop on Petri Nets and Graph Transformation*, 2004.

[18] V. Sassone and P. Sobociński. Congruences for contextual graph-rewriting. Technical Report RS-14, BRICS, 2004.

[19] P. Sewell. From rewrite rules to bisimulation congruences. *Theoretical Computer Science*, 274(1–2):183–230, 2002.

[20] P. Sobociński. *Deriving process congruences from reaction rules.* PhD thesis, BRICS, University of Aarhus, 2004.