BRICS

Basic Research in Computer Science

Congruences for Contextual Graph-Rewriting

Vladimiro Sassone
Paweł Sobociński

BRICS Report Series RS-04-11
ISSN 0909-0878 June 2004

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/04/11/

Congruences for Contextual Graph-Rewriting

Vladimiro Sassone Paweł Sobociński

Abstract

We introduce a comprehensive operational semantic theory of graph-rewriting. Graph-rewriting here is meant in a broad sense as we aim to cover and extend previous work based both on Milner's bigraphs and Ehrig and König's rewriting via borrowed contexts. The central idea is recasting rewriting frameworks as Leifer and Milner's reactive systems. Consequently, graph-rewriting systems are associated with canonical labelled transition systems, on which bisimulation equivalence is a congruence with respect to arbitrary graph contexts (cospans of graphs). The central technical contribution of the paper is the construction of groupoidal relative pushouts, introduced and developed by the authors in recent work, in input-linear cospan (bi)categories over arbitrary adhesive categories.

Contents

Introduction

Operational techniques, including coinductive arguments, which originated from research on the semantics of concurrency, have recently begun to be applied in other areas of (computer) science (cf. e.g. [1, 5]). The main facet of these approaches is the use of labelled transition systems (lts) and the accompanying notions of operational preorders and equivalences, bisimulation being chief among these.

Leifer and Milner's seminal [13] introduced reactive systems and initiated the investigation of their semantics. Reactive systems are 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 organised as the arrows of a category $\mathbb{C}$. Using a universal categorical construction, the *relative pushout* (RPO), each reactive system can be equipped with an lts. The labels of the lts are the 'smallest' contexts which allow reactions to occur – an idea due to Sewell [18]. Such ltss are very well-behaved; in particular, bisimulation is a congruence with respect to all contexts, provided that $\mathbb{C}$ has enough RPOs.

When applied naively, RPOs have proven inadequate in some reactive systems where contexts have non-trivial algebraic structure. In some cases they do not give the expected labels in the lts (cf. [17]), while in others, they do not exist (cf. [16]). The troublesome contexts often exhibit non-trivial automorphisms, which naturally form a part of a 2-dimensional structure on the underlying category $\mathbb{C}$. It is important to notice that such situations are the norm, rather than the exception. Context isomorphisms arise naturally already in simple process calculi, where terms are up to structural congruence. In [17], the authors proposed an enhanced approach based on a 2-dimensional generalisation of RPOs, the *groupoidal relative pushout* (GRPO), which has been shown in [16] to encompass previous approaches addressing these issues.

Several constructions of RPOs have been proposed in the literature for particular categories of models. For example, Leifer [12] constructs RPOs in a category of action graphs, while Jensen and Milner do so in the category of bigraphs [9]. A construction of (G)RPOs in a general setting has so far been missing. In this paper, we construct GRPOs in a general framework of abstract, uninterpreted contexts. Given a category of interest $\mathbb{C}$, we consider a "category of contexts" where the objects of $\mathbb{C}$ can be composed with each other through interfaces: the cospan bicategory on $\mathbb{C}$. Such bicategories have the same objects as $\mathbb{C}$, but the arrows are cospans

$$I \xrightarrow{\iota} C \xleftarrow{o} J ,$$

which can be viewed as an object C enriched with an "input" interface ι and and "output" interface o. Roughly, ι is the partial view of C attainable from its "holes," while o is the restricted view of C afforded to the "environment." Composition of cospans is performed by pushing out interfaces, which can be understood as "glueing together" an agent and its context along their common interface. Due to the nature of pushouts, composition is only associative up to a unique isomorphism.

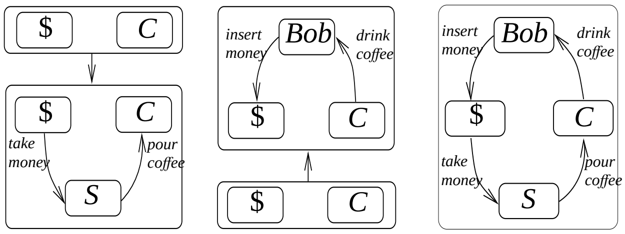


Figure 1: Example of a contextual system

As an example of these concepts, consider the simple model of a coffee vending machine, illustrated by the leftmost diagram of Figure 1. It has an output 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 input interfaces. As the output interface of the coffee machine and the input interface of the coffee drinker match, one may compose them and obtain the system pictured in the rightmost diagram. (Input and output interfaces of the vending machine and coffee drinker have been omitted.)

The main result of the paper is the construction of GRPOs in a class of cospan bicategories, which in turn allows the derivation of ltss for all reactive systems over such bicategories. Specifically, we require a linearity condition on the input interfaces, namely, that ι is mono. Additionally, our cospans are over adhesive categories [11], which are categories in which pushouts along monomorphisms exist and are suitably well-behaved. As we prove in the paper, adhesive categories have enough structure for the construction of GRPOs in our cospan bicategories.

Although technical in nature, the linearity condition does have an intuitive account. As alluded in the coffee drinker example, one can consider a cospan as a "black box," with an input interface and an output interface. The environment cannot see the internals of the system and only interacts with it through the output interface. The fact that the output interface need not be linear means that the system is free to connect the output interface arbitrarily to its internal representation. For example, the coffee machine could have two extra buttons in its output 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 output interface and is able to equate parts of it. On the other hand, the system cannot control what is plugged into one of its holes. 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 input interface are equal.

In order to prove the relevance and usefulness of the construction, we treat two large examples. Firstly, we apply it to derive lts for *double-pushout* (DPO) *graph-rewriting* systems. Graph rewriting is a well-established field of theoretical Computer Science [2], concerned with the extension of rewriting techniques from terms to graph structures. DPO graph rewriting can be generalised nicely to rewriting in arbitrary adhesive categories [11].

As DPO graph-rewriting systems can be seen as reactive systems on the bicategory Cospan(**Graph**), the bicategory of cospans over the (adhesive) category of graphs, we can derive ltss for graph rewriting directly and systematically. This equips any arbitrary graph rewriting system with a contextual semantics and a corresponding coinduction principle, so as to allow for the transfer of concepts and techniques from the field of process algebra to graph-rewriting. In other words, this yields a behavioural equivalence based uniquely on the interactions of (concurrent) dynamic systems with their environment, while the presence of a well-behaved lts allows the use of bisimulation to prove contextual equivalence.

When restricting cospans to purely linear (mono) maps, the lts we derive agrees, almost on-the-nose, with Ehrig and König's recently proposed approach, the so-called rewriting with borrowed contexts [5]. Consequently, Ehrig and König's congruence theorem can be understood as a corollary of the congruence theorem for GRPOs [17]. Without the restriction, the application of reactive systems to graph rewriting extends the borrowed-context approach by considering graph contexts where the output interface need not be injective. In this application, therefore, the paper contributes in two ways. Firstly, it is an extension of the results of Ehrig and König; secondly, it provides a missing link between their work and the work of Leifer and Milner [13].

Our second application is the construction of GRPOs for a version of Milner's *bigraphs* [9]. Bigraphs have been recently proposed as a formalism to model mobility of communication channels, or links (as in the π calculus), together with spatial mobility of agents, or places (as in distributed calculi). We introduce the adhesive category of place-link graphs. The cospan bicategories over place-link graphs resemble Milner's bigraphs, with some differences imposed by the respective linearity conditions. The general construction of GRPOs provides reactive systems over our bigraphs with a labelled transition semantics.

The advantages of a general approach to GRPOs based on abstract "categories of contexts" include, therefore, insights into how these are constructed and apply across a wide range of models. Moreover, given a reactive system within the class treated in this paper, the GRPO construction provides not only a canonical congruent process equivalence (bisimulation on the resulting lts), but also a proof method: the lts itself.

Structure of the paper. In the first of the three preliminary sections, section 1.1, we recall the recently introduced notion of adhesive category due to Lack and the second author [11]. Secondly, in section 1.2, we recall the notions of 2-categories and cospan bicategories. Finally, section 1.3 recalls the definition of a reactive system, which we generalise slightly so that we are able to consider a bicategory as the underlying category of a reactive system. This section relies heavily on technology previously introduced by the authors in [17] and [16]. The main result of the paper, the construction of GRPOs for a class of reactive systems over cospan bicategories, is stated and proved in section 2; sections 3.1 and 3.2 illustrate two applications, respectively the derivation of ltss for DPO graph rewriting and the construction of GRPOs for a variant of bigraphs.

1 Preliminaries

1.1 Adhesive categories

In order to construct GRPOs in cospan bicategories we shall need the notion of adhesive categories [11], which we recall below. Adhesive categories have a slogan: *pushouts along monomorphisms exist and are well-behaved*. We shall assume that the underlying category of the cospan bicategory is adhesive and use the structure of adhesive categories repeatedly in the proof of our main result, Theorem 2.1.

The definition of adhesive categories uses the notion of van Kampen square.

Definition 1.1 (van Kampen square). A *van Kampen (VK) square* (i) is a pushout which satisfies the following condition: given a commutative cube (ii) of which (i) forms the bottom face and the back faces are pullbacks, the front faces are pullbacks if and only if the top face is a pushout.

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

(i)

$$\begin{tikzcd}
& C' \arrow[dl, "m'"'] \arrow[dr, "f'"] \arrow[ddd, "c"] & \\
A' \arrow[dr, "g'"'] \arrow[ddd, "a"'] & & B' \arrow[dl, "n'"] \arrow[ddd, "b"] \\
& D' \arrow[ddd, "d"] & \\
& C \arrow[dl, "m"'] \arrow[dr, "f"] & \\
A \arrow[dr, "g"'] & & B \arrow[dl, "n"] \\
& D &
\end{tikzcd}$$

(ii)

Definition 1.2 (Adhesive category). A category $\mathbb{C}$ is said to be *adhesive* if it has pullbacks, pushouts along monos, and these latter are VK-squares.

Given $m : C \to A$ and $g : A \to D$, we say that B is a *pushout complement* of (m, g) when there exist $f : C \to B$ and $n : B \to D$ such that the resulting diamond (i) is a pushout diagram.

We shall need the following properties of adhesive categories for our constructions. The proof of the following lemma can be found in [11].

Lemma 1.3. *Let $\mathbb{C}$ be an adhesive category.*

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

The following simple lemma has not been published previously.

Lemma 1.4. Consider the following diagram in an adhesive category $\mathbb{C}$. If the outer region is a pushout, the right square is a pullback, and morphisms k, r, u and w are mono, then the left square is a pushout.

$$\begin{tikzcd}
A \arrow[r, tail, "k"] \arrow[d, "l"'] & B \arrow[r, tail, "r"] \arrow[d, "s"] & E \arrow[d, "v"] \\
C \arrow[r, tail, "u"'] & D \arrow[r, tail, "w"'] & F
\end{tikzcd}$$

Proof. The exterior pushout is stable under pullback along $w : D \rightarrowtail F$, as illustrated below.

$$\begin{tikzcd}[row sep=small, column sep=small]
 & A \arrow[dl, "l"'] \arrow[d] \arrow[dr, "k"] & & \\
C \arrow[d] \arrow[dr, "u"] & A \arrow[dl, "l"] \arrow[dr, "k"] & B \arrow[d] \arrow[dr] & \\
C \arrow[dr, "u"'] & D \arrow[d] \arrow[dr] & B \arrow[dr, tail, "r"] & B \arrow[dl, "s"'] \arrow[d, tail, "r"] \\
 & D \arrow[dr, tail, "w"'] & D \arrow[d, tail, "w"] & E \arrow[dl, "v"] \\
 & & F &
\end{tikzcd}$$

□

Examples of adhesive categories include **Set**, the category of sets and functions, and **Graph**, the category of graphs and their morphisms. Toposes, as well as slice and coslice categories over adhesive categories are adhesive. Indeed, several graph structures relevant to computer science form adhesive categories (cf. [11]).

1.2 2-categories and cospan bicategories

In this section we give only a minimal introduction to 2-categories and cospan bicategories. For an introductory treatment, the reader should refer to [14]. Roughly, a ***2-category*** $\mathbb{C}$ is a category where homsets (that is the collections of arrows between any pair of objects) are categories and, correspondingly, whose composition maps are functors. Explicitly, a 2-category $\mathbb{B}$ consists of the following.

- A class of *objects* denoted $X, Y, Z, \ldots$
- For any $X, Y \in \mathbb{C}$, a category $\mathbb{C}(X, Y)$. The objects $\mathbb{C}(X, Y)$ are called *1-cells*, or more often, *arrows* or *morphisms*, and denoted by $f\colon X \to Y$. Its morphisms are called *2-cells*, are written $\alpha\colon f \Rightarrow g\colon X \to Y$, or sometimes simply $\alpha\colon f \Rightarrow g$. Composition in $\mathbb{C}(X, Y)$ is denoted by $\bullet$ and referred to as '*vertical*' composition. Identity 2-cells are denoted by $1_f\colon f \Rightarrow f$. Isomorphic 2-cells are occasionally denoted as $\alpha\colon f \cong g$;
- For any objects X, Y, Z there is a functor $.\colon \mathbb{C}(Y, Z) \times \mathbb{C}(X, Y) \to \mathbb{C}(X, Z)$, the so-called '*horizontal*' composition, which we shall often denote by mere juxtaposition. On objects, the functor is just the ordinary composition in the underlying "ordinary" category. On arrows, the functor provides a horizontal composition of 2-cells; it is associative and admits 1_{id_X} as identities.

A *bicategory* can be thought of, intuitively, as a 2-category where associativity and identity laws of horizontal composition hold up to isomorphisms. We shall denote all associativity isomorphisms by ζ, as for example, $\zeta : h(gf) \Rightarrow (hg)f$. The isomorphisms are required to respect the well-known coherence axioms [15].

Cospan Bicategories. We will assume $\mathbb{C}$ to be a category with *chosen pushouts*. That is, for arrows $m : C \to A$ and $f : C \to B$, there exists a unique "chosen" object $A +_C B$ and arrows $i_1 : A \to A +_C B$ and $i_2 : B \to A +_C B$ such that the resulting square is a pushout. By the universality of pushouts, given any other object D and arrows $g : A \to D$ and $n : B \to D$ which render the resulting square a pushout, there exists a unique isomorphism $\alpha : A +_C B \to D$ such that $\alpha i_1 = g$ and $\alpha i_2 = n$. We shall adopt the convention of *always* labelling the morphisms into the chosen pushout by i_1 and i_2; when considering more than one chosen pushout we shall use the context in order to disambiguate.

The bicategory of cospans $\mathrm{Cospan}(\mathbb{C})$ has the same objects as $\mathbb{C}$, but arrows from I_1 to I_2 are cospans.

$$I_1 \xrightarrow{f} C \xleftarrow{g} I_2$$

We will denote such cospans $C_f^g : I_1 \to I_2$ or $C_{f:I_1}^{g:I_2}$, and omit f (resp. g) when I_1 (resp. I_2) is an initial object. We shall refer to I_1 and I_2 as the input and the output interfaces of C^g_f. Intuitively, we can think of a cospan as a generalised context, where C are the internals, (the image via g of) I_2 represents the public view of C, and (the image via f of) I_1 the view of C afforded to the 'holes' in it.

A 2-cell $h\colon C^g_f \Rightarrow C'^{g'}_{f'} : I_1 \to I_2$ is an arrow $h\colon C \to C'$ in $\mathbb{C}$ satisfying $hf = f'$ and $hg = g'$. The 2-cells that are iso (i.e. invertible) provide a canonical notion of "structural congruence." We shall denote the bicategory of cospans which has the 2-cells limited to isomorphisms by $\mathrm{Cospan}^{\cong}\,\mathbb{C}$. Given cospans $C^g_f : I_1 \to I_2$ and $D^{g'}_{f'} : I_2 \to I_3$, their composition $D^{g'}_{f'} \circ C^g_f : I_1 \to I_3$ is the cospan $(C +_{I_2} D)^{i_2 g'}_{i_1 f} : I_1 \to I_3$, as illustrated by the pushout diagram below.

$$\begin{tikzcd} & & C +_{I_2} D & & \\ I_1 \arrow[r, "f"] & C \arrow[ur, "i_1"] & I_2 \arrow[l, "g"'] \arrow[r, "f'"] & D \arrow[ul, "i_2"'] & I_3 \arrow[l, "g'"'] \end{tikzcd}$$

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

In the construction of 2 we shall need certain linearity restrictions. In particular, the notion of input-linear cospan.

Definition 1.5 (Linearity). A cospan C^g_m is said to be *input-linear* when m is a mono. A cospan C^n_m is said to be *linear* when both m and n are mono.

When working over an adhesive category, a simple corollary of the first part of Lemma 1.3 is that the composition of two input-linear cospans yields an input-linear cospan. Similarly, composition preserves linearity.

Definition 1.6 (Linear Cospans). Assuming that $\mathbb{C}$ is adhesive, let $\mathrm{ILC}(\mathbb{C})$ be the bicategory consisting of input-linear cospans and 2-isomorphisms. Similarly, let $\mathrm{LC}(\mathbb{C})$ be the bicategory of linear cospans and 2-isomorphisms.

1.3 Reactive systems and GRPOs

Here we shall briefly recall an extension of Leifer and Milner's notion of reactive system to two dimensional categories as introduced by the authors previously [17]. In this paper we shall consider cospan bicategories with isomorphic 2-cells, and therefore, we shall be concerned with reactive systems over such bicategories.

The intuition behind the 2-dimensional structure is that, while arrows of the underlying category are viewed as contexts, the (isomorphic) 2-cells are thought of as "proofs of structural congruence" between contexts (recall that a 2-cell $\varphi : a \Rightarrow a' : A \to B$ is an isomorphism when it is an isomorphism in $\mathbf{C}(A, B)$, that is, there exists a 2-cell $\psi : a' \Rightarrow a$ such that $\psi \bullet \varphi = 1_a$ and $\varphi \bullet \psi = 1_{a'}$). Recall that in the particular case of the bicategory $\mathrm{Cospan}^{\cong}(\mathbf{Graph})$, the 2-cells are precisely graph isomorphisms which respect the input and output interfaces.

Definition 1.7 (Reactive System). A *reactive system* $\mathbf{C}$ consists of

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

The reactive contexts are those inside which evaluation may occur. To reflect composition 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 $a \longrightarrow\!\!\triangleright\, dr$ if there is $\langle l, r\rangle \in \mathcal{R}$, $d \in \mathbb{D}$ and $\alpha\colon dl \Rightarrow a$. This represents that, up to structural congruence α, a is the left-hand side l of a reduction rule in a reaction context d.

Leifer and Milner [13] 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.

For category theorists, a groupoidal-relative-pushout (GRPO) can be described concisely as a bipushout in a pseudo-slice category. We refer the reader to [17] for a more accessible definition and fundamental properties. Note that although GRPOs are introduced there in the setting of G-categories (2-categories with iso 2-cells), the development is easily transferred to bicategories with iso 2-cells. Here we give a brief sketch. Given a 2-cell $\alpha\colon ca \Rightarrow db\colon I_1 \to I_4$, as illustrated in ($i$) below, a candidate is a tuple $C = \langle I_5, e, f, g, \beta, \gamma, \delta\rangle$, as illustrated in ($ii$) below, so that the 2-cells paste together

$$\begin{tikzcd}[column sep=small]
 & I_4 & \\
I_2 \arrow[ur, "c"] & \alpha & I_3 \arrow[ul, "d"'] \\
 & I_1 \arrow[ul, "a"] \arrow[ur, "b"'] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}[column sep=small]
 & I_4 & \\
 & \gamma \quad \delta & \\
I_2 \arrow[uur, "c"] \arrow[r, "e"] & I_5 \arrow[uu, "g"] & I_3 \arrow[uul, "d"'] \arrow[l, "f"'] \\
 & \beta & \\
 & I_1 \arrow[uul, "a"] \arrow[uur, "b"'] &
\end{tikzcd}$$

(i) $\qquad\qquad\qquad$ (ii)

(taking into account the associativity isomorphisms) to give α. It is a GRPO if it is the smallest such candidate, in the following sense: given another such candidate

C' there exists a mediating morphism $u : I_5 \to I_5'$ and appropriate 2-cells which make the two candidates compatible. Such a mediating morphism is required to be *essentially unique*, meaning that given any other mediating morphism $u' : I_5 \to I_5'$, there exists a unique isomorphic 2-cell $\xi \colon u \Rightarrow u'$ which makes the two mediating morphisms compatible.

Definition 1.8 (GIPO). Diagram (i) is said to be a G-idem-pushout (GIPO) if $\langle I_4, c, d, \mathrm{id}, \alpha, \mathbf{1}_c, \mathbf{1}_d \rangle$ is its GRPO.

Definition 1.9 (LTS). For $\mathbf{C}$ a reactive system and $\mathbb{B}$ its underlying bicategory, define GLTS($\mathbf{C}$) as follows:

- the states GLTS($\mathbf{C}$) are iso-classes of arrows with domain the chosen object 0 (two arrows $a, a' : 0 \to I_1$ are in the same iso-class when there exists an isomorphic 2-cell $\varphi : a \Rightarrow a'$). We shall denote the iso-class of a as $[a] : 0 \to I_1$;
- there is a transition $[a] \xrightarrow{[f]} [dr]$ if there exists a 2-cell α, a rule $\langle l, r \rangle \in \mathcal{R}$, and $d \in \mathbb{D}$, such that the diagram below is a GIPO.

$$\begin{tikzcd} & I_4 & \\ I_2 \arrow[ur, "f"] & \alpha & I_3 \arrow[ul, "d"'] \\ & 0 \arrow[ul, "a"] \arrow[ur, "l"'] & \end{tikzcd} \tag{1}$$

Definition 1.10. A reactive system $\mathbf{C}$ is said to have *redex-GRPOs* when every redex-square (1), with a, 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 an extension of a theorem for 2-categories which can be found in [17], the proof requires only slight modifications for the extension in generality.

Theorem 1.11. *Let* $\mathbf{C}$ *be a reactive system with redex-GRPOs. Then bisimulation on* GLTS($\mathbf{C}$) *is a congruence with respect to all the contexts of* $\mathbf{C}$.

Theorem 1.11 is quite robust with respect to the equivalence under consideration. For example, trace and failures equivalences on GLTS($\mathbf{C}$) are also congruences. Therefore, the derived lts may be considered as very well-behaved.

2 Constructing GRPOs for input linear cospans

In this section we present a general construction of GRPOs for a class of reactive systems over cospan bicategories. We shall conclude with several examples of

GRPOs in the bicategory ILC(**Graph**), the bicategory of input-linear cospans in the category of directed graphs.

Formally, let $\mathbb{C}$ be an adhesive category. We shall prove that ILC($\mathbb{C}$) has GRPOs. This implies that any reactive system over ILC($\mathbb{C}$) has redex-GRPOs and therefore can be given a canonical labelled transition system semantics.

Theorem 2.1. ILC($\mathbb{C}$) *has GRPOs.*

We shall first outline an algorithm for the construction of the desired minimal candidate. A redex square in ILC($\mathbb{C}$), as illustrated in diagram (*i*) below, amounts to a commutative diagram (*ii*) in $\mathbb{C}$, with α an isomorphism. We shall adopt the convention of $\rightsquigarrow$ representing the isomorphisms of $\mathbb{C}$ which correspond to the 2-cells of ILC($\mathbb{C}$).

$$\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, "C"] \arrow[rr, phantom, "\alpha"] & & I_3 \arrow[ul, "D"'] \\
 & I_1 \arrow[ul, "A"] \arrow[ur, "B"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & C \arrow[d, "i_2"] & I_4 \arrow[l, "o_C"'] \arrow[r, "o_D"] & D \arrow[d, "i_2"'] & \\
I_2 \arrow[ur, tail, "\iota_C"] \arrow[dr, "o_A"'] & A +_{I_2} C \arrow[rr, squiggly, "\alpha"] & & B +_{I_3} D & I_3 \arrow[ul, tail, "\iota_D"'] \arrow[dl, "o_B"] \\
 & A \arrow[u, "i_1"'] & I_1 \arrow[l, tail, "\iota_A"] \arrow[r, tail, "\iota_B"'] & B \arrow[u, "i_1"] &
\end{tikzcd}$$

(*i*) (*ii*)

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 $\mu' k = \mu$. We shall abuse notation by confusing the subobject (equivalence classes of monos) with its representative μ (one mono).

In the following, we let $X = B +_{I_3} D$ and $Y = A \cup B$, a subobject of X. We obtain $\mu : Y \to X$, $\epsilon_1 : A \to Y$ and $\epsilon_2 : B \to Y$ satisfying $\mu\epsilon_1 = \alpha i_1$ and $\mu\epsilon_2 = i_1$.

Algorithm 2.2 (GRPO Construction in ILC($\mathbb{C}$)**).** The construction of the components of the minimal candidate is outlined below. They are illustrated in diagrams (*iii*) and (*iv*). We obtain the components by constructing:

1. G as the pullback of $\alpha i_2 : C \to X$ and $i_2 : D \to X$;
2. E as the pullback of $\mu : Y \to X$ and $\alpha i_2 : C \to X$;
3. F as the pullback of $\mu : Y \to X$ and $i_2 : D \to X$;
4. I_5 as the pullback of $\rho_2 : F \to D$ and $\pi_2 : G \to D$; Notice that due to the properties of pullbacks, we obtain a morphism $o_E : I_5 \to E$ such that all the faces of (*iv*) are pullbacks.

Proof. We give an outline of the proof in two parts. Firstly, we show that the components constructed in Algorithm 2.2 are a candidate, secondly, we show that the candidate is universal.

Construction of candidate.

$$
\begin{tikzcd}
& I_4 & \\
I_2 \arrow[ur, "C"] \arrow[r, dashed, "E"'] & I_5 \arrow[u, dashed, "G"] \arrow[ul, phantom, "\gamma"] \arrow[ur, phantom, "\delta"] & I_3 \arrow[ul, "D"'] \arrow[l, dashed, "F"] \\
& I_1 \arrow[ul, "A"] \arrow[ur, "B"'] \arrow[u, phantom, "\beta"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & I_5 \arrow[dll, "o_E"'] \arrow[dr, tail, "\iota_G"] \arrow[ddd, dashed, "o_F"] & \\
E \arrow[dr, tail, "\rho_1"] \arrow[ddd, "\sigma_1"'] & & & G \arrow[dll, "\pi_1"'] \arrow[ddd, "\pi_2"] \\
& C \arrow[ddd, "\alpha i_2"] & & \\
& & F \arrow[dr, tail, "\rho_2"] \arrow[dll, "\sigma_2"'] & \\
Y \arrow[dr, tail, "\mu"'] & & & D \arrow[dll, "i_2"] \\
& X & &
\end{tikzcd}
$$

(iii) (iv)

Construct the cube (iv), by taking pullbacks as explained in Algorithm 2.2. The exterior rectangles of diagrams (v) and (vi) are pushouts (dia. (ii)), and therefore, commutative.

$$
\begin{tikzcd}
I_2 \arrow[r, tail, "\iota_E"] \arrow[d, "o_A"'] \arrow[rr, bend left, tail, "\iota_C"] & E \arrow[r, tail, "\rho_1"] \arrow[d, "\sigma_1"'] \arrow[dr, phantom, "(\dagger)"] & C \arrow[d, "\alpha i_2"] \\
A \arrow[r, tail, "\epsilon_1"'] & Y \arrow[r, tail, "\mu"'] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
I_3 \arrow[r, tail, "\iota_F"] \arrow[d, "o_B"'] \arrow[rr, bend left, tail, "\iota_D"] & F \arrow[r, tail, "\rho_2"] \arrow[d, "\sigma_2"'] \arrow[dr, phantom, "(\ddagger)"] & D \arrow[d, "i_2"] \\
B \arrow[r, tail, "\epsilon_2"'] & Y \arrow[r, tail, "\mu"'] & X
\end{tikzcd}
$$

(v) (vi)

Using the pullback property, we obtain $\iota_E : I_2 \to E$ satisfying $\sigma_1\iota_E = i_1 o_A$ and $\rho_1\iota_E = \iota_C$. Similarly, we obtain $\iota_F : I_3 \to F$ which satisfies analogous equations. We can now use Lemma 1.4 to conclude that the left hand squares of diagrams (v) and (vi) are pushouts, which in turn implies that (†) and (‡) are pushouts, using ordinary pushout pasting.

Since all the side faces of (iv) are pullbacks and the bottom face is a pushout, we can conclude by adhesiveness that the top face is a pushout. Similarly, the front left face being a pushout implies that the back right face is a pushout. Using the fact that G was defined as a pullback, we obtain a unique morphism $o_G : I_4 \to G$ such that $\pi_1 o_G = o_C$ and $\pi_2 o_G = o_D$. We summarise the parts of the candidate constructed so far in diagram (vii), below.

$$
\begin{tikzcd}
& & I_4 \arrow[dl, bend right, "o_C"'] \arrow[dr, bend left, "o_D"] \arrow[d, dashed, "o_G"] & & \\
& C & G \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & D & \\
I_2 \arrow[ur, bend left, tail, "\iota_C"] \arrow[r, tail, "\iota_E"'] & E \arrow[u, tail, "\rho_1"] & I_5 \arrow[l, "o_E"] \arrow[r, "o_F"'] \arrow[u, tail, "\iota_G"'] & F \arrow[u, tail, "\rho_2"'] & I_3 \arrow[ul, bend right, tail, "\iota_D"'] \arrow[l, tail, "\iota_F"]
\end{tikzcd}
$$

(vii)

Let $E +_{I_5} G$ denote the chosen pushout of o_E and ι_G and let $\gamma : C \to E +_{I_5} G$ be the unique induced isomorphism. Similarly, we obtain a unique isomorphism $\delta : F +_{I_5} G \to D$. Referring back to diagrams (iv) and (v), we have $A +_{I_2} E \cong$

$Y \cong B +_{I_3} F$. Let $\beta : A +_{I_2} E \to B +_{I_3} F$ denote the unique compatible isomorphism. The isomorphisms are illustrated in diagram (*viii*) as arrows of $\mathbb{C}$, and as 2-cells of ILC($\mathbb{C}$) in diagram (*iii*). It is straightforward but tedious to check that $\delta B \bullet G\beta \bullet \gamma A = \alpha$ in ILC($\mathbb{C}$).

$$\begin{tikzcd}
C \arrow[r, "\gamma", tail] & E +_{I_5} G & G \arrow[l] \arrow[r] & F +_{I_5} G \arrow[r, "\delta", tail] & D \\
I_2 \arrow[u, "\iota_C"] \arrow[r, "\iota_E"] \arrow[d, "o_A"'] & E \arrow[u] \arrow[d] & I_5 \arrow[u, "\iota_G"] \arrow[l, "o_E"'] \arrow[r, "o_F"] & F \arrow[u] \arrow[d] & I_3 \arrow[u, "\iota_D"'] \arrow[l, "\iota_{R_3}"'] \arrow[d, "o_B"] \\
A \arrow[r] & A +_{I_2} E \arrow[rr, squiggly, "\beta"'] & & B +_{I_3} F & B \arrow[l]
\end{tikzcd}$$

(*viii*)

$$\begin{tikzcd}
 & I_4 & \\
 & G' \arrow[u] & \\
I_2 \arrow[uur, "C"] \arrow[r, "E'"] \arrow[ur, phantom, "\gamma'"] & I_6 \arrow[u] & I_3 \arrow[uul, "D"'] \arrow[l, "F'"'] \arrow[ul, phantom, "\delta'"] \\
 & \arrow[u, phantom, "\beta'"] & \\
 & I_1 \arrow[uul, "A"] \arrow[uur, "B"'] &
\end{tikzcd}$$

(*ix*)

Universality. Suppose that there is another candidate, as illustrated in diagram (*ix*). Letting $\pi_1' = \gamma'^{-1} i_2 : G' \to C$ and $\pi_2{}' = \delta' i_2 : G' \to D$ and using the fact that G is a pullback object, we get an arrow $\lambda : G' \to G$ such that $\pi_1 \lambda = \pi_1' : G' \to C$ and $\pi_2 \lambda = \pi_2' : G' \to C$. Consider diagram (*x*) below, where T denotes $B +_{I_3} F'$. We have an isomorphism $(B +_{I_3} \delta')\zeta : T +_{I_6} G' \to X$, Let $\eta : T \to X$ and $\theta : G' \to X$ denote the corresponding morphisms derived by precomposing with i_1 and i_2.

$$\begin{tikzcd}
 & E' \arrow[dr, "\beta' i_2"] & & F' \arrow[dl, "i_2"'] & \\
I_2 \arrow[ur, tail, "\iota_{E'}"] \arrow[dr, "o_A"'] & & T & & I_3 \arrow[ul, tail, "\iota_F'"'] \arrow[dl, "o_B"] \\
 & A \arrow[ur, tail, "\beta' i_1"'] & & B \arrow[ul, tail, "i_1"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I_6 \arrow[dl, "o_T"'] \arrow[dr, tail, "\iota_{G'}"] & \\
T \arrow[dr, tail, "\eta"'] & & G' \arrow[dl, "\theta"] \\
 & X &
\end{tikzcd}$$

(*x*) (*xi*)

Let $\mu' : Y \to T$ be the arrow induced by $\beta' i_1$ and i_1 in diagram (*x*). One can verify that $\eta\mu' = \mu$.

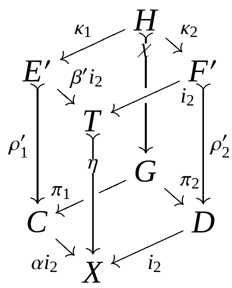
In diagram (*xii*) below, we let ρ_1' denote $\gamma'^{-1} i_1$, while in diagram (*xiii*) let $\rho_2' = \delta' i_1$. Consider diagram (*xii*) again, and note that regions (†) and (†) + (‡) are pushouts. Hence, (‡) is a pushout and, using the second part of Lemma 1.3, a pullback. Thus there exists a morphism $\nu_1 : E \to E'$ such that $\rho_1' \nu_1 = \rho_1$ and $\beta' i_2 \nu_1 = \mu' \sigma_1$. A similar chain of reasoning involving diagram (*xiii*) allows us to derive the existence of $\nu_2 : F \to F'$ which satisfies $\rho_2' \nu_2 = \rho_2$ and $i_2 \nu_2 = \mu' \sigma_2$.

One can now deduce that the leftmost squares in the two diagrams are pullbacks, and therefore, that ν_1 and ν_2 are both mono, since they are pullbacks of mono μ' along, respectively, $\beta' i_2$ and i_2.

$$\begin{tikzcd}
 & I_6 \arrow[r, tail, "\iota_{G'}"] \arrow[d, "o_{E'}"'] \arrow[dr, phantom, "(\dagger)"] & G' \arrow[d, "\pi_1'"] \\
E \arrow[r, tail, "\nu_1"] \arrow[d, "\sigma_1"'] & E' \arrow[r, tail, "\rho_1'"] \arrow[d, "\beta' i_2"'] \arrow[dr, phantom, "(\ddagger)"] & C \arrow[d, "\alpha i_2"] \\
Y \arrow[r, tail, "\mu'"'] & T \arrow[r, tail, "\eta"'] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I_6 \arrow[r, tail, "\iota_{G'}"] \arrow[d, "o_{F'}"'] & G' \arrow[d, "\pi_2'"] \\
F \arrow[r, tail, "\nu_2"] \arrow[d, "\sigma_2"'] & F' \arrow[r, tail, "\rho_2'"] \arrow[d, "i_2"'] & D \arrow[d, "i_2"] \\
Y \arrow[r, tail, "\mu'"'] & T \arrow[r, tail, "\eta"'] & X
\end{tikzcd}$$

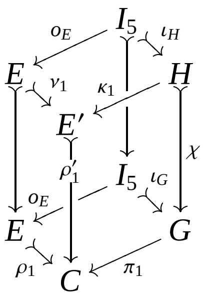
(*xii*) (*xiii*)

In diagram (*xiv*), H and morphisms κ_1, κ_2 and χ are chosen so that the two rear faces are pullbacks. One could do this, e.g., by first taking the pullback of π_2 and ρ'_2 and obtaining κ_1 from the pullback property of the front left face.

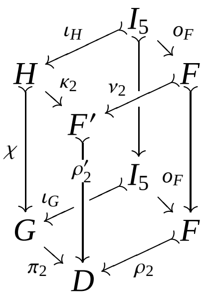


(*xiv*)

One may consider H as the mediating morphism from I_5 to I_6. Indeed, in diagrams (*xv*), (*xvi*) and we use adhesiveness in order to deduce that the top faces are pushouts.



(*xv*)



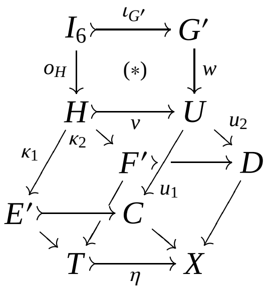
(*xvi*)

Because the top face of diagram (*xiv*) is a pullback, we obtain a unique morphism $o_H : I_6 \to H$ such that $\kappa_1 o_H = o_{E'}$ and $\kappa_2 o_H = o_{F'}$.

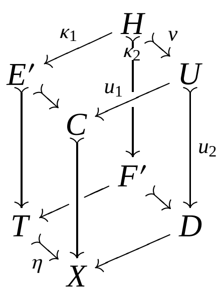
We shall show that diagram

$$\begin{tikzcd} I_6 \arrow[r, tail, "\iota_{G'}"] \arrow[d, "o_H"'] & G' \arrow[d, "\lambda"] \\ H \arrow[r, tail, "\chi"'] & G \end{tikzcd}$$

(*xvii*)



(*xviii*)



(*xix*)

is a pushout. First notice that it is commutative, we have $\pi_1\chi o_H = \rho'_1\kappa_1 o_H = \rho'_1 o_{E'} = \pi'_1\iota_{G'} = \pi_1\lambda\iota_{G'}$ and similarly $\pi_2\chi o_H = \rho'_2\kappa_2 o_H = \rho'_2 o_{F'} = \pi'_2\iota'_G = \pi_2\lambda\iota_{G'}$

which implies that $\chi o_H = \lambda\iota_{G'}$, using the fact that the bottom face of diagram (*xiv*) is a pullback.

We shall now construct diagram (*xviii*). We start by taking the pushout $(*)$. Now since region (†) of diagram (*xii*) is a pushout, there exists a unique morphism $u_1 : U \to C$ such that $u_1 w = \pi'_1$ and $u_1 v = \rho'_1\kappa_1$. Similarly, using the fact that the corresponding region of diagram (*xiii*) is a pushout, there exists a unique morphism $u_2 : U \to D$ such that $u_2 w = \pi'_2$ and $u_2 v = \rho'_2\kappa_2$. Using the standard decomposition property of pushouts, the two newly constructed regions are pushouts. The two lower regions of diagram (*xviii*) are the pushouts which appear as the two front faces of diagram (*xiv*). The left face of the cube is the top face of (*xiv*) which is a pullback. Spinning the cube around into diagram (*xix*) we use the fact that the bottom and top faces are pushouts, and the back faces pullbacks to deduce that the front right face is a pullback.

Since also the bottom face of diagram (*xiv*) is a pullback, we obtain a unique isomorphism $\zeta : G \to U$ such that $u_1\zeta = \pi_1$ and $u_2\zeta = \pi_2$. As we have assumed that $(*)$ is a pushout, to prove that diagram (*xvii*) is a pushout it remains to show that $\zeta\lambda = w$ and $\zeta\chi = v$.

Indeed $u_1\zeta\lambda = \pi_1\lambda = \pi'_1 = u_1 w$ and $u_2\zeta\lambda = \pi_2\lambda = \pi'_2 = u_2 w$ which implies that $\zeta\lambda = w$. Also $u_1\zeta\chi = \pi_1\chi = \rho'_1\kappa_1 = u_1 v$ and $u_2\zeta\chi = \pi_2\chi = \rho'_2\kappa_2 = u_2 v$ which implies that $\zeta\chi = v$.

Let $\varphi : E' \to E +_{I_5} H$, $\psi : F +_{I_5} H \to F'$ and $\tau : H +_{I_6} G' \to G$ be the unique induced isomorphisms. It can be verified that $\tau E \bullet G'\varphi \bullet \gamma' = \gamma$, $\delta' \bullet G'\psi \bullet \tau F = \delta$ and $\psi B \bullet H\beta \bullet \varphi A = \beta'$ in ILC($\mathbb{C}$).

Essential uniqueness of $H : I_5 \to I_6$ can be shown by using the third part of Lemma 1.3, applied to diagram (*xvii*), we omit the details here. □

Notice that we can give a simplified presentation of the construction of the minimal candidate if we assume that all the cospans are linear.

Algorithm 2.3 (GRPO Construction in LC($\mathbb{C}$)**).** When all the morphisms in diagram (*ii*) are mono, constructing the pullback amounts to computing the intersection of subobjects of X. Indeed, we let:

1. $G = C \cap D$;
2. $E = Y \cap C$;
3. $F = Y \cap D$;
4. $I_5 = F \cap G = F \cap E = E \cap G$.

In the simple examples below, meant to illustrate the application of Algorithm 2.2, we shall consider ILC(**Graph**) as our category of contexts. In the accompanying Figures 2–4, we label the nodes of the graphs in order to clarify the

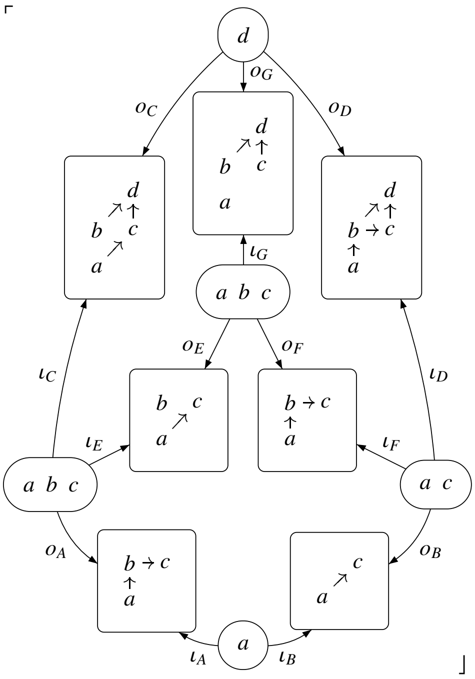


Figure 2: GRPO in LC(**Graph**)

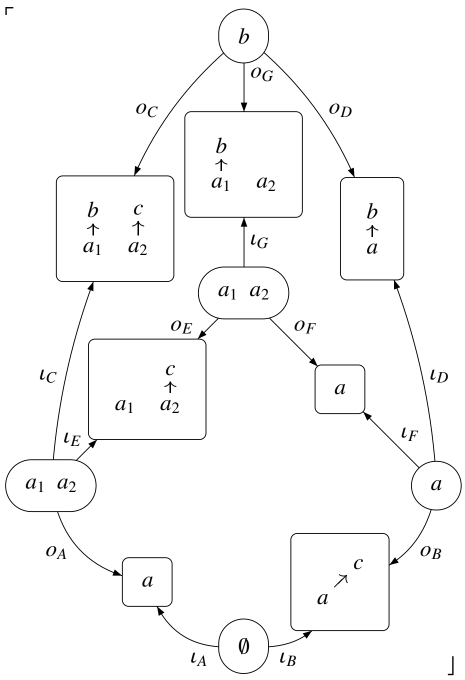


Figure 3: GRPO in ILC(**Graph**)

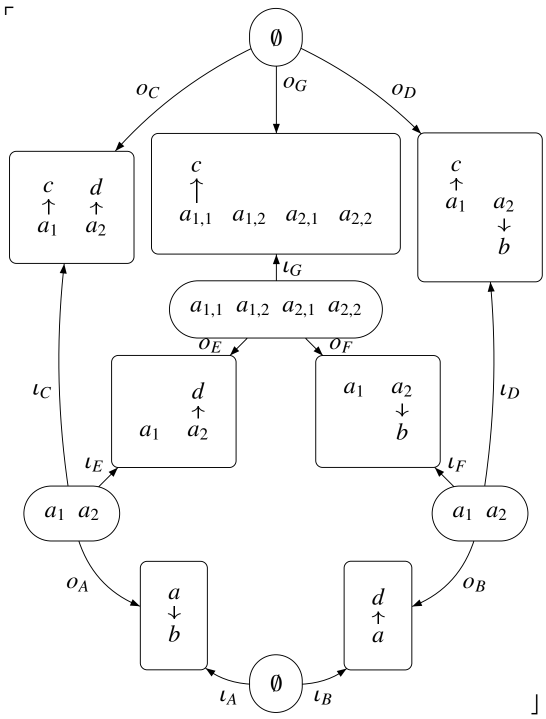


Figure 4: GRPO in ILC(**Graph**)

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. We shall make reference to the directed graphs below.

$$X_1 = \begin{tikzcd}[row sep=small, column sep=small] & d \\ b \arrow[ur] \arrow[r] & c \arrow[u] \\ a \arrow[u] \arrow[ur] & \end{tikzcd} \qquad X_2 = \begin{tikzcd}[row sep=small, column sep=small] b & c \\ a \arrow[u] \arrow[ur] & \end{tikzcd} \qquad X_3 = \begin{tikzcd}[row sep=small, column sep=small] c & d \\ & a \arrow[ul] \arrow[u] \arrow[d] \\ & b \end{tikzcd}$$

Example 2.4. Graph X_1 can be decomposed as illustrated by the exterior of Figure 2. Here, all the graph morphisms are injective. The reader may wish to go through the steps of Algorithm 2.2 to construct the GRPO is this particular case, it is illustrated in the interior of Figure 2.

Example 2.5. Graph X_2 can be broken up as illustrated by the exterior of Figure 3. Notice that o_A is not injective. The GRPO is illustrated in the interior of Figure 3.

Example 2.6. We illustrate a GRPO for a partition of X_3 in Figure 4. Notice that here both o_A and o_B are not injective.

3 Applications

In this section we shall introduce two immediate applications of our construction of GRPOs in input linear cospan bicategories.

First, after a brief review of the theory of double-pushout graph rewriting, we shall show that we may use the construction to derive congruences for graphs enriched with an output interface. Graph contexts here are input-linear cospans of graphs. The labelled transition system derived using our technology admits labels which are the smallest contexts which allow a double-pushout rewrite. The results in this section are closely related to rewriting via borrowed contexts due to Ehrig and König [5]. In particular, in Theorem 3.10 we show that the labelled transition systems are essentially the same, the difference being that the nodes and labels of our transition system are quotiented by isomorphism.

Our results both shed light and extend rewriting via borrowed contexts. Firstly, because borrowed contexts correspond to GIPOs, they satisfy a universal property. Secondly, we show that borrowed contexts fall within the framework of reactive systems [13, 17, 16] 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 theorem for reactive systems (Theorem 1.11). Finally, due to the generality of Theorem 2.1, we relax some of the technical conditions imposed by Ehrig and König and introduce the notion of extended borrowed contexts (Definition 3.12).

Our second application concerns Milner's theory of bigraphs [9]. We show how bigraphs can be represented by a cospan bicategory over an adhesive category. As a consequence, we derive labelled transition systems for reactive systems over an input-linear variant of bigraphs. Such variant is incomparable to Milner's, as our formalism allows bigraphs not allowed in Milner's approach, and vice-versa. It appears that a closer correspondence to Milner's theory would be achieved by considering an output-linear variant. It is, therefore, an interesting line of future work whether one could derive a general construction of GRPOs for an interesting and general class of output-linear cospan bicategories.

Another consequence of representing bigraphs by cospans is that because of the close correspondence between double-pushout rewriting systems and reactive systems over cospan bicategories demonstrated by Lemma 3.5, one can view bigraphical reactive systems as certain double-pushout rewriting systems. In particular, this could mean that some of the theory and technology developed for the latter can perhaps be applied successfully to the former.

3.1 Double-pushout rewriting and borrowed contexts

Double-pushout (DPO) graph rewriting is a well known, widely studied topic [2]. Introduced in [6], it has recently been generalised in [3, 11, 4]. We shall describe a variant of DPO-graph rewriting, working at the level of an arbitrary adhesive category $\mathbb{C}$. The reader may of course, safely substitute the adhesive category **Graph** of graphs and graph homomorphisms for $\mathbb{C}$. We start by relating DPO rewriting and reactive systems.

Definition 3.1 (Rewrite Rule). A rewrite rule p is a span

$$L \xleftarrow{l} K \xrightarrow{r} R \tag{2}$$

in $\mathbb{C}$. Observe that we *do not* assume that either l or r are mono.

Here L and R represent respectively the left and right-hand side of the rule, while K is the context, 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 $f : L \to C$.

Definition 3.2 (Adhesive Grammar). An adhesive grammar G is a pair $\langle \mathbb{C}, \mathbf{P} \rangle$ where $\mathbb{C}$ is an adhesive category and $\mathbf{P}$ is a set of arbitrary rewrite rules.

Definition 3.3 (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 the following diagram are pushouts.

$$\begin{tikzcd} L \arrow[d, "f"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "g"] & R \arrow[d, "h"] \\ C & E \arrow[l, "v"] \arrow[r, "w"'] & D \end{tikzcd} \tag{3}$$

We shall write $C \mathrel{\longrightarrow\!\!\!\blacktriangleright} D$ if there exist $p \in P$ and $f : L \to C$ such that $C \mathrel{\longrightarrow\!\!\!\blacktriangleright}_{p,f} D$.

Proposition 3.4. An adhesive grammar can be seen as a reactive system on the category Cospan($\mathbb{C}$). Let 0 denote the empty graph, and the set $\mathcal{R}$ contain for each rewrite rule p as in (2), a pair

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

We choose all arrows of Cospan($\mathbb{C}$) be reactive. Let $\mathrel{\longrightarrow\!\!\!\triangleright}$ denote the resulting reaction relation.

It is useful to point out that the process described in Proposition 3.4 can be reversed. Starting with a reactive system over a cospan category over $\mathbb{C}$ with chosen object the initial object $0 \in \mathbb{C}$, one can obtain a double-pushout rewriting system by adding a rewrite rule $L \xleftarrow{l} K \xrightarrow{r} R$ for every $\left\langle 0 \to L \xleftarrow{l} K\,,\ 0 \to R \xleftarrow{r} K \right\rangle \in \mathcal{R}$. It turns out that these "encodings" are actually very well behaved. In this section we shall present Lemma 3.5 which shows that the DPO rewrite relation is exactly the reactive system reaction relation when it makes sense to compare them. The main result of this section, Theorem 3.10 makes the correspondence even stronger by relating the operational theories developed for the two approaches. Indeed, we shall show that it is equivalent to consider GIPOs in reactive systems over cospans on the one hand and Ehrig and König's borrowed contexts on the other.

The following lemma is similar to a previously published result (cf. [7]) and can be considered folklore. As well as being simple to prove, it is a relatively little-known result; it is therefore worthwhile to state and prove it here. It is crucial for us because it serves as a foundation for relating the theory of DPO rewriting and the theory of reactive systems.

We use the shorthand $C \mathrel{\longrightarrow\!\!\!\triangleright} D$ to mean that C and D are cospans with empty input and output contexts.

Lemma 3.5. *$C \mathrel{\longrightarrow\!\!\!\blacktriangleright} D$ iff $C \mathrel{\longrightarrow\!\!\!\triangleright} D$.*

Proof. If $C \mathrel{\longrightarrow\!\!\!\triangleright} D$ then $C \cong E_g \circ L^l$ and $D \cong E_g \circ R^r$, which is equivalent to requiring that the two squares below are pushouts.

$$
\begin{tikzcd}
 & & 0 \arrow[d, "!"] & & \\
 & C & E \arrow[l, "v"'] \arrow[r, "w"] & D & \\
0 \arrow[r, "!"] & L \arrow[u, "f"] & K \arrow[l, "l"] \arrow[r, "r"'] \arrow[u, "g"] & R \arrow[u, "h"'] & 0 \arrow[l, "!"']
\end{tikzcd}
$$

This means $C \longrightarrow D$, since the middle part of the diagram is a DPO rewrite as in (3). Note that the output interface of C and D is actually arbitrary, as long as it factors through E. □

The equivalence exhibited by Lemma 3.5 between the rewrite relation in a DPO rewriting system and the reaction relation of a reactive system over a cospan bicategory is a bridge which relates the two theories.

This may be compared with an easy lemma about term rewriting. In a ground term rewriting system over a signature Σ, one usually defines the rewrite relation as follows: a term c rewrites to d, written $c \longrightarrow d$, if one can find l as a subterm of c, and replace it with r. On the other hand, one may recast such a term rewriting system as a reactive system over the free linear Lawvere theory $\mathbb{C}_\Sigma$ [17]. In $\mathbb{C}_\Sigma$, the objects are natural numbers, while arrows $m \to n$ are n-tuples of terms built up from Σ which contain exactly one occurrence each of m ordered variables. Composition in this category is substitution of terms, done in the obvious way. The reaction rules $\mathcal{R}$ consist of pairs $\langle l, r \rangle$, where $l, r : 0 \to 1$ are respectively the left and the right hand sides of a rewrite rule. One then defines the rewrite relation as follows: $c \longrightarrow\!\!\triangleright\, d$ if $c = c' \circ l$, $d = c' \circ r$ and $\langle l, r \rangle \in \mathcal{R}$. In other words, the reaction rules are closed under all linear contexts. It is easy to show that $c \longrightarrow d$ if and only if $c \longrightarrow\!\!\triangleright\, d$.

Indeed, Lemma 3.5 implies that the reactive systems on Cospan($\mathbb{C}$) with all cospans reactive include all DPO rewriting systems over $\mathbb{C}$.

Lts Semantics for DPO Rewriting Systems. In order to apply Theorem 2.1, we restrict to graph rewriting systems corresponding to bicategories of input-linear cospans.

The following notion, which we shall refer to as input-linear rewrite rule application is sometimes referred to in graph rewriting literature as rewriting with *injective matching*.

Definition 3.6 (Input-Linear Rewrite Application). Object C rewrites to D with rule p input-linearly, in symbols $C \longrightarrow^{il}_{p,f} D$, if $C \longrightarrow_{p,f} D$ and in addition f, g and h of (3) are mono.

Definition 3.7. For $G = \langle \mathbb{C}, \mathbf{P} \rangle$ an adhesive grammar with an input-linear rewrite relation $\longrightarrow^{il}$, we construct a reactive system $\mathbf{C}$ over the bicategory of input-linear cospans ILC($\mathbb{C}$) using the translation of Proposition 3.4.

Lemma 3.5 clearly specialises to double-pushout rewriting systems and reactive systems over input-linear cospan bicategories.

Proposition 3.8. Suppose that $\mathbb{C}$ is adhesive and consider an arbitrary adhesive grammar G, and let $\mathbf{C}$ be the corresponding reactive system over an *input-linear* cospan bicategory. Let $\longrightarrow\!\!\triangleright$ denote the reaction relation in $\mathbf{C}$. Then

$$C \longrightarrow^{il}_{p,f} D \text{ iff } C \longrightarrow\!\!\triangleright\ D.$$

We are now ready to examine the operational theory derived using GRPOs. Let LTS(G) be GLTS($\mathbf{C}$) (cf. Definition 1.9). Using the congruence theorem (Theorem 1.11) for bisimulation on labelled transition systems generated by GI-POs [17], we obtain the following.

Corollary 3.9. *Bisimulation on* LTS(G) *is a congruence.*

A rewrite rule (2) 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 recent work due to Ehrig and König, called rewriting with borrowed contexts [5]. Remarkably, we derive the same labelled transition systems, modulo quotienting the transitions obtained in the borrowed-contexts approach by isomorphism, as we explain below.

Precisely, given an adhesive grammar $G = \langle \mathbb{C}, \mathbf{P} \rangle$, where $\mathbf{P}$ consists of linear rewrite rules, let RBC(G) be the lts derived via rewriting with borrowed contexts. (We refer to reader to [5] for the details of such construction, which cannot be spelt out here.)

Theorem 3.10. *There is a transition* $[G^{o_G:I_2}] \xrightarrow{[F^{o_F:I_5}_{\iota_F:I_2}]} [H^{o_H:I_5}]$ *in* LTS(G) *if and only if there is a transition* $G^{o_G:I_2} \xrightarrow{F^{o_F:I_5}_{\iota_F:I_2}} H^{o_H:I_5}$ *in* RBC(G).

Proof. (1) From GIPOs to Borrowed Contexts:

Suppose that $[G^{o_G}] \xrightarrow{[F^{o_F}_{\iota_F}]} [H^{o_H}]$ in LTS(G). then $F^{o_F}_{\iota_F}$ must be a part of an GIPO diagram. Since every GIPO can be constructed as a GRPO, we have a redex diagram

$$\begin{tikzcd}
 & I_4 & \\
I_2 \arrow[ur, "F'"] \arrow[r, dashed, "F"] \arrow[r, phantom, shift left=3, "\gamma"] & I_5 \arrow[u, dashed, "G"] \arrow[d, phantom, "\beta"] & I_3 \arrow[ul, "C'"'] \arrow[l, dashed, "C"'] \arrow[l, phantom, shift right=3, "\delta"'] \\
 & 0 \arrow[ul, "G"] \arrow[ur, "L"'] &
\end{tikzcd}$$

as the outside of the diagram, with $\langle L^{o_L}, R^{o_R}\rangle$ being a reaction rule, corresponding via the translation of Lemma 3.5 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 above is the GRPO obtained via the construction of Algorithm 2.2.

We also have $H^{o_H} \cong C^{o_C}_{i_C} \circ R^{o_R}$, which means that the diagram (i) below is a pushout

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

(i)

with $o_H : I_5 \to H$ equal to $\theta_2 \circ o_C$.

Recall that from the construction, we have that

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

(ii) (iii) (iv) (v)

diagram (ii) is a pushout, diagram (iii) is a pullback, diagrams (iv) and (v) are pushouts.

Notice that in diagrams (i) to (v) we have indicated which morphisms are assumed to be mono in the construction of Algorithm 2.2. While Ehrig and König assume that all morphisms are mono, it shall be useful for us here to indicate only the necessary ones as we shall use the extra generality to extend the notion of borrowed context in Definition 3.12.

We can construct the following diagram from the five diagrams above:

$$\begin{tikzcd}
G \cap L \arrow[r, tail, "\epsilon_1'"] \arrow[d, "\epsilon_2'"'] & L \arrow[d, tail, "\epsilon_2"] & I_3 \arrow[l, "o_L"'] \arrow[r, "o_R"] \arrow[d, tail, "\iota_C"] & R \arrow[d, tail, "\theta_1"] \\
G \arrow[r, tail, "\epsilon_1"] & Y & C \arrow[l, "\sigma_2"'] \arrow[r, "\theta_2"] & H \\
I_2 \arrow[u, "o_G"] \arrow[r, tail, "\iota_F"'] & F \arrow[u, tail, "\sigma_1"] & I_5 \arrow[l, "o_F"] \arrow[u, "o_C"'] &
\end{tikzcd}$$

This is exactly the definition of $F^{o_F}_{\iota_F}$ constituting a borrowed context for G^{o_G}, ie. $G^{o_G} \xrightarrow{F^{o_F}_{\iota_F}} H^{o_H}$ in RBC(G).

(2) From Borrowed Contexts to GIPOs In this section of the proof we shall use the notation from [5]. We shall also follow *loc. cit.* in assuming that all morphisms are mono. Given a borrowed context diagram (*vi*) below,

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

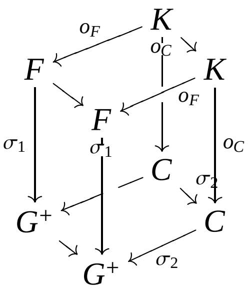
(*vi*)

we shall show that one may construct a GIPO. First recall that a commutative diagram (*vi*) is a borrowed context diagram when squares (*1*), (*2*), (*3*) and (*4*) are pushouts, while square (*5*) is a pullback.

Indeed, consider the redex diagram (*vii*)

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

(*vii*)



(*viii*)

and assume without loss of generality that $L +_I C = G^+$. Let $\alpha : F +_J G \to G^+$ be the unique isomorphism such that $\alpha i_i = \sigma_1 : F \to G^+$ and $\alpha i_2 = \epsilon_1 : G \to G^+$. Note that the isomorphism exists because square (*4*) in diagram (*vi*) is a pushout.

Since square (*1*) of diagram (*vi*) is a pushout of monos and we are in an adhesive category, it is also a pullback. This implies that $L \cup G = G^+$, because in adhesive categories one forms unions of subobjects by forming the pushout of their intersection. The central cube diagram from the construction of GRPOs is illustrated in (*viii*); in the construction we use only the fact that square (*5*) of diagram (*vi*) is a pullback. Thus, the GRPO of the redex square (*vii*) is $\langle K, F, C, \mathrm{id}, \mathrm{id}, \mathrm{id}, \alpha\rangle$, meaning that it is a GIPO. $\square$

We say that two graphs $A^{o_A:I}$ and $A'^{o_{A'}:I}$ with the same output interface are equivalent when there exists an isomorphism $\varphi : A \to A'$ such that $\varphi o_A = o_{A'}$. Similarly, we say that two transitions from $A^{o_A:I}$ to $B^{o_B:J}$, with labels $C^{o_C:J}_{\iota_C:I}$ and $C'^{o_{C'}:J}_{\iota_{C'}:I}$, respectively, are equivalent when there exists an isomorphism $\psi : C \to C'$ with $\psi\iota_C = \iota_{C'}$ and $\psi o_C = o_{C'}$.

Let $\mathrm{RBC}_{\cong}(G)$ be the lts whose nodes are equivalence classes of the nodes of $\mathrm{RBC}(G)$ and whose transitions are equivalence classes of transitions of $\mathrm{RBC}(G)$. The following corollary is a straightforward consequence of Theorem 3.10.

Corollary 3.11. $\mathrm{RBC}_{\cong}(G) = \mathrm{LTS}(G)$.

The results of this section can be seen from two different perspectives. From the point of view of reactive systems, the borrowed context conditions of Ehrig and König, once extended by allowing the appropriate morphisms to be non-mono, can be seen as being an elegant and simple characterisation of GIPOs in a input-linear cospan bicategory. On the other hand, the results of this section show that borrowed contexts satisfy a universal property. This means that the congruence theorem for bisimilarity of Ehrig and König can actually be seen as a special case of Theorem 1.11. Similarly, other results and technology developed for reactive systems transfers to the setting of borrowed contexts, this includes the congruence theorems for equivalences other than bisimilarity as well as the elegant technique for deriving "weak" transition systems (in the sense of weak bisimilarity) developed by Jensen in his upcoming PhD thesis [8].

We end this section with an extension of borrowed contexts suggested by the linearity conditions imposed in the construction of Algorithm 2.2.

Definition 3.12 (Extended Borrowed Contexts). Given an adhesive grammar $G = \langle \mathbb{C}, \mathbf{P} \rangle$ with an *arbitrary* set of rules $\mathbf{P}$, we shall construct a labelled transition system with:

1. Nodes: Graphs with output interfaces, $J \xrightarrow{o_G} G$ where o_G is arbitrary;
2. 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 $[G^{o_G}] \xrightarrow{[F^{o_F}_{\iota_F}]} [H^{o_H}]$ if there exists a commutative diagram as illustrated below, where

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

(*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 3.10 and the congruence Theorem 1.11, we have the following.

Corollary 3.13. Bisimulation on the labelled transition system resulting from Definition 3.12 is a congruence with respect to arbitrary input-linear graph contexts, that is cospans

$$J \xrightarrow{\iota_F} F \xleftarrow{o_F} K$$

where ι_F is mono and o_F is arbitrary.

3.2 Bigraphs as cospans

Bigraphs were introduced by Milner to model dynamic systems with independent locality and connectivity structures (cf. [9] for a comprehensive exposition), and have been used by Jensen and Milner [10] to model and derive an lts for the asynchronous π-calculus.

In this section, we recast the notion of bigraph in our approach, and obtain structures very similar to Milner's. Although we briefly discuss the differences between our product and Milner's bigraphs, we remark that a perfect match is not our objective. Rather, we aim at demonstrating that algebraic constructs relevant to the semantics of mobility and communication fall naturally within the realm of cospan bicategories over adhesive categories. To that end, we introduce the adhesive category of place-link graphs, which can be considered "bigraphs without interfaces." Interfaces will then be added when we consider the bicategory of cospans over the category of place-link graphs. We do not develop the notions of width, inactive sites, nor parametric reaction rules, which do not seem to have an effect on the actual construction of GRPOs for bigraphs. The construction shall be given by Algorithm 2.2.

We define a *place-graph* to be a directed graph with nodes labelled over an alphabet Σ. Additionally, there is an arity function $ar : \Sigma \to \mathbb{N}$ (the natural numbers), and place-graph morphisms are directed graph morphisms which preserve node labelling. The intuition is that the elements of Σ are (names of) controls, each equipped with an ordered set of ports. The connectivity of each control is determined by the number of its ports. For a place-graph G with node-set V and labelling function $l : V \to \Sigma$, we denote the ith port of v by v_i, where $0 \leq i < ar(l(v))$. One can construct the total *set of ports of* G by taking the disjoint union:

$$P = \sum_{v \in V} \{ v_i \mid 0 \leq i < ar(l(v)) \}.$$

Definition 3.14 (Place-Link Graphs). A place-link (pl) graph is a place-graph G over a set of controls Σ together with a set S and a link map $l : P \to S$, where P is

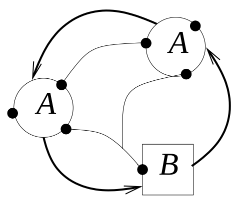


Figure 5: Typical place-link graph

the set of ports of G. A place-link morphism $\langle f_0, f_1, f_2\rangle : G \to G'$ is a place-graph morphism $\langle f_0, f_1\rangle$ together with a function $f_2 : S \to S'$ such that $l' f_p = f_2 l$, where $f_p : P \to P'$ is the morphism sending v_i to $f_0(v)_i$. Let $\mathbf{PLGraph}_\Sigma$ be the category of pl-graphs and pl-graph morphisms over a set of controls Σ.

The intuition is that S is the set of equivalence classes of ports in G. When two ports are in the same equivalence class (that is they map to the same element of S), we say that they are connected. Figure 5 illustrates a typical place-link graph with two kind of controls – A and B, respectively with three and two ports – where directed arcs represent the place structure, and the undirected ones are the elements of S.

It is easy to construct for each Σ a category $\mathbb{X}_\Sigma$ so that $\mathbf{PLGraph}_\Sigma \cong \mathbf{Set}^{\mathbb{X}_\Sigma}$, in other words, $\mathbf{PLGraph}_\Sigma$ is a presheaf category and, as such, adhesive.

Definition 3.15 (PL-Graphs with Interfaces). We refer to $\mathrm{Cospan}^{\cong}(\mathbf{PLGraph}_\Sigma)$ as the bicategory of pl-graphs with interfaces. Restricting to input-linear cospans, we obtain the bicategory $\mathrm{ILC}(\mathbf{PLGraph}_\Sigma)$.

Corollary 3.16. $\mathrm{ILC}(\mathbf{PLGraph}_\Sigma)$ *has GRPOs, calculated using Algorithm 2.2.*

There are two aspects of our bicategory of pl-graphs with interfaces which generalise the theory of bigraphs. Firstly, the theory of bigraphs one traditionally considers only “discrete” interfaces, i.e., discrete place-graphs (which can be seen as strings over the alphabet Σ) together with *name sets* (cf. Definition 3.17). Secondly, place graphs are usually forests of trees, and their input (resp. output) interfaces reach only leaves (resp. roots). Fortunately, we can apply the necessary restrictions and still be able to perform the construction of Algorithm 2.2.

Definition 3.17 (Discrete PL-Graph). A discrete pl-graph $\langle m, X\rangle$, where m is a finite ordinal labelled over Σ and X is a finite *set of names*, is a pl-graph with m as its set of nodes and no edges. Its link map is the injection $P \to P + X$, for P the set of ports of m.

Let $\mathbf{TPLGraph}_\Sigma$ be the full subcategory of $\mathbf{PLGraph}_\Sigma$ consisting of pl-graphs with forests of trees as place-graphs.

Definition 3.18 (Bicategory of Bigraphs). The bicategory of bigraphs $\mathbf{Bigraph}_\Sigma$ over a set of controls Σ has:

- Objects: Discrete pl-graphs $\langle m, X\rangle$;
- Arrows: Input-linear cospans $\langle m, X\rangle \rightarrowtail G \leftarrow \langle n, Y\rangle$ with $G \in \mathbf{TPLGraph}_\Sigma$; additionally, the left interface must reach only the leaves of G while the right interface must reach only the roots of G;
- 2-cells: isomorphisms between cospans.

Theorem 3.19. $\mathbf{Bigraph}_\Sigma$ *has GRPOs.*

Proof. It suffices to verify that Algorithm 2.2 preserves the conditions on cospans, and that I_5 is a discrete bigraph. □

Interestingly, a main difference between bigraphs as defined here and Milner's [9] is the effect of input-linearity on aliasing of names. Milner's formalism allows a bigraph which equates names from its input interface, which is disallowed by our input-linearity condition. Conversely, our formalism allows a bigraph which equates names within its output interface, an operation not allowed in Milner's. Observe that since (G)RPOs-derived contexts only exercise output interfaces, expressive power may be lost in some applications by not being able to equate ports in the input interface. Although this appears to be often compensated for by equating ports in output interfaces, we are currently investigating a generalised notion of relative pushout which acts at the same time on both interfaces.

In particular, in the link-graphs illustrated below,

$$\overset{x \frown y}{\langle 0, \{x, y\}\rangle \to \langle 0, \emptyset\rangle} \qquad\qquad \overset{x \smile y}{\langle 0, \emptyset\rangle \to \langle 0, \{x, y\}\rangle}$$

the left one is not input-linear (as input names x and y are mapped to the same element), and therefore not allowed by our formalism. On the other hand, the right one, which is input-linear but not output-linear (as output names x and y are coalesced), is allowed in our formalism but not in Milner's.

It appears that an output-linear version of Definition 3.18 would capture the existing bigraphs almost exactly. We leave a general construction of GIPOs for output-linear cospan bicategories as future work.

4 Conclusion

We have constructed groupoidal relative pushouts 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. We have focused on two notable, comprehensive examples ('metamodels'), the theories of double-pushout graph rewriting and of bigraphs.

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

References

[1] V. Danos and C. Laneve. Graphs for core molecular biology. In *International Workshop on Computational Methods in Systems Biology, CMSB '03*, 2003.

[2] H. Ehrig, G. Engels, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors. *Handbook of Graph Grammars and Computing by Graph Transformation, Volumes 1-3*. World Scientific, 1997-1999.

[3] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high-level replacement systems. *Math. Struct. in Comp. Science*, 1, 1991.

[4] H. Ehrig, A. Habel, J. Padberg, and U. Prange. Adhesive high-level replacement categories and systems. In *International Conference on Graph Transformation ICGT '04*, 2004. To appear.

[5] H. Ehrig and B. König. Deriving bisimulation congruences in the dpo approach to graph rewriting. In *Foundations of Software Science and Computation Structures FOSSACS '04*, volume 2987 of *Lecture Notes in Computer Science LNCS*, pages 151–166. Springer, 2004.

[6] H. Ehrig, M. Pfender, and H. Schneider. Graph-grammars: an algebraic approach. In *IEEE Conf. on Automata and Switching Theory*, pages 167–180, 1973.

[7] F. Gadducci and R. Heckel. An inductive view of graph transformation. In *Workshop on Algebraic Development Techniques*, pages 223–237, 1997.

[8] O. H. Jensen. *Bigraphs*. PhD thesis, University of Aalborg, 2004. To appear.

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

[10] O. H. Jensen and R. Milner. Bigraphs and transitions. In *Principles of Programming Languages, POPL '03*. ACM Press, 2003.

[11] S. Lack and P. Sobociński. Adhesive categories. In *Foundations of Software Science and Computation Structures FOSSACS '04*, volume 2987 of *Lecture Notes in Computer Science LNCS*, pages 273–288. Springer, 2004.

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

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

[14] S. MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer Verlag, 2nd edition, 1992.

[15] S. MacLane and R. Paré. Coherence for bicategories and indexed categories. *Journal of Pure and Applied Algebra*, pages 59–80, 1985.

[16] V. Sassone and P. Sobociński. Deriving bisimulation congruences: 2-categories vs. precategories. In *FOSSACS '03*, volume 2620 of *LNCS*. Springer, 2003.

[17] V. Sassone and P. Sobociński. Deriving bisimulation congruences using 2-categories. *Nordic Journal of Computing*, 10(2):163–183, 2003.

[18] P. Sewell. From rewrite rules to bisimulation congruences. *Lecture Notes in Computer Science*, 1466:269–284, 1998.

Recent BRICS Report Series Publications

RS-04-11 Vladimiro Sassone and Paweł Sobociński. *Congruences for Contextual Graph-Rewriting*. June 2004. 29 pp.

RS-04-10 Daniele Varacca, Hagen Völzer, and Glynn Winskel. *Probabilistic Event Structures and Domains*. June 2004.

RS-04-9 Ivan B. Damgård, Serge Fehr, and Louis Salvail. *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks*. May 2004. 22 pp.

RS-04-8 Petr Jančar and Jiří Srba. *Highly Undecidable Questions for Process Algebras*. April 2004. 25 pp. To appear in Lévy, Mayr and Mitchell, editors, *3rd IFIP International Conference on Theoretical Computer Science*, TCS '04 Proceedings, 2004.

RS-04-7 Mojmír Křetínský, Vojtěch Řehák, and Jan Strejček. *On the Expressive Power of Extended Process Rewrite Systems*. April 2004. 18 pp.

RS-04-6 Gudmund Skovbjerg Frandsen and Igor E. Shparlinski. *On Reducing a System of Equations to a Single Equation*. March 2004. 11 pp. To appear in Schicho and Singer, editors, *ACM SIGSAM International Symposium on Symbolic and Algebraic Computation*, ISSAC '04 Proceedings, 2004.

RS-04-5 Biernacki Dariusz and Danvy Olivier. *From Interpreter to Logic Engine by Defunctionalization*. March 2004. 20 pp. To appear in Bruynooghe, editor, *International Symposium on Logic Based Program Development and Transformation, LOPSTR '03 Proceedings*, Revised Selected Papers, LNCS, 2003. This report supersedes the earlier BRICS report RS-03-25.

RS-04-4 Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. *Optimal Strategies in Priced Timed Game Automata*. February 2004. 32 pp.

RS-04-3 Mads Sig Ager, Olivier Danvy, and Jan Midtgaard. *A Functional Correspondence between Call-by-Need Evaluators and Lazy Abstract Machines*. February 2004. 17 pp. This report supersedes the earlier BRICS report RS-03-24. Extended version of an article to appear in *Information Processing Letters*.