Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 173 (2007) 121–137
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Directed Bigraphs ⋆

Davide Grohmann[a,1] Marino Miculan[a,2]

[a] *Department of Mathematics and Computer Science, University of Udine, Italy*

Abstract

We introduce *directed bigraphs*, a bigraphical meta-model for describing computational paradigms dealing with *locations* and *resource communications*. Directed bigraphs subsume and generalize both original Milner's and Sassone-Sobociński's variants of bigraphs. The key novelty is that directed bigraphs take account of the "resource request flow" inside link graphs, from controls to edges (through names), by means of the new notion of *directed link graph*. We give RPO and IPO constructions for this model, generalizing and unifying the constructions independently given by Jensen-Milner and Sassone-Sobociński in their respective variants. Moreover, the very same construction can be used for calculating RPBs as well, and hence also luxes (when these exist). Therefore, directed bigraphs can be used as a general theory for deriving labelled transition systems (and congruence bisimulations) from (possibly open) reactive systems.

Keywords: Bigraphical models; Categorical models of concurrent, reactive, distributed, mobile systems.

1 Introduction

The fundamental importance of *labelled transition systems* (LTS) for defining the dynamics of a calculus is well known. In spite of this, defining a satisfactory LTS for a given calculus is not an easy task. Essentially, the problem boils down to identify correctly the *observations*, that is, the "labels" of the LTS, which must represent *exactly* (i.e., no more and no less) all possible interactions with any context which can surround a system. Traditionally, LTSs are crafted "by hand", but the more complex is the calculus, the more difficult is to devise its LTS.

For this reason, often the semantics of a calculus is given by means of a *reaction* (or *reduction*) system. Reaction systems are easier to define, understand and justify than LTSs, but are not as useful in supporting tools and analytic techniques such as bisimulations and model checking. Thus, a natural question is whether, and how, is possible to construct a "good" labelled transitions system out of a reduction system.

In the last years much work has been spent in looking for general procedures for deriving LTSs from reduction systems. Sewell [13] argued that the labels c of transi-

⋆ Work supported by Italian MIUR project 2005015824 ART.

[1] Email: grohmann@dimi.uniud.it

[2] Email: miculan@dimi.uniud.it

1571-0661
doi:10.1016/j.entcs.2007.02.031

tions of a term t are the contexts $c[\cdot]$ such that $c[t]$ yields a reaction; remarkably, the bisimulation induced by a such LTS is always a congruence. However, we want to take as labels the contexts really relevant to t only, i.e., in $c[t]$ the reaction has to involve (part of) t and not only the surrounding context c. To this end, a major breakthrough has been achieved by Leifer and Milner with the observation that a natural concept of "minimal context" is elegantly expressed by the categorical notions of *relative pushout* (RPO) and *idem-relative pushout* (IPO) [7,8]. The notion of RPO has been later generalized to *groupoidal* RPO for dealing with syntactic congruences [11], and dualized into (groupoidal) *relative pullback* (RPB) to take into account also open (i.e., non-ground) terms and reaction rules. Eventually, RPO and RPB have been merged into the single concept of *locally universal hexagons* (luxes) [5].

Now, given this general and elegant theory, we have to find the categories where the calculi and systems used in Concurrency can be conveniently represented, and RPOs, RPBs and luxes can be constructed.

To this end, an emerging meta-model are Milner's *bigraphs* [9,10], for which a construction of RPOs has been given in [3]. A bigraph is composed by two orthogonal structures: a hierarchical *place graph* describing locations, and a *link (hyper-)graph* describing connections. These structures allow to represent many formalisms such as CCS, π-calculus, Ambients, and Petri nets among others. Thus, bigraphs can be seen as a promising meta-model for Concurrency.

On the other hand, Sassone and Sobociński presented in [12] a general approach for constructing RPOs in a wide range of models, namely those which can be expressed as input-linear cospans over *adhesive* categories [6]. Adhesive categories are quite common in Computer Science; e.g., presheaf categories (and hence **Set** and **Graph**) are adhesive. An input-linear cospan $X \rightarrowtail A \leftarrow Y$ represents a system A whose input and output interfaces are X and Y, respectively. However, despite its generality, this construction cannot be applied to Milner's bigraphs, due to the input-linearity condition: bigraphs are actually *output-linear* (and not input-linear) cospans in an adhesive category of place-link graphs [12].

Summarizing, so far we have two kinds of bigraphs: "output-linear" (i.e. original Milner's) bigraphs, with Jensen-Milner's RPO construction; and "input-linear" bigraphs, with Sassone-Sobocińksi's RPO construction. These two categories and constructions do not generalize each other, although they agree on the intersection (i.e., input- and output-linear bigraphs). A natural question then arises: is there a generalization of both kinds of bigraphs, with an RPO construction subsuming both Jensen-Milner's and Sassone-Sobociński's constructions?

The answer is affirmative: in this paper we introduce *directed bigraphs*, which subsume and generalize both previous theories. A directed bigraph is composed by a place graph and a *directed link graph*, which is a natural generalization of input-linear link graphs and output-linear link graphs. In this model, we give a construction of RPOs (and IPOs), generalizing and unifying the known constructions in the previous models (Actually, the IPO construction for input-linear bigraphs obtained in this way is the first one, up to our knowledge). Moreover, since the (pre)category of directed link graphs turns out to be self-dual, the RPO construction can be used for calculating RPBs as well, and hence for the construction of luxes.

Intuitively, the basic idea of directed ling graphs is to notice that names are not resources on their own, but only a way for denoting (abstract) resources (here represented by the edges). In a system, a name may be not denoting any resource (i.e., not associated to any edge); in this case, the name can be seen as a formal parameter of the system which is *asking* through it for a resource from outside itself. Thus, we can discern a "resource request flow" which starts from control ports, goes through names and terminates in edges. In output-linear link graphs, this request flow enters a system from its inner interface (i.e., the system offers its resources to inner modules) and exits through its outer interface (i.e., the system asks for resources to the outer environment); that is, the flow moves *ascending* the place graph hierarchy. The converse happens in input-linear link graphs, where the requests flow *descends* the place graph hierarchy. Therefore, we can generalize both situations by allowing resource requests to go *in both directions:* a module may ask for resources and offer resources on each interface at once. In order to avoid inconsistencies, however, we must take care of the "polarity" of names in interfaces, according as their meaning flows "upward" or "downward"—hence the adjective *directed*.

The rest of the paper is organized as follows. In Section 2 we present the precategories 'DLG and 'DBig of directed link graphs and directed bigraphs, and their basic properties. The constructions of RPOs and IPOs for directed link graphs are described in Section 3. As an application, in Section 4 we show how input-linear and output-linear link graphs are subsumed by directed link graphs. Due to lack of space, we cannot describe how directed link graphs can be conveniently used for representing specific calculi, even with binders (such as λ-calculus) without the need of further extensions; we refer the interested reader to [1]. Conclusions and directions for future work are in Section 5.

2 Directed link graphs and bigraphs

In this section we introduce directed link graphs, and present their main properties. In order to allow a direct comparison with traditional (i.e., output-linear, Milner's) bigraphs, we work with *precategories*. We refer the reader to [4, §3] and [8] for an introduction to the theory of supported monoidal precategories.

Let $\mathcal{K}$ be a given signature of controls.

Definition 2.1 *A* polarized interface X *is a pair of disjoint sets of names* $X = (X^-, X^+)$*; the two components are called* downward *and* upward *faces, respectively.*

A directed link graph $A : X \to Y$ *is* $A = (V, E, ctrl, link)$ *where* X *and* Y *are the* inner *and* outer *interfaces,* V *is the set of* nodes*,* E *is the set of* edges*,* $ctrl : V \to \mathcal{K}$ *is the* control map*, and* $link : \mathsf{Pnt}(A) \to \mathsf{Lnk}(A)$ *is the* link map*, where the* ports*, the* points *and the* links *of* A *are defined as follows:*

$$\mathsf{Prt}(A) \triangleq \sum_{v \in V} ar(ctrl(v)) \qquad \mathsf{Pnt}(A) \triangleq X^+ \uplus Y^- \uplus \mathsf{Prt}(A) \qquad \mathsf{Lnk}(A) \triangleq X^- \uplus Y^+ \uplus E$$

The link map cannot connect downward and upward names of the same interface,

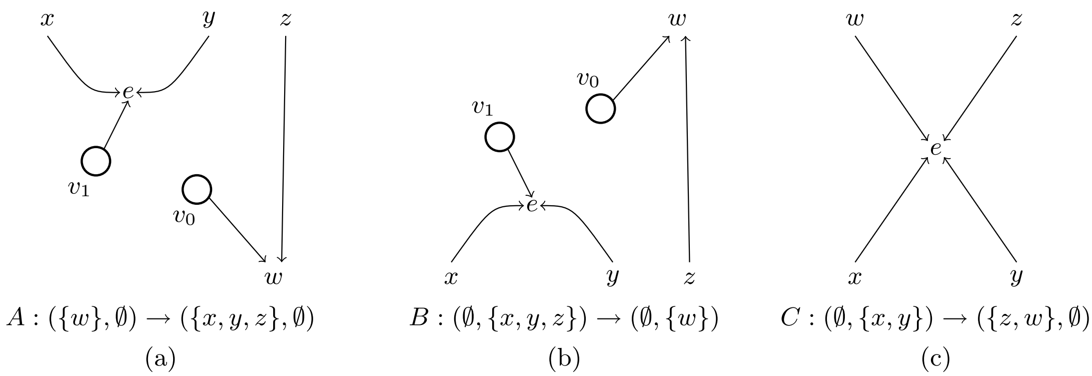


Fig. 1. Examples of directed link graphs.

i.e., the following condition must hold: $(link(X^+) \cap X^-) \cup (link(Y^-) \cap Y^+) = \emptyset$.

Directed link graphs are graphically depicted much like ordinary link graphs, with the difference that edges are explicitly represented as vertices of the graph, and not as hyper-arcs connecting points and names; points and names are associated to edges (or other names) by (simple, non hyper) directed arcs. Some examples are given in Figure 1. This notation aims to make clear the "resource request flow": ports and names in the interfaces can be associated either to internal or to external resources. In the first case, ports and names are connected to an edge; these names are "inward" because they declare to the context how to get to an internal resource. In the second case, the ports and names are connected to an outward name, which is waiting to be plugged by the context into a resource.

Notice that input-linear (output-linear, respectively) link graphs are just special cases of this definition: just restrict to empty upward (downward, respectively) interfaces (Figure 1.a and b). However, there are directed link graphs which are neither input-linear nor output-linear, nor any combination of these; e.g. $C \triangleq (\emptyset, \{e\}, \emptyset, \{(x, e), (y, e), (z, e), (w, e)\}) : \{x, y\} \to \{z, w\}$ in Figure 1.c.

Directed link graphs can be alternatively defined as the composition of an input linear link graph and an output linear link graph defined on the same support (as suggested by R. Milner). Notice that to this end, control ports must be partitioned in two subsets: those used in the input linear link graph and those used in the outer linear link graph. This corresponds to assign a precise direction (either upward or downward) to the connections. Notice in this way, the constraint that two names of the same interface cannot be linked together, is automatically ensured.

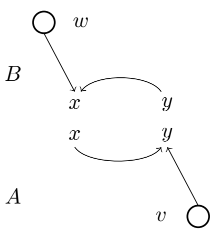
In the following, by "interface" and "link graphs" we will intend "polarized interface" and "directed link graphs" respectively, unless otherwise noted.

Definition 2.2 ('DLG) *The precategory of* directed link graphs *has polarized interfaces as objects, and directed link graphs as morphisms.*

Given two directed link graphs $A_i = (V_i, E_i, ctrl_i, link_i) : X_i \to X_{i+1}$ $(i = 0, 1)$, *the composition* $A_1 \circ A_0 : X_0 \to X_2$ *is defined when the two link graphs have disjoint nodes and edges. In this case,* $A_1 \circ A_0 \triangleq (V, E, ctrl, link)$, *where* $V \triangleq V_0 \uplus V_1$, $ctrl \triangleq ctrl_0 \uplus ctrl_1$, $E \triangleq E_0 \uplus E_1$ *and* $link : X_0^+ \uplus X_2^- \uplus P \to E \uplus X_0^- \uplus X_2^+$ *is defined* *as follows (where $P = \mathsf{Prt}(A_0) \uplus \mathsf{Prt}(A_1)$):*

$$link(p) \triangleq \begin{cases} link_0(p) & \text{if } p \in X_0^+ \uplus \mathsf{Prt}(A_0) \text{ and } link_0(p) \in E_0 \uplus X_0^- \\ link_1(x) & \text{if } p \in X_0^+ \uplus \mathsf{Prt}(A_0) \text{ and } link_0(p) = x \in X_1^+ \\ link_1(p) & \text{if } p \in X_2^- \uplus \mathsf{Prt}(A_1) \text{ and } link_1(p) \in E_1 \uplus X_2^+ \\ link_0(x) & \text{if } p \in X_2^- \uplus \mathsf{Prt}(A_1) \text{ and } link_1(p) = x \in X_1^-. \end{cases}$$

The identity link graph of X is $id_X \triangleq (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, Id_{X^- \uplus X^+}) : X \to X$.



It is easy to check that composition is associative, and that given a link graph $A : X \to Y$, the compositions $A \circ id_X$ and $id_Y \circ A$ are defined and equal to A. Definition 2.1 forbids connections between names of the same interface in order to avoid undefined link maps after compositions. An example is shown aside, where the composition of two apparently unproblematic directed link graphs A, B would yield a "loop" and hence not a directed link graph.

Proposition 2.3 *The precategory $'\mathrm{DLG}$ is self-dual, that is $'\mathrm{DLG} \cong {'\mathrm{DLG}}^{op}$.*

Proof. We can define the functor $\overline{(\_)} : {'\mathrm{DLG}} \to {'\mathrm{DLG}}^{op}$ on objects as $\overline{(X^-, X^+)} \triangleq (X^+, X^-)$, and on a morphism $A = (V, E, ctrl, link) : X \to Y$ as A itself but with swapped interfaces: $\overline{A} \triangleq A : (V, E, ctrl, link) : \overline{Y} \to \overline{X}$. It is easy to check that this is a full and faithful functor, and that $\overline{\overline{A}} = A$. ☐

Definition 2.4 *The* support *of a link graph $A = (V, E, ctrl, link) : X \to Y$ is the set $|A| = V \oplus E$.*

Proposition 2.5 *The precategory $'\mathrm{DLG}$ is well supported.*

Proof. A lengthy check that $|A_1 \circ A_2| = |A_1| \uplus |A_2|$, and that all the properties about support translation are verified. ☐

Definition 2.6 (idle, lean, open, closed, peer) *Let $A : X \to Y$ be a link graph.*

A link $l \in \mathsf{Lnk}(A)$ is idle *if it is not in the image of the link map (i.e., $l \notin link(\mathsf{Pnt}(A))$). The link graph A is* lean *if there are no idle links.*

A link l is open *if it is an inner downward name or an outer upward name (i.e., $l \in X^- \cup Y^+$); it is* closed *if it is an edge.*

A point p is open *if $link(p)$ is an open link; otherwise it is closed. Two points p_1, p_2 are* peer *if they are mapped to the same link, that is, $link(p_1) = link(p_2)$.*

Proposition 2.7 *A link graph $A : X \to Y$ is epi iff there are no peer names in Y^- and no idle names in Y^+. Dually, A is mono iff there are no idle names in X^- and no peer names in X^+.*

A is an isomorphism iff it has no nodes, no edges, and its link map can be decomposed in two bijections $link^+ : X^+ \to Y^+$, $link^- : Y^- \to X^-$.

Definition 2.8 *The* tensor product $\otimes$ *in $'\mathrm{DLG}$ is defined as follows. Given two objects X, Y, if these are pairwise disjoint then $X \otimes Y \triangleq (X^- \uplus Y^-, X^+ \uplus Y^+)$. Given two link graphs $A_i = (V_i, E_i, ctrl_i, link_i) : X_i \to Y_i$ $(i = 0, 1)$, if the tensor* *products of the interfaces are defined and the sets of nodes and edges are pairwise disjoint then the tensor product* $A_0 \otimes A_1 : X_0 \otimes X_1 \to Y_0 \otimes Y_1$ *is defined as* $A_0 \otimes A_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1)$.

It is not difficult to check $|A_1 \otimes A_2| = |A_1| \uplus |A_2|$ and the following proposition.

Proposition 2.9 $'$DLG *is a well-supported monoidal precategory.*

Finally, we can define the *directed bigraphs* as the composition of standard place graphs (see [4, §7] for definitions) and directed link graphs.

Definition 2.10 *An* interface *is composed by a* width *(a finite ordinal) and by a pair of finite sets of names (from a global set* $\mathcal{X}$*).*

Let $I = \langle m, X\rangle$ *and* $J = \langle n, Y\rangle$ *be two interfaces. A* directed bigraph G *with signature* $\mathcal{K}$ *from* I *to* J *is* $G = (V, E, ctrl, prnt, link) : I \to J$*, where* I *and* J *are its* inner *and* outer *interfaces, respectively.* V *and* E *are the sets of nodes and edges respectively, and* $prnt$*,* $ctrl$ *and* $link$ *are the parent, control and link maps, such that* $G^P \triangleq (V, ctrl, prnt) : m \to n$ *is a place graph and* $G^L \triangleq (V, E, ctrl, link) : X \to Y$ *is a directed link graph.*

We denote G as combination of G^P and G^L by $G = \langle G^P, G^L\rangle$. In this notation, a place graph and a (directed) link graph can be put together iff they have the same sets of nodes and edges.

Definition 2.11 **($'$DBig)** *The precategory* $'$DBig *of directed bigraph with signature* $\mathcal{K}$ *has interfaces* $I = \langle m, X\rangle$ *as objects and directed bigraphs* $G = \langle G^P, G^L\rangle : I \to J$ *as morphisms. If* $H : J \to K$ *is another directed bigraph with sets of nodes and edges disjoint from* V *and* E *respectively, then their composition is defined by composing their components, i.e.:*

$$H \circ G \triangleq \langle H^P \circ G^P, H^L \circ G^L\rangle : I \to K.$$

The identity directed bigraph of $I = \langle m, X\rangle$ *is* $\langle id_m, Id_{X^- \uplus X^+}\rangle : I \to I$.

Proposition 2.12 *A directed bigraph* G *in* $'$DBig *is epi (respectively mono) iff its two components* G^P *and* G^L *are epi (respectively mono).*

The isomorphisms in $'$DBig *are all the combinations* $\iota = \langle \iota^P, \iota^L\rangle$ *of an isomorphism in* $'$PLG *and an isomorphism in* $'$DLG.

Definition 2.13 *The* tensor product $\otimes$ *in* $'$DBig *is defined as follows. Given* $I = \langle m, X\rangle$ *and* $J = \langle n, Y\rangle$*, where* X *and* Y *are pairwise disjoint, then* $\langle m, X\rangle \otimes \langle n, Y\rangle \triangleq \langle m + n, (X^- \uplus Y^-, X^+ \uplus Y^+)\rangle$*. The tensor product of two bigraphs* $G_i : I_i \to J_i$ *is defined when the tensor products of the interfaces are defined and the sets of nodes and edges are pairwise disjoint, then:*

$$G_0 \otimes G_1 \triangleq \langle G_0^P \otimes G_1^P, G_0^L \otimes G_1^L\rangle : I_0 \otimes I_1 \to J_0 \otimes J_1.$$

Proposition 2.14 *For every signature* $\mathcal{K}$*, the precategory* $'$DBig *is wide monoidal; the origin is* $\epsilon = \langle 0, (\emptyset, \emptyset)\rangle$ *and the interface* $\langle n, X\rangle$ *has width* n*.*

In virtue of this result, $'$DBig can be used for applying the theory of *wide reaction* *systems* and *wide transition systems* as developed by Jensen and Milner; see [4, §4, §5] for details. To this end, we need to show that $'$DBIG has RPOs and IPOs. Since place graphs are the usual ones, it suffices to show that directed link graphs have RPOs and IPOs; this is the subject of the next section.

3 RPO and IPO for directed link graphs

3.1 Construction of relative pushouts and pullbacks

We first give an idea of how the construction works. Suppose $D_0 : X_0 \to Z$, $D_1 : X_1 \to Z$ is a bound for a span $A_0 : W \to X_0$, $A_1 : W \to X_1$ and we wish to construct the RPO (B_0, B_1, B). In the following we will denote a pair (A_0, A_1) by $\vec{A}$ and the link map of A simply by A. To form the pair $\vec{B}$ we truncate $\vec{D}$ by removing all the edges, nodes and ports not present in $\vec{A}$. Then in the outer interface of $\vec{B}$, we create an outer name for each point unlinked by the truncation: the downward names connected to the same link (name or edge) must be "bound together", i.e. we must consider all the possible ways to associate a downward name of A_0 with one of A_1 and vice versa; further we must equate an upward name of A_0 with one of A_1 only if they are both connected to a point shared between A_0 and A_1. Formally:

Construction 3.1 (RPOs in directed link graphs) A relative pushout $(\vec{B} : \vec{X} \to \hat{X}, B : \hat{X} \to Z)$, for a pair $\vec{A} : W \to \vec{X}$ of link graphs relative to a bound $\vec{D} : \vec{X} \to Z$, will be built in three stages. Since RPOs are preserved by isomorphisms, we can assume the components of X_0 and X_1 disjoint.

nodes and edges If V_i are the nodes of A_i $(i = 0, 1)$, then the nodes of D_i are $V_{D_i} = (V_{\bar{i}} \setminus V_i) \uplus V_2$ for some V_2. Define the nodes of B_i and B to be $V_{B_i} \triangleq V_{\bar{i}} \setminus V_i$ $(i = 0, 1)$ and $V_B \triangleq V_2$. Edges E_i and ports P_i of A_i are treated analogously.

interface Construct the shared codomain $\hat{X} = (\hat{X}^-, \hat{X}^+)$ of $\vec{B}$ as follows: first we define the names in each $X_i = (X_i^-, X_i^+)$, for $i = 0, 1$, that must be mapped into $\hat{X} = (\hat{X}^-, \hat{X}^+)$:

$$X_i'^- \triangleq \{x \in X_i^- \mid \exists y \in X_{\bar{i}}^- \text{ s.t. } A_i(x) = A_{\bar{i}}(y) \text{ or } A_i(x) \in (E_i \setminus E_{\bar{i}})\}$$
$$X_i'^+ \triangleq \{x \in X_i^+ \mid D_i(x) \in (E_2 \uplus Z^+)\} \ .$$

We define for each $l \in (W^- \uplus (E_0 \cap E_1))$ the set of names in $X_i'^-$ linked to l:

$$X_i'^-(l) \triangleq \{x \in X_i'^- \mid A_i(x) = l\} \quad (i = 0, 1).$$

Now we must "bind together" names connected to the same link, so we create all the possible pairs between a name in $X_0'^-$ and a name in $X_1'^-$. Further we must add to $\hat{X}^-$ all the names in $X_i'^-$ "not associable" to any name of $X_{\bar{i}}'^-$. Then the set of downward names of $\vec{B}$ is:

$$\hat{X}^- \triangleq \bigcup_{l \in (W^- \uplus (E_0 \cap E_1))} X_0'^-(l) \times X_1'^-(l) \cup \sum_{i \in \{0,1\}} \bigcup_{e \in (E_i \setminus E_{\bar{i}})} X_i'^-(e).$$

Next, on the disjoint sum $X_0'^+ \oplus X_1'^+$, define $\cong$ to be the smallest equivalence for which $(0, x_0) \cong (1, x_1)$ iff there exists $p \in (W^+ \uplus (P_0 \cap P_1))$ such that $A_0(p) = x_0$ and $A_1(p) = x_1$. Then define:

$$\hat{X}^+ \triangleq (X_0'^+ \oplus X_1'^+)/_{\cong}.$$

For each $x \in X_i'^+$ we denote the equivalence class of (i, x) by $\widehat{i, x}$.

links Define the link maps of B_i as follows:

$$\text{for } x \in X_i^+ : B_i(x) \triangleq \begin{cases} D_i(x) & \text{if } x \in (X_i^+ \setminus X_i'^+) \\ \widehat{i, x} & \text{if } x \in X_i'^+; \end{cases}$$

$$\text{for } p \in (P_{\bar{i}} \setminus P_i) : B_i(p) \triangleq \begin{cases} D_i(p) & \text{if } A_{\bar{i}}(p) \notin X_{\bar{i}}^+ \\ \widehat{\bar{i}, x} & \text{if } A_{\bar{i}}(p) = x \in X_{\bar{i}}^+; \end{cases}$$

$$\text{for } \hat{x} \in \hat{X}^- : B_i(\hat{x}) \triangleq \begin{cases} x & \text{if } \hat{x} = (x, y) \text{ and } i = 0 \\ y & \text{if } \hat{x} = (x, y) \text{ and } i = 1 \\ \hat{x} & \text{if } \hat{x} \in (\hat{X}^- \cap X_i^-) \\ A_{\bar{i}}(\hat{x}) & \text{if } \hat{x} \in (\hat{X}^- \cap X_{\bar{i}}^-). \end{cases}$$

Finally we define the link map of B:

$$\text{for } \hat{x} \in \hat{X}^+ : B(\hat{x}) \triangleq D_i(x) \text{ where } \hat{x} = \widehat{i, x} \text{ and } x \in X_i^+;$$

$$\text{for } p \in (P_2 \uplus Z^-) : B(p) \triangleq \begin{cases} D_i(p) & \text{if } D_i(p) \in (E_2 \uplus Z^+) \\ D_{\bar{i}}(p) & \text{if } D_i(p) \in (E_{\bar{i}} \setminus E_i) \\ D_i(p) & \text{if } D_{\bar{i}}(p) \in (E_i \setminus E_{\bar{i}}) \\ (x, y) & \text{if } D_0(p) = x \in X_0^- \text{ and} \\ & D_1(p) = y \in X_1^-. \end{cases}$$

Theorem 3.2 *In* $'$DLG*, whenever a pair* $\vec{A}$ *of link graphs has a bound* $\vec{D}$*, there exists an RPO* $(\vec{B}, B)$ *for* $\vec{A}$ *to* $\vec{D}$*, and Construction 3.1 yields such an RPO.*

Proof. The proof is in two parts. First we have to check that $(\vec{B}, B)$ is an RPO candidate; this is done by long and tedious calculations.

Next, for any other candidate $(\vec{C}, C)$, we have to construct the unique arrow $\hat{C}$ such that the diagram aside commutes. This link graph $\hat{C}$ can be constructed as follows: let be V_C the nodes of C, for $i = 0, 1$ the set of nodes of C_i is $V_{C_i} \triangleq (V_{\bar{i}} \setminus V_i) \uplus V_3$, where V_3 is such that $V_2 = V_3 \uplus V_C$; edges E_{C_i} and ports P_{C_i} of C_i are defined analogously. Then $\hat{C}$ has V_3, E_3 and P_3 as sets of nodes, edges and ports respectively. Its link map is defined as follows:

$$\text{for } \widehat{j,x} \in \hat{X}^+ : \hat{C}(\widehat{j,x}) \triangleq C_j(x);$$

$$\text{for } p \in (P_3 \uplus \hat{Y}^-) : \ \hat{C}(p) \ \triangleq \begin{cases} C_i(p) & \text{if } C_i(p) \in (E_3 \uplus \hat{Y}^+) \\ C_0(p) & \text{if } C_0(p) \in (\hat{X}^- \cap X_0^-) \\ C_1(p) & \text{if } C_1(p) \in (\hat{X}^- \cap X_1^-) \\ (x,y) & \text{if } C_0(p) = x \in X_0^- \text{ and } C_1(p) = x \in X_1^-. \end{cases}$$

□

As an immediate consequence, we can calculate RPBs as well.

Corollary 3.3 *In* $'$DLG, *whenever a pair* $\vec{D} : \vec{X} \to W$ *of link graphs has a co-bound* $\vec{A} : Z \to \vec{X}$, *there exists an RPB* $(\vec{B} : \hat{X} \to \vec{X}, B : Z \to \hat{X})$ *for* $\vec{A}$ *to* $\vec{D}$, *and Construction 3.1 can be used for calculating such an RPB.*

Proof. Consider the pair $\overline{\vec{D}} : \overline{W} \to \overline{\vec{X}}$, which is in $'$DLG for Proposition 2.3; this pair has the bound $\overline{\vec{A}} : \overline{\vec{X}} \to \overline{Z}$, and hence, for Theorem 3.2, Construction 3.1 yields a RPO $(\vec{C} : \overline{\vec{X}} \to \overline{\hat{X}}, C : \overline{\hat{X}} \to \overline{Z})$. Then, take $\vec{B} \triangleq \overline{\vec{C}}$ and $B \triangleq \overline{C}$. □

Finally, one may wonder whether this construction can be used for calculating locally universal hexagons (luxes). Actually $'$DLG does not have all luxes, although it has RPOs and RPBs. In fact, it is easy to construct an hexagon such that its RPO and RPB do not commute; the result follows from [5, Theorem 1].

However, there are two subprecategories of $'$DLG where luxes do exist:

Proposition 3.4 *Let* $'$MDLG *be the wide subprecategory of* $'$DLG *of mono directed link graphs. Then,* $'$MDLG *has luxes, which can be constructed using Construction 3.1 twice.*

Proof. Follows from [5, Corollary 3], and previous results. □

Clearly, the result applies also to $'\mathrm{EDLG} = {}'\mathrm{MDLG}^{op}$, the subprecategory of $'$DLG of epi directed link graphs. Notice that, by Proposition 2.7, mono and epi link graphs are easy to recognize and single out.

3.2 *Construction of idem-relative pushouts*

We now proceed to characterise all the IPOs for a given pair $\vec{A} : W \to \vec{X}$ of link graphs. The first step is to establish consistency conditions.

Definition 3.5 *We define four* consistency conditions *on a pair* $\vec{A} : W \to \vec{X}$ *of link graphs.*

CDL0 $ctrl_0(v) = ctrl_1(v)$ *if* $v \in (V_0 \cap V_1)$;

CDL1 *if* $p \in (P_0 \cap P_1)$ *and* $A_i(p) \in ((E_0 \cap E_1) \uplus W^-)$, *then* $A_{\bar{i}}(p) = A_i(p)$;

CDL2 *if* $p_2 \in (P_0 \cap P_1)$ *and* $A_i(p_2) \in (E_i \setminus E_{\bar{i}})$, *then* $A_{\bar{i}}(p_2) = x_{\bar{i}}$ *for some* $x_{\bar{i}} \in X_{\bar{i}}^+$, *and further if* $A_{\bar{i}}(p) = A_{\bar{i}}(p_2)$ *then* $p \in (W^+ \uplus (P_0 \cap P_1))$ *and* $A_i(p) = A_i(p_2)$, *or* $p \in (P_{\bar{i}} \setminus P_i)$ *and exists* $x_i \in X_i^-$ *such that* $A_i(x_i) = A_i(p_2)$;

CDL3 *for each* $p \in (P_i \setminus P_{\bar{i}})$ *such that* $A_i(p) \in (W^- \uplus (E_0 \cap E_1))$, *then exists* $x_{\bar{i}} \in X^-_{\bar{i}}$ *such that* $A_{\bar{i}}(x_{\bar{i}}) = A_i(p)$.

Informally, CDL1 says that if a shared point p in A_i is linked to a shared link l, then in $A_{\bar{i}}$ the shared point p must be linked to the same link l. CDL2 says that if the link of a shared point p_2 in A_i is closed and unshared, then its link in $A_{\bar{i}}$ must be an outer upward name, further any peer p of p_2 in $A_{\bar{i}}$ must also be its peer in A_i, or if p is not shared, then in A_i there exists an outer downward name linked to the unshared edge of p_2. Finally, CDL3 says that if an unshared point in A_i is linked to a shared link, then in $A_{\bar{i}}$ there is an outer downward name linked to the shared link.

Proposition 3.6 *If a pair of link graphs* $\vec{A}$ *has a bound, then the consistency conditions hold.*

Now, assuming the consistency conditions of Definition 3.5, we shall construct a non-empty family of IPOs for $\vec{A}$ denoted by $IPO(\vec{A})$.

Construction 3.7 (IPOs in directed link graphs) Assume that the consistency conditions 3.5 hold for the pair $\vec{A} : W \to \vec{X}$ of link graphs. We define $\vec{C} : \vec{X} \to Y$ an IPO for $\vec{A}$ as follows:

nodes and edges Define the nodes of C_i to be $V_{C_i} \triangleq V_{\bar{i}} \setminus V_i$. Edges and ports of C_i are defined analogously.

interface For $i = 0, 1$ choose any subset L^+_i of the names X^+_i such that all members of L^+_i are idle. Define

$$\tilde{P}_i \triangleq \{p \in P_i \setminus P_{\bar{i}} \mid A_i(p) \in X^+_i \text{ and } \nexists p' \in (P_i \cap P_{\bar{i}}) \uplus W^+ \text{ s.t. } A_i(p) = A_i(p')\}$$

and choose $Q^+_i \subseteq A_i(\tilde{P}_i) \cap X^+_i$. Let $K^+_i \triangleq X^+_i \setminus (L^+_i \cup Q^+_i)$ and let $K'^+_i \subseteq K^+_i$ be the names to be mapped to the codomain Y^+. We define (for $i = 0, 1$):

$$X'^-_i \triangleq \{x \in X^-_i \mid \exists y \in X^-_{\bar{i}} \text{ s.t. } A_i(x) = A_{\bar{i}}(y) \text{ or } A_i(x) \in (E_i \setminus E_{\bar{i}})\}$$
$$K'^+_i \triangleq \{x \in K^+_i \mid \forall p \in (W^+ \uplus (P_0 \cap P_1)).A_i(p) = x \in X^+_i \Rightarrow A_{\bar{i}}(p) \in X^+_{\bar{i}}\}\ .$$

As in Construction 3.1, we define for each $l \in (W^- \uplus (E_0 \cap E_1))$ the set $X'^-_i(l)$ of names linked to l, and define:

$$Y^- \triangleq \bigcup_{l \in (W^- \uplus (E_0 \cap E_1))} X'^-_0(l) \times X'^-_1(l) \cup \sum_{i \in \{0,1\}} \bigcup_{e \in (E_i \setminus E_{\bar{i}})} X'^-_i(e).$$

Next, on the disjoint sum $K'^+_0 \oplus K'^+_1$, define $\simeq$ to be the smallest equivalence for which $(0, x_0) \simeq (1, x_1)$ iff there exists $p \in (W^+ \uplus (P_0 \cap P_1))$ such that $A_0(p) = x_0$ and $A_1(p) = x_1$. Then define:

$$Y^+ \triangleq (K'^+_0 \oplus K'^+_1)/_{\simeq}.$$

For each $x \in K'^+_i$ we denote the equivalence class of (i, x) by $\widehat{i, x}$.

links For $i = 0, 1$, choose three arbitrary functions:

$$\eta_i : L_i^+ \to E_{\bar{i}} \setminus E_i;$$
$$\xi_i : Q_i^+ \to \{e \in E_i \setminus E_{\bar{i}} \mid \exists x \in X_i^- \text{ s.t. } A_i(x) = e\};$$

and for each $l \in (W^- \uplus (E_0 \cap E_1))$ for which there exists $x_i \in X_i^-$ and $p \in (P_{\bar{i}} \setminus P_i)$ such that $A_i(x_i) = l$ and $A_{\bar{i}}(p) = l$, choose an arbitrary function:

$$\theta_i^l : \{p \in (P_{\bar{i}} \setminus P_i) \mid A_{\bar{i}}(p) = l\} \to X_i'^-(l).$$

Then define the link maps $C_i : X_i \to Y$ as follows:

$$\text{for } x \in X_i^+ : C_i(x) \triangleq \begin{cases} A_{\bar{i}}(p) & \text{if } x \in (K_i^+ \setminus K_i'^+) \text{, then} \\ & \exists p \in (W^+ \uplus (P_0 \cap P_1)) \text{ s.t. } A_i(p) = x \\ \widehat{i, x} & \text{if } x \in K_i'^+ \\ \eta_i(x) & \text{if } x \in L_i^+ \\ \xi_i(x) & \text{if } x \in Q_i^+; \end{cases}$$

$$\text{for } p \in (P_{\bar{i}} \setminus P_i) : C_i(p) \triangleq \begin{cases} A_{\bar{i}}(p) & \text{if } A_{\bar{i}}(p) \in (E_{\bar{i}} \setminus E_i) \\ \widehat{\bar{i}, x} & \text{if } A_{\bar{i}}(p) = x \in X_{\bar{i}}^+ \setminus Q_{\bar{i}}^+ \\ \theta_i^l(p) & \text{if } A_{\bar{i}}(p) = l \in ((E_0 \cap E_1) \uplus W^-) \\ \theta_i^e(p) & \text{if } p \in \tilde{P}_{\bar{i}} \text{ and } e = \xi_{\bar{i}}(A_{\bar{i}}(p)); \end{cases}$$

$$\text{for } y \in Y^- : C_i(y) \triangleq \begin{cases} x & \text{if } \hat{x} = (x, y) \text{ and } i = 0 \\ y & \text{if } \hat{x} = (x, y) \text{ and } i = 1 \\ y & \text{if } y \in (Y^- \cap X_i^-) \\ A_{\bar{i}}(y) & \text{if } y \in (Y^- \cap X_{\bar{i}}^-). \end{cases}$$

The maps η_i are called *elision*; this refers to the fact that the idle names L_i^+ in A_i are not exported in the IPO interface Y, but instead mapped into C_i.

The maps ξ_i are called *inversion*; this refers to the fact that in the bound $C_{\bar{i}}$ of $A_{\bar{i}}$ we can invert the direction of some link from upward to downward. In this way we can connect a port p of $P_i \setminus P_{\bar{i}}$ to an edge e in $E_{\bar{i}} \setminus E_i$ also when there is no shared port, connected to the same name of p, which is linked to e in $A_{\bar{i}}$.

The maps θ_i^l are called *random link*; this refers to the fact that if a link has more then one name linked to it, then in the bound it is indifferent to which name a point is linked to, because the effect of composition is the same.

There is a distinct IPO for each choice of L_i^+, Q_i^+, η_i, ξ_i and θ_i^l. When $\vec{A}$ are both epi then there are no elisions of idle names and there not exists two different names in X_i^- that are peers, then the IPO is unique and hence a pushout.

Theorem 3.8 *A pair $\vec{C} : \vec{X} \to Y$ is an IPO for $\vec{A} : W \to \vec{X}$ iff it is generated (up to isomorphism) by Construction 3.7.*

Proof. ($\Rightarrow$) $\vec{B}$ is an IPO for $\vec{A}$ iff it is the legs of an RPO w.r.t. some bound $\vec{D}$. So we can assume w.l.g. that $\vec{B}$ is generated by Construction 3.1. Now apply

Construction 3.7 to create $\vec{C}$ by choosing $\vec{L^+}$, $\vec{Q^+}$, $\vec{\eta}$, $\vec{\xi}$ and $\vec{\theta^l}$ as in $\vec{D}$. Then $\vec{C}$ coincides with $\vec{B}$.

($\Leftarrow$) Consider any $\vec{C}$ generated by Construction 3.7. Now apply the Construction 3.1 to yield an RPO $(\vec{B}, B)$ for $\vec{A}$ to $\vec{C}$. Then $\vec{B}$ coincides with $\vec{C}$. □

4 Embedding output-linear and input-linear link graphs in directed link graphs

In this section, we show how the previous theories of output-linear and input-linear bigraphs are related to directed link graphs.

Let us first recall the definition of bigraphs, as given by Milner [10]. For clarity, we add the adjective "output linear".

Definition 4.1 *An* output-linear link graph *is a tuple* $A = (V, E, ctrl, link) : X \to Y$*, where* V *is the set of* nodes*,* E *is the set of* edges*, and* X *and* Y *are the sets of* inner *and* outer *names, respectively;* $ctrl : V \to \mathcal{K}$ *is the* control map*, and finally* $link : P \uplus X \to E \uplus Y$ *is the* link map*, where* $P \triangleq \sum_{v\in V} ar(ctrl(v))$ *is the set of* ports *of* A*. Inner names and ports are the* points*, while outer names and edges are the* links*.*

The support *of the output-linear link graph* A *is the set* $|A| \triangleq V \oplus E$*.*

Then, we recall the definition of the category of output-linear link graphs (cf. [4, Def. 8.3], there called ʹLIG):

Definition 4.2 **(ʹOLG)** *The precategory of* output-linear link graphs ʹOLG *has sets of names as objects, and output-linear link graphs as morphisms. Composition of two link graphs* A_0, A_1 *is defined when their supports are disjoint; in this case,* $A_1 \circ A_0 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link)$*, where* $link : P \uplus X_0 \to E \uplus X_2$ *(where* $P = P_0 \uplus P_1$*) is defined as follows:*

$$link(p) \triangleq \begin{cases} link_0(p) & \text{if } p \in X_0 \uplus P_0 \text{ and } link_0(p) \in E_0 \\ link_1(x) & \text{if } p \in X_0 \uplus P_0 \text{ and } link_0(p) = x \in X_1 \\ link_1(p) & \text{if } p \in P_1. \end{cases}$$

The identity link graph at X *is* $id_X \triangleq (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, Id_X) : X \to X$*.*

The precategory ʹOLG is well-supported; actually it is a well-supported monoidal precategory. See [4] for details. Moreover, whenever a span $\vec{A}$ in ʹOLG has a bound $\vec{D}$, there exists an RPO for $(\vec{A}, \vec{D})$; see [4, Construction 8.8].

The precategory ʹILG of input-linear link graphs is defined much like ʹOLG, just by swapping the input and output interfaces in the arity of the link functions (i.e., for $A : X \to Y$, its link map is $link : P \uplus Y \to E \uplus X$). The composition has to be changed accordingly: given two input-linear link graphs $A_0 : X_0 \to X_1, A_1 : X_1 \to X_2$ the composition $A_1 \circ A_0$ is defined when their supports are disjoint; in this case, $A_1 \circ A_0 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link)$, where $link : P \uplus X_2 \to E \uplus X_0$

(where $P = P_0 \uplus P_1$) is defined as follows:

$$link(p) \triangleq \begin{cases} link_1(p) & \text{if } p \in X_2 \uplus P_1 \text{ and } link_1(p) \in E_1 \\ link_0(x) & \text{if } p \in X_2 \uplus P_1 \text{ and } link_1(p) = x \in X_1 \\ link_0(p) & \text{if } p \in P_0. \end{cases}$$

It is immediate to see that an output-linear link graph $(V, E, ctrl, link : P \uplus X \to E \uplus Y) : X \to Y$ is also an input-linear link graph $(V, E, ctrl, link : P \uplus X \to E \uplus Y) : Y \to X$, and vice versa. Thus:

Proposition 4.3 *$'\mathrm{OLG} \cong {'\mathrm{ILG}}^{op}$.*

Corollary 4.4 *Let $\vec{A}$ be a span in $'\mathrm{OLG}$, with a bound $\vec{D}$. A triple $(\vec{B}, B)$ is an RPO for $(\vec{A}, \vec{D})$ in $'\mathrm{OLG}$ iff $(\vec{B}^{op}, B^{op})$ is an RPB for $(\vec{A}^{op}, \vec{D}^{op})$ in $'\mathrm{ILG}$.*

As a consequence, the RPO construction in $'\mathrm{OLG}$ can be used for constructing RPBs in $'\mathrm{ILG}$, but it does not work for constructing RPOs. On the converse, an RPOs construction in $'\mathrm{ILG}$ would give an RPB construction in $'\mathrm{OLG}$ for free.

Actually, an RPO construction in $'\mathrm{ILG}$ can be recovered by noticing that input-linear link graphs correspond to input-linear cospans over a certain adhesive category **LGraph** of hypergraphs, as observed in [12]. Thus we can apply the general (G)RPO construction presented in *loc. cit.* (and fully detailed in [14]). In this paper, for a more direct comparison with the constructions in $'\mathrm{DLG}$ and $'\mathrm{OLG}$ (and in order to avoid to introduce 2-categorical machinery), we present a version of this construction tailored to the specific precategory $'\mathrm{ILG}$.

Construction 4.5 (RPOs in input-linear link graphs) An RPO in $'\mathrm{ILG}$ is built as follows:

nodes and edges If V_i are the nodes of A_i $(i = 0, 1)$; then the nodes of D_i are $V_{D_i} = (V_{\bar{i}} \setminus V_i) \uplus V_2$ for some V_2. Define the nodes of B_i and B to be $V_{B_i} \triangleq V_{\bar{i}} \setminus V_i$ $(i = 0, 1)$ and $V_B \triangleq V_2$. Edges E_i and ports P_i of A_i are treated analogously.

interface Construct the shared codomain $\hat{X}$ of $\vec{B}$ as follows: first we define the names in each X_i, for $i = 0, 1$, that must be mapped into $\hat{X}$:

$$X_i' \triangleq \{x \in X_i \mid \exists y \in X_{\bar{i}} \text{ s.t. } A_i(x) = A_{\bar{i}}(y) \text{ or } A_i(x) \in (E_i \setminus E_{\bar{i}})\}\ .$$

We define for each $l \in (W \uplus (E_0 \cap E_1))$ the set of names in X_i' linked to l:

$$X_i'(l) \triangleq \{x \in X_i' \mid A_i(x) = l\}\ (i = 0, 1).$$

Now we must "bind together" names connected to the same link, so we create all the possible pairs between a name in X_0' and a name in X_1'. Further we must add to $\hat{X}$ all the names of X_i' "not associable" to any name of $X_{\bar{i}}'$. Then the set of outer names of $\vec{B}$ is:

$$\hat{X} \triangleq \bigcup_{l \in (W \uplus (E_0 \cap E_1))} X_0'(l) \times X_1'(l) \cup \sum_{i \in \{0,1\}} \bigcup_{e \in (E_i \setminus E_{\bar{i}})} X_i'(e).$$

links Define B_i as follows:

for $p \in (P_{\bar{i}} \setminus P_i) : B_i(p) \triangleq D_i(p)$;

$$\text{for } \hat{x} \in \hat{X} : B_i(\hat{x}) \triangleq \begin{cases} x & \text{if } \hat{x} = (x, y) \text{ and } i = 0 \\ y & \text{if } \hat{x} = (x, y) \text{ and } i = 1 \\ \hat{x} & \text{if } \hat{x} \in (\hat{X} \cap X_i) \\ A_{\bar{i}}(\hat{x}) & \text{if } \hat{x} \in (\hat{X} \cap X_{\bar{i}}). \end{cases}$$

Finally we define B:

$$\text{for } p \in (Z \uplus P_2) : B(p) \triangleq \begin{cases} D_i(p) & \text{if } D_i(p) \in E_2 \\ D_{\bar{i}}(p) & \text{if } D_i(p) \in (E_{\bar{i}} \setminus E_i) \\ D_i(p) & \text{if } D_{\bar{i}}(p) \in (E_i \setminus E_{\bar{i}}) \\ (x, y) & \text{if } D_0(p) = x \text{ and } D_1(p) = y. \end{cases}$$

Proposition 4.6 *If a span $\vec{A}$ in* $'$ILG *has a bound $\vec{D}$, then there exists an RPO for $(\vec{A}, \vec{D})$, which is constructed by Construction 4.5.*

Actually, both constructions in $'$ILG and $'$OLG are special cases of Construction 3.1 in $'$DLG, since the former precategories are embedded in the latter:

Proposition 4.7 $'$ILG *and* $'$OLG *are equivalent to two well-supported monoidal subprecategories of* $'$DLG.

Proof. The monoidal embeddings $F_I : {}'\mathrm{ILG} \to {}'\mathrm{DLG}$ and $F_O : {}'\mathrm{OLG} \to {}'\mathrm{DLG}$ are defined as obvious: on objects, $F_I(X) = (X, \emptyset)$ and $F_O(X) = (\emptyset, X)$; on morphisms, simply as $F_I(A) = F_O(A) = A$. It is easy to check that these are two faithful functors, respecting supports and the monoidal operations. □

Proposition 4.8 *Given a span $\vec{A}$ with a bound $\vec{D}$ in* $'$ILG, *a triple $(\vec{B}, B)$ is an RPO for $(\vec{A}, \vec{D})$ iff $(F_I(\vec{B}), F_I(B))$ is an RPO for $(F_I(\vec{A}), F_I(\vec{D}))$ in* $'$DLG.

Thus, in order to calculate an RPO for a square $(\vec{A}, \vec{D})$ in either $'$ILG or $'$OLG, we just embed the square in $'$DLG, apply Construction 3.1, and drop the empty sets from the interfaces of the resulting RPO.

This result, in virtue of the self-duality of $'$DLG, extends to RPB as well, thus we have an algorithm for calculating RPBs in $'$ILG and $'$OLG.

Proposition 4.9 *Let $\vec{A}$ be a span with a bound $\vec{D}$ in* $'$ILG *or* $'$OLG; *then there exists an RPB $(\vec{B}, B)$ for $(\vec{A}, \vec{D})$.*

Proof. Consider the square $(\overline{F_I(\vec{D})}, \overline{F_I(\vec{A})})$ in $'$DLG. By applying Construction 3.1, we get an RPO $(\vec{C}, C)$ for it. Then, the RPB for $(\vec{A}, \vec{D})$ is obtained by taking $(\overline{\vec{C}}, \overline{C})$, and cancelling the empty sets from the interfaces. □

Luxes

Since $'$OLG and $'$ILG have RPOs and RPBs, one may wonder whether we can build luxes as well. The answer is the same of $'$DLG: not always. In both categories we can build hexagons which do not have luxes. However, if we restrict to either all epi or all mono link graphs, luxes do exist, and can be calculated by embedding an hexagon in ′DLG and applying Proposition 3.4.

IPOs

Also the consistency conditions and the construction of IPOs in ′OLG and ′ILG are subsumed by Definition 3.5 and Construction 3.7. Let us recall the consistency condition for output-linear graphs as in [4, Definition 8.10].

Definition 4.10 *The three* consistency conditions *on a pair $\vec{A} : W \to \vec{X}$ of output-linear link graphs are the following:*

COL0 *$ctrl_0(v) = ctrl_1(v)$ if $v \in (V_0 \cap V_1)$;*

COL1 *if $A_i(p) \in (E_0 \cap E_1)$, then $p \in (W \uplus (P_0 \cap P_1))$ and $A_{\bar{i}}(p) = A_i(p)$;*

COL2 *for $p_2 \in (P_0 \cap P_1)$, if $A_i(p_2) \in (E_i \setminus E_{\bar{i}})$ then $A_{\bar{i}}(p_2) \in X_{\bar{i}}$, and if also $A_{\bar{i}}(p) = A_{\bar{i}}(p_2)$ then $p \in (W \uplus (P_0 \cap P_1))$ and $A_i(p) = A_i(p_2)$.*

On the other hand, the consistency conditions for input-linear graphs are quite different:

Definition 4.11 *The three* consistency conditions *on a pair $\vec{A} : W \to \vec{X}$ of input-linear link graphs are the following:*

CIL0 *$ctrl_0(v) = ctrl_1(v)$ if $v \in (V_0 \cap V_1)$;*

CIL1 *if $p \in (P_0 \cap P_1)$, then $A_0(p) = A_1(p)$;*

CIL2 *for $p \in (P_i \setminus P_{\bar{i}})$ such that $A_i(p) \in (W \uplus (E_0 \cap E_1))$, there exists $x_{\bar{i}} \in X_{\bar{i}}$ such that $A_{\bar{i}}(x_{\bar{i}}) = A_i(p)$.*

In both precategories, if a bound satisfies the relevant consistency conditions, its IPOs can be calculated using Construction 3.7, in virtue of the following result:

Proposition 4.12 *Let $\vec{A}$ a span in ′OLG. If $\vec{A}$ satisfy the conditions in Definition 4.10 then $IPO(\vec{A}) \cong IPO(F_O(\vec{A}))$.*

Let $\vec{A}$ a span in ′ILG. If $\vec{A}$ satisfy the conditions in Definition 4.11 then $IPO(\vec{A}) \cong IPO(F_I(\vec{A}))$.

Thus, we have automatically an algorithm for calculating IPOs for a span of input-linear link graphs $\vec{A}$: just apply Construction 3.7 to $F_I(\vec{A})$ and drop the empty sets from the interfaces of the IPOs so obtained. As far as we know, these are the first consistency conditions and IPO construction for input-linear link graphs, which we give here for sake of completeness.

Construction 4.13 (IPOs in input-linear link graphs) Assume that the consistency conditions 4.11 hold for the pair $\vec{A} : W \to \vec{X}$ of link graphs. We define $\vec{C} : \vec{X} \to Y$ an IPO for $\vec{A}$ as follows:

nodes and edges Define the nodes of C_i to be $V_{C_i} \triangleq V_{\bar{i}} \setminus V_i$. Edges and ports of C_i are defined analogously.

interface As in Construction 4.5, build the shared codomain Y of $\vec{C}$ as follows:

$$X'_i \triangleq \{x \in X_i \mid \exists y \in X_{\bar{i}} \text{ s.t. } A_i(x) = A_{\bar{i}}(y) \text{ or } A_i(x) \in (E_i \setminus E_{\bar{i}})\} \quad (i = 0, 1)$$

$$X'_i(l) \triangleq \{x \in X'_i \mid A_i(x) = l\} \quad (i = 0, 1)$$

$$Y \triangleq \bigcup_{l \in (W \uplus E_2)} X'_0(l) \times X'_1(l) \cup \sum_{i \in \{0,1\}} \bigcup_{e \in (E_i \setminus E_{\bar{i}})} X'_i(e).$$

links For $i = 0, 1$ and for each $l \in (W \uplus E_2)$ for which there exists $x_i \in X_i$ and $p \in (P_{\bar{i}} \setminus P_i)$ such that $A_i(x_i) = l$ and $A_{\bar{i}}(p) = l$, choose an arbitrary function:

$$\theta_i^l : \{p \in (P_{\bar{i}} \setminus P_i) \mid A_{\bar{i}}(p) = l\} \to X'_i(l).$$

Then define the link maps $C_i : X_i \to Y$ as follows:

$$\text{for } p \in (P_{\bar{i}} \setminus P_i) : C_i(p) \triangleq \begin{cases} A_{\bar{i}}(p) & \text{if } A_{\bar{i}}(p) \in (E_{\bar{i}} \setminus E_i) \\ \theta_i^l(p) & \text{if } A_{\bar{i}}(p) = l \in (W \uplus E_2); \end{cases}$$

$$\text{for } y \in Y : C_i(y) \triangleq \begin{cases} x_0 & \text{if } y = (x_0, x_1) \text{ and } i = 0 \\ x_1 & \text{if } y = (x_0, x_1) \text{ and } i = 1 \\ y & \text{if } y \in (Y \cap X_i) \\ A_{\bar{i}}(y) & \text{if } y \in (Y \cap X_{\bar{i}}). \end{cases}$$

5 Conclusions

In this paper, we have presented the *directed bigraphs*, whose connection graphs, called *directed link graphs*, generalize both output-linear (i.e., Milner's) and input-linear (i.e., Sassone-Sobociński's) link graphs. We have given a constructions of RPOs generalizing and unifying the known constructions in the previous theories. Moreover, the RPO construction can be used for calculating RPBs as well, and, in suitable subcategories, also luxes. We have proposed new consistency conditions for the existence of IPOs, and a general construction of IPOs, in directed link graphs. These conditions and construction subsume those proposed for Milner's bigraphs; moreover, these have been specialized to the input-linear case yielding the first consistency conditions and IPO construction for this variant.

Due to lack of space, we cannot present here the algebraic theory of directed bigraphs, and in particular the elementary constructors and a normal form for 'DBig. This theory will be useful for representing calculi in this framework; for instance, the λ-calculus (among others) can be conveniently represented in directed bigraphs without the need of further notions for dealing with binders (as it happens, instead, with Milner's binding bigraphs). We refer the interested reader to [1,2].

Future work We plan to use directed bigraphs for representing some calculi of interest, in particular calculi with resources, locations, etc., which can be represented by edges. It will be interesting to see which kind of wide transition system we would obtain from our theory, in these cases.

Another future work is to move the theory of directed link graphs into the realm of groupoidal 2-categories. Actually, due to their intrinsic bi-directional linearity, representing directed link graphs simply as input-linear cospans in some adhesive G-category does not seem feasible. We suppose that a generalization of input-linear cospans, and the corresponding GRPO construction, will be required to this end.

Acknowledgements The authors wish to thank Robin Milner and the anonymous referees for their helpful comments.

References

[1] Grohmann, D. and M. Miculan, *Directed bigraphs: theory and applications*, Technical Report UDMI/12/2006/RR, University of Udine (2006), at http://www.dimi.uniud.it/miculan/Papers/.

[2] Grohmann, D. and M. Miculan, *An algebra for directed bigraphs*, in: I. Mackie and D. Plump, editors, *Pre-proceedings of TERMGRAPH 2007*, ENTCS (2007), to appear.

[3] Jensen, O. H. and R. Milner, *Bigraphs and transitions*, in: *Proc. POPL*, 2003, pp. 38–49.

[4] Jensen, O. H. and R. Milner, *Bigraphs and mobile processes (revised)*, Technical report UCAM-CL-TR-580, Computer Laboratory, University of Cambridge (2004).

[5] Klin, B., V. Sassone and P. Sobociński, *Labels from reductions: Towards a general theory*, in: *Proc. CALCO*, Lecture Notes in Computer Science **3629** (2005), pp. 30–50.

[6] Lack, S. and P. Sobociński, *Adhesive categories*, in: I. Walukiewicz, editor, *Proc. FoSSaCS*, Lecture Notes in Computer Science **2987** (2004), pp. 273–288.

[7] Leifer, J. J. and R. Milner, *Deriving bisimulation congruences for reactive systems*, in: C. Palamidessi, editor, *Proc. CONCUR*, Lecture Notes in Computer Science **1877** (2000), pp. 243–258.

[8] Leifer, J. J. and R. Milner, *Transition systems, link graphs and petri nets*, Mathematical Structures in Computer Science **16** (2006), pp. 989–1047.

[9] Milner, R., *Bigraphical reactive systems*, in: K. G. Larsen and M. Nielsen, editors, *Proc. 12th CONCUR*, Lecture Notes in Computer Science **2154** (2001), pp. 16–35.

[10] Milner, R., *Pure bigraphs: Structure and dynamics*, Inf. Comput. **204** (2006), pp. 60–122.

[11] Sassone, V. and P. Sobociński, *Deriving bisimulation congruences: A 2-categorical approach*, in: *Proc. EXPRESS'02*, Electr. Notes Theor. Comput. Sci. **68(2)**, 2002.

[12] Sassone, V. and P. Sobociński, *Reactive systems over cospans*, in: *Proc. LICS* (2005), pp. 311–320.

[13] Sewell, P., *From rewrite rules to bisimulation congruences*, Theor. Comput. Sci. **274** (2002), 183–230.

[14] Sobociński, P., "Deriving process congruences from reaction rules," Ph.D. thesis, BRICS, University of Aarhus (2004).