Research Report UDMI/12/2006/RR

Directed Bigraphs: Theory and Applications*

Davide Grohmann Marino Miculan
Department of Mathematics and Computer Science, University of Udine, Italy
grohmann@dimi.uniud.it miculan@dimi.uniud.it

December 29, 2006

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*.

For this model we give RPO and IPO constructions, 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.

We study also the algebraic structure of directed bigraphs: we give a sound and complete axiomatization of the (pre)category of directed bigraphs. Moreover, we use this axiomatization for encoding the λ-calculus, both in call-by-name and call-by-value variants.

*Work supported by Italian MIUR project 2005015824 Art.

Contents

1 Introduction

The fundamental importance of *labelled transition systems* (LTS) for defining the dynamics of a calculus is well known. Many fundamental notions and techniques, such as "bisimulations" and "model checking", rely on this kind of model. The key feature (and advantage) of an LTS semantics is that it is *compositional:* the behaviour of a system is explained in terms of the behaviour of its components, usually by a syntax-directed set of rules.

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 (compare, e.g., the LTSs of CCS, π-calculus and Ambients [8, 11, 7]).

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 [16] argued that the labels c of transitions 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) [6]. The notion of RPO has been later generalized to *groupoidal* RPO for dealing with syntactic congruences [14], 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) [4].

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 [2]. 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 [15] a general approach for constructing RPOs in a wide range of models, namely those which can be ex-

pressed as input-linear cospans over *adhesive* categories [5]. 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. The Sassone-Sobociński construction to the category of input-linear cospans over **Graph** yields (and generalizes) Ehrig and König's *borrowed-context rewriting* [1]. However, 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 [15].

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-Sobociński'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.

The key idea of directed link graphs is to notice that in link graphs names are not resources on their own, but only a way for denoting (abstract) resources, 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 points (i.e., control ports), goes through names and eventually 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 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 inconsistent situations, however, we must take care of the "polarity" of names in interfaces, according as their meaning flows "upward" or "downward"—hence the adjective *directed*.

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.

Like Milner's bigraphs, also for the precategory of directed bigraphs a notion of normal form can be given, together with a complete axiomatization. This is very useful for encoding specific calculi, as we will see in the case of λ-calculus.

Synopsis 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, and in Section 4 we show how input-linear and output-linear link graphs are subsumed by directed link graphs.

In Section 5 we analyze the algebraic structure of the precategory ′DBIG, giving a sound and complete axiomatization of this (pre)category. This analysis is then carried on to the category DBIG in Section 6.

In Section 7 we put directed bigraphs at work, giving the encodings of λ-calculus, both in call-by-name and call-by-value variants. Notably, we do not need to introduce further extensions (such as binding signatures) to this end; thus, directed bigraphs turn out to be more expressive than the two previous variants.

Conclusions and direction for future work are in Section 8.

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 [3, §3] 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 sets of names* $X = (X^-, X^+)$*; the two components are called* downward *and* upward *interfaces, 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, 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 explicit objects, and not 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.

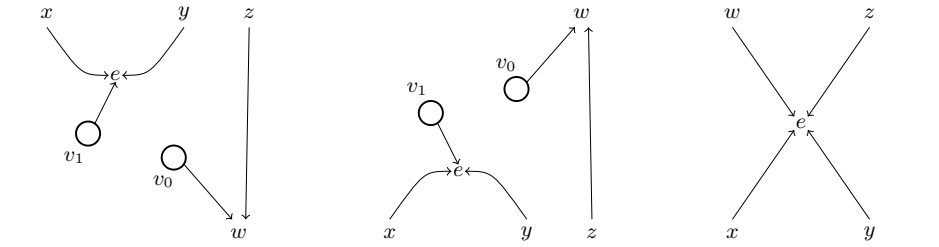


Figure 1: Examples of directed link graphs.

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.

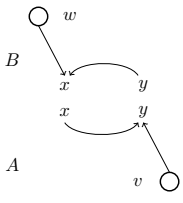
In the following, by "interface" and "link graphs" we will intend "polarized interface" and "directed link graphs" respectively, unless otherwise noted.

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 would yield a "loop" and hence not a directed link graph.

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

Proof. 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 $'$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 $'$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 [3, §7] for definitions) and directed link graphs.

Definition 2.10 *A* directed bigraph *with signature $\mathcal{K}$ is $G = (V, E, ctrl, prnt, link) : I \to J$, where $I = \langle m, X \rangle$ and $J = \langle n, Y \rangle$ are its inner and outer interfaces respectively. An interface is composed by a* width *(a finite ordinal) and by a pair of finite sets of names (from a global set $\mathcal{X}$). 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:*

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

It is very simple to check the following proposition.

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 [3, §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.

Actually, in many situations we do not want to distinguish bigraphs differing only on the identity of nodes and edges. To this end, we introduce the category DBIG of *abstract directed bigraphs*. The category DBIG is constructed from 'DBIG forgetting the identity of nodes and edges and any idle edge. More precisely, abstract bigraphs are bigraphs taken up-to an equivalence $\eqslantless$ (see [3] for details).

Definition 2.15 (abstract directed bigraphs) *Two concrete directed bigraphs G and H are* lean-support equivalent*, written $G \eqslantless H$, if they are support equivalent after removing any idle edges.*

The category DBIG *of abstract directed bigraphs has the same objects as* 'DBIG*, and its arrows are lean-support equivalence classes of directed bigraphs. We denote by $\mathcal{A} :'$ DBIG $\to$ DBIG the associated quotient functor.*

We remark that DBIG is a category (not only a precategory); moreover, $\mathcal{A}$ enjoys several important properties; see [3].

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:

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

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

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

Finally we define the link map of B:

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

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

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 cobound* $\vec{A} : Z \to \vec{X}$*, 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 [4, 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 [4, Corollary 3], and previous results. Clearly, the result applies also to $'$EDLG $='$ 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 the consistency conditions 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 $Q_i^+ \subseteq A_i(\tilde{P}_i)$. Let be $K_i^+ = X_i^+ \setminus (L_i^+ \cup Q_i^+)$, define $K_i'^+ \subseteq K_i^+$, the names to be mapped to the codomain Y^+. Then 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:

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

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:

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

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

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

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 refer to the fact that in the bound $C_{\bar{i}}$ of $A_{\bar{i}}$ we can invert the verse 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 $E_{\bar{i}} \setminus E_i$ without any explicit motivation, i.e. there is no shared port connected to the same name of p that in $A_{\bar{i}}$ is linked to e.

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. [3, Def. 8.3], there called 'LIG):

Definition 4.2 ('OLG) *The precategory of* output-linear link graphs 'OLG *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 'OLG is well-supported; actually it is a well-supported monoidal precategory. See [3] for details. Moreover, whenever a span $\vec{A}$ in 'OLG has a bound $\vec{D}$, there exists an RPO for $(\vec{A}, \vec{D})$; see [3, Construction 8.8].

The precategory 'ILG of input-linear link graphs is defined much like 'OLG, 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 $'$OLG, with a bound $\vec{D}$. A triple $(\vec{B}, B)$ is an RPO for $(\vec{A}, \vec{D})$ in $'$OLG iff $(\vec{B}^{op}, B^{op})$ is an RPB for $(\vec{A}^{op}, \vec{D}^{op})$ in $'$ILG.*

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

Actually, an RPO construction in $'$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 [15]. Thus we can apply the general (G)RPO construction presented in *loc. cit.* (and fully detailed in [17]). In this paper, for a more direct comparison with the constructions in $'$DLG and $'$OLG (and in order to avoid to introduce 2-categorical machinery), we present a version of this construction tailored to the specific precategory $'$ILG.

Construction 4.5 (RPOs in input-linear link graphs) An RPO in $'$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 downward names of $\vec{B}$ is:

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

links Define B_i as follows:

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

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

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 $F_I :' \text{ILG} \to' \text{DLG}$ and $F_O :' \text{OLG} \to' \text{DLG}$ 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 [3, 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.

5 Algebraic structure of 'DBIG

We begin this section introducing some useful notations.

Remark. We abbreviate an interface $\langle 0, (X^-, X^+)\rangle$ simply by (X^-, X^+), and a set like $\{x\}$ by x; in similar way we write m for $\langle m, (\emptyset, \emptyset)\rangle$. The two interfaces $(\emptyset, \emptyset)$ and 0 are equal and stand for the origin ϵ, so the identity id_ϵ can be expressed as ϵ, $(\emptyset, \emptyset)$ or 0.

A bigraph $A : (\emptyset, X^+) \to (\emptyset, Y^+)$ is defined by a (not necessarily surjective) function $\sigma : X^+ \to Y^+$, called *substitution*, if it has no nodes and no edges and the link map is σ; analogously a bigraph $A : (X^-, \emptyset) \to (Y^-, \emptyset)$ is defined by a (not necessarily surjective) function $\delta : Y^- \to X^-$, called *fusion*, if it has no nodes and no edges and the link map is δ. With abuse of notation, we write σ and δ to mean their corresponding bigraphs.

Let two vectors $\vec{x}$ and $\vec{y}$ of the same length; we write $(y_0/x_0, y_1/x_1, \dots)$ or simply $\triangle^{\vec{y}}_{\vec{x}}$, where all the x_i are distinct, for the surjective substitution $x_i \mapsto y_i$; similarly, we write $(y_0/x_0, y_1/x_1, \dots)$ or $\nabla^{\vec{y}}_{\vec{x}}$, where all y_i are distinct, for the surjective substitution $y_i \mapsto x_i$.

We denote by $\triangle^X : (\emptyset, \emptyset) \to (\emptyset, X)$ the bigraph defined by the empty substitution $\sigma : \emptyset \to X$, in the same way we denote $\nabla_X : (X, \emptyset) \to (\emptyset, \emptyset)$ for the bigraph defined by the empty fusion $\delta : \emptyset \to X$.

Note that each substitution σ can be expressed in a unique way as $\sigma = \tau \otimes \triangle^X$, where τ is a surjective substitution; while each fusion δ can be expressed in a unique way as $\delta = \zeta \otimes \nabla_X$, where ζ is a surjective fusion. We denote the renamings by α, i.e. the bijective substitution or bijective fusion.

Finally, we introduce the *closure bigraphs*. The *closure* $\blacktriangleright^x_y : (\emptyset, y) \to (x, \emptyset)$ has no nodes, a unique edge e and the link map is $link(x) = e = link(y)$. Two other types of closures are obtained by composing the closure $\blacktriangleright^x_y$ and $\triangle^x$ or ∇_y respectively:

- the *up-closure* $\blacktriangle^y : (\emptyset, y) \to (\emptyset, \emptyset)$ has no nodes, a unique edge e and $link(y) = e$;
- the *down-closure* $\blacktriangledown_x : (\emptyset, \emptyset) \to (x, \emptyset)$ has no nodes, a unique edge e and $link(x)$=e.

Definition 5.1 (wirings) *A* wiring *is a bigraph whose interfaces have zero width (and hence has no nodes). The wirings ω are generated by the composition or tensor product of three base elements: the substitutions $\sigma : (\emptyset, X^+) \to (\emptyset, Y^+)$; the fusions $\delta : (Y^-, \emptyset) \to (X^-, \emptyset)$; and the closures $\blacktriangleright^x_y : (\emptyset, y) \to (x, \emptyset)$.*

Definition 5.2 (prime bigraph) *An interface is* prime *if it has width equal to 1. Often we abbreviate a prime interface $\langle 1, (X^-, X^+)\rangle$ with $\langle (X^-, X^+)\rangle$, in particular $\langle (\emptyset, \emptyset)\rangle = 1$. A prime bigraph $P : \langle m, (Y^-, \emptyset)\rangle \to \langle (\emptyset, X^+)\rangle$ has no upward inner names and no downward outer names, and has a prime outer interface.*

An important prime bigraph is $merge : m \to 1$, it has no nodes and it maps m sites to an unique root. A bigraph $G : n \to \langle m, (X^-, X^+)\rangle$ without inner names, it can be simply converted in a prime bigraph as follows: $(merge \otimes id_{(X^-,X^+)}) \circ G$.

Definition 5.3 (discrete bigraph) *A bigraph $D : \langle n, (X^-, X^+)\rangle \to \langle m, (Y^-, Y^+)\rangle$ is* discrete *if it has no edges and the link map is a bijection. That means all points are open, no pair of points is a peer and no link is idle.*

The discreteness is well-behaved, and preserved by composition and tensor products. It is easy to see that discrete bigraphs form a monoidal sub-precategory of $'$DBIG.

Definition 5.4 (ion, atom and molecule) *For any non atomic control K with arity k and a pair of sequence $\vec{x}^-$ and $\vec{x}^+$ of distinct names, whose overall length is k, we define the discrete* ion $K(v)^{\vec{x}^+}_{\vec{x}^-} : \langle(\vec{x}^-, \emptyset)\rangle \to \langle(\emptyset, \vec{x}^+)\rangle$ *as the bigraph with a unique K-node v, whose ports are separately linked to $\vec{x}^-$ or to $\vec{x}^+$. We omit v when it can be understood.*

For a prime discrete bigraph P with outer names (Y^-, Y^+) the composite $(K^{\vec{x}^+}_{\vec{x}^-} \otimes id_{(Y^-,Y^+)}) \circ P$ is a discrete molecule. *If K is atomic, we define the discrete* atom $K^{\vec{x}^+}_{\vec{x}^-} : (\vec{x}^-, \emptyset) \to \langle(\emptyset, \vec{x}^+)\rangle$*; it resembles an ion, but has no site.*

An arbitrary (non-discrete) ion, molecule or atom is formed by the composition of $\omega \otimes id_1$ with a discrete one. Often we omit $\cdots \otimes id_I$ in the compositions, when there is no ambiguity; for example we write $merge \circ G$ to mean $(merge \otimes id_{(X^-,X^+)}) \circ G$ and $K^{\vec{x}^+}_{\vec{x}^-} \circ P$ to mean $(K^{\vec{x}^+}_{\vec{x}^-} \otimes id_{(Y^-,Y^+)}) \circ P$. Note that every atom and every molecule are prime, furthermore an atom is also ground, but a molecule is not necessarily ground, since it may have sites.

Now, we define some variants of the tensor product, whose can allow the sharing of names. Process calculi often have a parallel product $P \mid Q$, that allows the processes P and Q to share names. In directed bigraphs, this sharing can involve inner downward names and/or outer upword names, as described by the following definitions.

Definition 5.5 (sharing products) *The* outer sharing product*,* inner sharing product *and* sharing product *of two link graphs $A_i : X_i \to Y_i$ ($i = 0, 1$) are defined as follows:*

$$(X^-, X^+) \curlywedge (Y^-, Y^+) \triangleq (X^- \uplus Y^-, X^+ \cup Y^+)$$
$$(X^-, X^+) \curlyvee (Y^-, Y^+) \triangleq (X^- \cup Y^-, X^+ \uplus Y^+)$$

$$A_0 \curlywedge A_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1) : X_0 \otimes X_1 \to Y_0 \curlywedge Y_1$$
$$A_0 \curlyvee A_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1) : X_0 \curlyvee X_1 \to Y_0 \otimes Y_1$$
$$A_0 \parallel A_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1) : X_0 \curlyvee X_1 \to Y_0 \curlywedge Y_1$$

defined when their interfaces are defined and A_i have disjoint node and edge sets.

The outer sharing product*,* inner sharing product *and* sharing product *of two bigraphs $G_i : I_i \to J_i$ are defined by extending the corresponding products on their link graphs with the tensor product on widths and place graphs:*

$$\langle m, X\rangle \barwedge \langle n, Y\rangle \triangleq \langle n+m, X \barwedge Y\rangle \qquad \langle m, X\rangle \barvee \langle n, Y\rangle \triangleq \langle n+m, X \barvee Y\rangle$$

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

defined when their interfaces are defined and G_i have disjoint node and edge sets.

It is simple to verify that $\barwedge$, $\barvee$ and $\parallel$ are associative, with unit ϵ.

Another way of constructing a sharing product of two bigraphs G_0, G_1 is to disjoin the names of G_0 and G_1, then take the tensor product of the two bigraphs and finally merge the name again:

Proposition 5.6 *Let G_0 and G_1 be bigraphs with disjoint node and edge sets. Then*

$$G_0 \barwedge G_1 = \sigma(G_0 \otimes \tau G_1 \zeta) \qquad G_0 \barvee G_1 = (G_0 \otimes \tau G_1 \zeta)\delta \qquad G_0 \parallel G_1 = \sigma(G_0 \otimes \tau G_1 \zeta)\delta$$

where the substitution σ and τ are defined in the following way: if z_i $(i \in n)$ are the upward outer names shared by G_0 and G_1, and w_i are fresh names in bijection with the z_i, then $\tau(z_i) = w_i$ and $\sigma(w_i) = \sigma(z_i) = z_i$ $(i \in n)$. The substitution δ and ζ are defined in a very similar way, but acting on the downward inner names.

Definition 5.7 (prime products) *The* prime outer sharing product *and* prime sharing product *of two bigraphs $G_i : I_i \to J_i$ are defined as follows:*

$$\langle m, (X^-, X^+)\rangle \barwedge\!\!\!\!\barwedge \langle n, (Y^-, Y^+)\rangle \triangleq \langle (X^- \uplus Y^-, X^+ \cup Y^+)\rangle$$

$$G_0 \barwedge\!\!\!\!\barwedge G_1 \triangleq merge_{(width(J_0)+width(J_1))} \circ (G_0 \barwedge G_1) : I_0 \otimes I_1 \to J_0 \barwedge\!\!\!\!\barwedge J_1$$
$$G_0 \mid G_1 \triangleq merge_{(width(J_0)+width(J_1))} \circ (G_0 \parallel G_1) : I_0 \barvee I_1 \to J_0 \barwedge\!\!\!\!\barwedge J_1.$$

defined when their interfaces are defined and G_i have disjoint node and edge sets.

It is easy to show that $\barwedge\!\!\!\!\barwedge$ and $\mid$ are associative, with unit 1 when applied to prime bigraphs. Note that for a wiring ω and a prime bigraph P, we have $\omega \barwedge\!\!\!\!\barwedge P = \omega \barwedge P$ and $\omega \mid P = \omega \parallel P$, because in this case these products have the same meaning.

Now, we can describe *discrete bigraphs,* which are the complement of wirings:

Theorem 5.8 (discrete normal form) *1. Every bigraph G can be expressed uniquely (up to iso) as: $G = (\omega \otimes id_n) \circ D \circ (\omega' \otimes id_m)$, where D is a discrete bigraph and ω, ω' are two wirings satisfying the following conditions:*

- *in ω, if two outer downward names are peer, then their target is an edge;*

- *in ω' there are no edges, and no two inner upward names are peer.*

2. *Every discrete bigraph $D : \langle m, (X^-, X^+)\rangle \to \langle n, (Y^-, Y^+)\rangle$ may be factored uniquely (up to iso) on the domain of each factor D_i, as:*

$$D = \alpha \otimes ((D_0 \otimes \cdots \otimes D_{n-1}) \circ (\pi \otimes id_{dom(\vec{D})}))$$

with α a renaming, each D_i prime and discrete, and π a permutation.

Proof. For the first part, let us consider a generic bigraph $G : \langle n, (X^-, X^+)\rangle \to \langle m, (Y^-, Y^+)\rangle$. We show how to divide G in three parts: a discrete $D : \langle n, (Z^-, Z^+)\rangle \to \langle m, (W^-, W^+)\rangle$ and two wirings $\omega : (W^-, W^+) \to (Y^-, Y^+)$ and $\omega' : (X^-, X^+) \to (Z^-, Z^+)$ satisfying the previous conditions. We proceed by cases:

$p \in P$, $link_G(p) = e \in E$**:** we add a fresh name $w_e \in W^+$ and define $link_D(p) = w_e$ and $link_\omega(w_e) = e$;

$p \in P$, $link_G(p) = y \in Y^+$**:** we add a fresh name $w_y \in W^+$ and define $link_D(p) = w_y$ and $link_\omega(w_y) = y$;

$p \in P$, $link_G(p) = x \in X^-$**:** this case is analogous to the previous one;

$y \in Y^-$, $link_G(y) = e \in E$**:** we define $link_\omega(y) = e$;

$x \in X^+$, $link_G(y) = e \in E$**:** we add a fresh name $z_e \in Z^+$, a fresh name $w_e \in W^+$ and define $link_{\omega'}(x) = z_e$, $link_D(z_e) = w_e$, $link_\omega(w_e) = e$;

$y \in Y^-$, $link_G(y) = x \in X^-$**:** we add a fresh name $w_x \in W^-$, a fresh name $z_x \in Z^-$ and define $link_\omega(y) = w_x$, $link_D(w_x) = z_x$ and $link_{\omega'}(z_x) = x$;

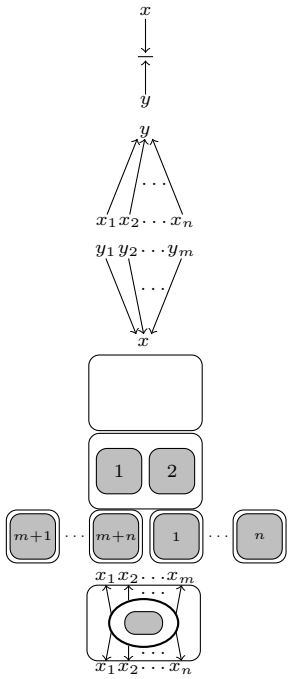
$x \in X^+$, $link_G(x) = y \in Y^+$**:** this case is analogous to the previous one; it is sufficient to invert the direction of links and swap the rule of ω with ω'.

Note that there are no idle names in Z^-, Z^+, W^- and W^+, so those sets are formed only by the fresh names defined in this proof. Furthermore, the three conditions above holds because we create a fresh name every time we need one.

The proof of the second part is easy. Since the outer interface of D has width n, we can decompose D in n discrete and prime parts, obtaining $D_0 \otimes \cdots \otimes D_{n-1}$. The renaming α describe the connections between the inner interface and the outer one. Finally the permutation π gives the right sequence of the sites, so we can take the tensor product of D_i $(i = 0, \ldots . n-1)$ in any order. □

We call this unique factorization *discrete normal form* (DNF). The DNF applies to abstract bigraphs as well, and indeed it will play an important part in the complete axiomatization of DBIG, as we will discuss in the next section.

Note that a renaming is discrete but not prime (since it has zero width); this is why the factorization in Theorem 5.8(2) has such a factor. This unique factorization depends on the fact that the prime bigraphs have no upward inner names and downward outer names. In the special case that D is ground, the factorization in Theorem 5.8(2) is simply $D = d_0 \otimes \cdots \otimes d_{n-1}$, that is a product of discrete and prime ground bigraphs.



$\blacktriangle^x_y{:}(\emptyset, y) \to (x, \emptyset)$	closure
$\triangle^y_X{:}(\emptyset, X) \to (\emptyset, y)$	substitution
$\triangledown^Y_x{:}(x, \emptyset) \to (Y, \emptyset)$	fusion
$1{:}\epsilon \to 1$	a barren root
$merge{:}2 \to 1$	mapping 2 sites in 1 root
$\gamma_{m,n}{:}m + n \to n + m$	swapping m with n
$K^{\vec{x}^+}_{\vec{x}^-}{:}\langle(\vec{x}^-, \emptyset)\rangle \to \langle(\emptyset, \vec{x}^+)\rangle$	a discrete ion

Figure 2: Elementary Bigraphs

6 Algebraic structure of DBIG

In this section we describe a sound and complete axiomatization for directed abstract bigraphs. Furthermore we give a normal form for discrete bigraphs, that is useful to prove the completeness of the axiomatization.

First we introduce the *algebraic signature*, that is a set of elementary bigraphs able to define any other bigraph (Figure 2).

We have to show that all bigraphs can be constructed from these elementary ones by composition and tensor product. Before giving a formal result, we provide an intutive explanation of the meaning of these elementary bigraphs.

- The first three bigraphs build up all wirings, i.e. all the link graphs having no nodes. Indeed, all substitutions (fusions, resp.) can be obtained as tensor products of elementary substitutions $\triangle^y_X$ (fusions $\triangledown^Y_x$, resp.); the tensor products of singleton substitutions $\triangle^y_x$ and/or singleton fusions $\triangledown^x_y$ give all renamings. The composition and the tensor product of substitutions, fusions and closures give all wirings.

- The next three bigraphs define all placings, i.e. all place graphs having no nodes; for example $merge_m : m \to 1$, merging m sites in a unique root, are

defined as:

$$merge_0 \triangleq 1 \qquad merge_{m+1} \triangleq merge \circ (id_1 \otimes merge_m).$$

Notice that $merge_1 = id$ and $merge_2 = merge$, and that all permutations $\pi : m \to m$ are constructed by composition and tensor from the $\gamma_{m,n}$.

- Finally, for expressing any direct bigraph we need to add only the discrete ions $K^{\vec{x}^+}_{\vec{x}^-}$. In particular, we can express any discrete atoms as $K^{\vec{x}^+}_{\vec{x}^-} \circ 1$.

The following proposition shows that every bigraph can be expressed in a normal form, called (again) *discrete normal form* (DNF). We will use D, Q and N to denote primes, discrete prime bigraphs, and the discrete molecules respectively.

Proposition 6.1 (discrete normal form) *In* DBIG *every bigraph G, discrete D, discrete and prime Q and discrete molecule N can be described by an expression of the respective following form:*

$$G = (\omega \otimes id_n) \circ D \circ (\omega' \otimes id_m) \tag{1}$$

where ω, ω' satisfy the conditions given in Theorem 5.8(1);

$$D = \alpha \otimes ((Q_0 \otimes \cdots \otimes Q_{n-1}) \circ (\pi \otimes id_{dom(\vec{Q})})) \tag{2}$$

$$Q = (merge_{n+p} \otimes id_{\emptyset, Y^+}) \circ (id_n \otimes N_0 \otimes \cdots \otimes N_{p-1}) \circ (\pi \otimes id_{(Y^-, \emptyset)}) \tag{3}$$

$$N = (K^{\vec{x}^+}_{\vec{x}^-} \otimes id_{\emptyset, Y^+}) \circ Q. \tag{4}$$

Furthermore, the expression is unique up to isomorphisms on the parts.

Proof. The proof is quite similar to the proof of Theorem 5.8. □

We can use these equations for normalizing any bigraph G as follows; first, we apply equations (1), (2) to G once, obtaining an expression containing discrete and prime bigraphs $Q_0, \ldots, Q_{n-1}$. These are decomposed further using equations (3), (4) repeatedly: each Q_i is decomposed into an expression containing molecules $N_{i,0}, \ldots, N_{i,p_i-1}$, each of which is decomposed in turn into an ion containing another discrete and prime bigraph $Q'_{i,j}$. The last two steps are repeated recursively until the ions are atoms. Note that the unit 1 is a special case of Q when $n = p = 0$.

In Figure 3 we give a set of axioms which we prove to be sound and complete.

Each of these equations holds only when both sides are defined; in particular, recall that the tensor product of two bigraphs is defined only if the name sets are disjoint. It is important to notice also that for ions only the renaming axiom is needed (because the names are treated positionally).

Theorem 6.2 (Completeness of the axiomatization) *Let E_0, E_1 be two expressions constructed from the elementary bigraphs by composition and tensor product. Then, E_0 and E_1 denote the same bigraph in* DBIG *if and only if the equation $E_0 = E_1$ can be proved by the axioms in Figure 3.*

Categorical Axioms

$$A \circ id = A = id \circ A \qquad A \circ (B \circ C) = (A \circ B) \circ C$$
$$A \otimes id_\epsilon = A = id_\epsilon \otimes A \qquad A \otimes (B \otimes C) = (A \otimes B) \otimes C$$
$$\gamma_{I,\epsilon} = id_I \qquad \gamma_{J,I} \circ \gamma_{I,J} = id_{I \otimes J}$$
$$(A_1 \otimes B_1) \circ (A_0 \otimes B_0) = (A_1 \circ A_0) \otimes (B_1 \circ B_0)$$
$$\gamma_{I,K} \circ (A \otimes B) = (B \otimes A) \circ \gamma_{H,J} \qquad (\text{where } A : H \to I, B : J \to K)$$

Link Axioms

$$\bigtriangledown_y^x \circ \bigtriangleup_z^y = \bigtriangledown_z^x \qquad \nabla_x^z \circ \bigtriangledown_y^x = \bigtriangledown_y^z \qquad \nabla_x \circ \bigtriangledown_y^x \circ \bigtriangleup^y = id_\epsilon$$
$$\bigtriangleup^z_{(Y \uplus y)} \circ (id_{(\emptyset,Y)} \otimes \bigtriangleup^y_X) = \bigtriangleup^z_{(Y \uplus X)} \qquad (id_{(Y,\emptyset)} \otimes \nabla_y^X) \circ \nabla_z^{(Y \uplus y)} = \nabla_z^{(X \uplus Y)}$$

Place Axioms

$$merge \circ (1 \otimes id_1) = id_1 \qquad merge \circ \gamma_{1,1} = merge$$
$$merge \circ (merge \otimes id_1) = merge \circ (id_1 \otimes merge)$$

Node Axioms

$$(id_1 \otimes \alpha) \circ K^{\vec{x}^+}_{\vec{x}^-} = K^{\alpha(\vec{x}^+)}_{\vec{x}^-} \qquad K^{\vec{x}^+}_{\vec{x}^-} \circ (id_1 \otimes \alpha) = K^{\vec{x}^+}_{\alpha(\vec{x}^-)}$$

Figure 3: Axiomatization for the abstract directed bigraphs.

Proof. The proof is similar to that of [3, Theorem 10.2]. The "if" direction is simple to prove, since it requires to check that each axiom is valid. The "only if" direction is in two steps. First, we prove by induction on the structure of expressions, that the equality between an expression and its DNF is derivable from the axioms. Next, since DNFs are taken up to iso, we have to show that the equality between isomorphic DNFs is provable from the axioms. This is proved by showing that the axioms can prove the isomorphisms of the components of a DNF, which are ions, discrete and prime bigraphs, and discrete bigraphs. □

7 The λ-calculus

In this section we describe an encoding of both the call-by-name and the call-by-value λ-calculus. Recall that the set Λ of λ-terms are the terms up-to α-equivalence generated by the following grammar:

$$M, N ::= x \mid \lambda x.M \mid MN.$$

A *value* is either a λ-abstraction or a variable; values are ranged over by V.

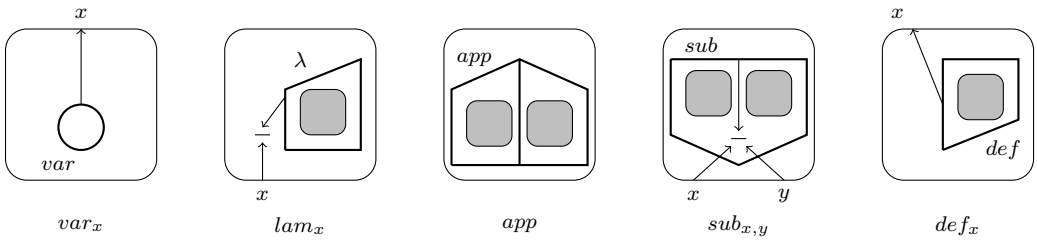


Figure 4: The signature for the λ-calculus.

The call-by-name reduction semantics is defined by the following rules

$$(\lambda x.M)N \to M[N/x] \quad (\beta) \qquad \frac{M \to M'}{MN \to M'N} \qquad \frac{N \to N'}{MN \to MN'}$$

while the call-by-value reduction semantics is defined by the following rules

$$(\lambda x.M)V \to M[V/x] \quad (\beta_v) \qquad \frac{M \to M'}{MN \to M'N} \qquad \frac{N \to N'}{MN \to MN'}$$

In Figure 4 we give a signature for representing the λ-calculus "with single substitutions", that is where a substitution is performed once for each variable occurrence. This signature resembles Milner's encoding using binding bigraphs, but in directed bigraphs we do not need to introduce further binding structures.

We can define a translator operator $[\![\cdot]\!] : \Lambda \to \text{DBIG}$ as follows:

$$[\![x]\!] = var_x \qquad [\![\lambda x.M]\!] = lam_x \circ ([\![M]\!] \jmath \triangle^x) \qquad [\![MN]\!] = app \circ ([\![M]\!] \jmath [\![N]\!])$$

Intuitively, a λ-term M is represented by a ground bigraph $[\![M]\!] : \epsilon \to \langle(\emptyset, X^+)\rangle$ whose place hierarchy reflects the syntactic tree of M and the outer upwards names X^+ are the free variables of M. Each λ-expression is represented by a control and a local resource which is bound to a upward name in the inner interface.

Proposition 7.1 *Let M, N be two λ-terms; then, $M \equiv_\alpha N$ iff $[\![M]\!] = [\![N]\!]$.*

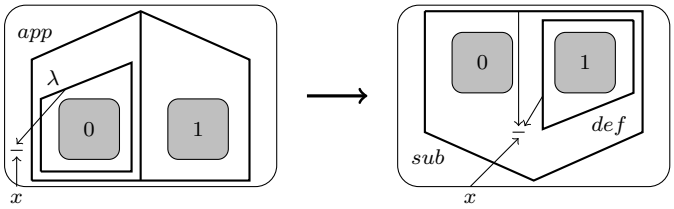
Let us now see how we can represent the two semantics of the λ-calculus. For the call-by-name semantics, we define the controls *lam* and *def* as passive, *sub* and *app* as active. The reaction rules are given in Figure 5.

For the call-by-value λ-calculus, we have to replace the App_{cbn} rule with two rules $App_{cbv\text{-}var}$ and $App_{cbv\text{-}lam}$ (Figure 6) corresponding to the two cases of values where the application can be performed.

For both variants, we can prove the following result:

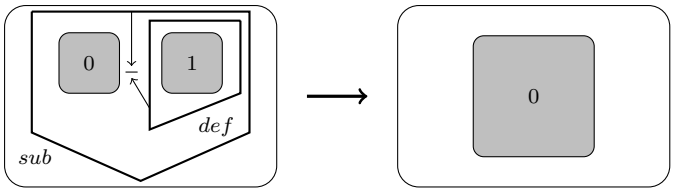
Proposition 7.2 *Let M, M' be two λ-terms.*

1. *If $M \to M'$ then $[\![M]\!] \to^* [\![M']\!]$;*



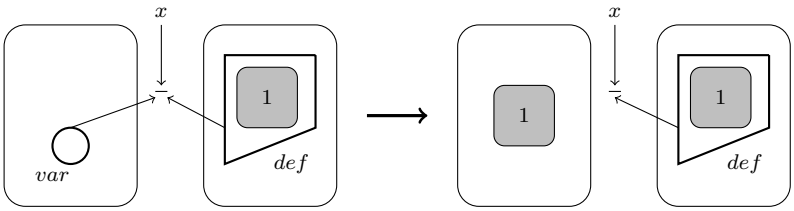
App_{cbn}

$$app \circ (lam_x \curlywedge id_1) \longrightarrow sub_{x,y} \circ (id_1 \curlywedge def_y)$$



$Sub_{dispose}$

$$sub_{x,y} \circ (id_1 \curlywedge \triangle^x \curlywedge def_y) \longrightarrow id_1$$



Sub_{var}

$$⧗_y^x \circ (var_y \curlywedge def_y) \longrightarrow ⧗_y^x \circ (id_1 \curlywedge def_y)$$

Figure 5: Reactions for the call-by-name λ-calculus.

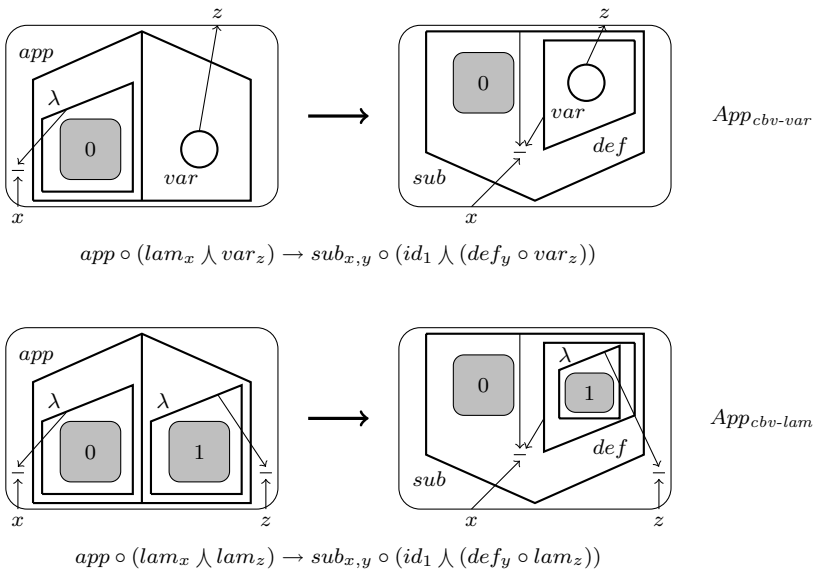


Figure 6: Reactions for the call-by-value λ-calculus.

2. If $[\![M]\!] \rightarrow^* [\![M']\!]$ *then* $M \rightarrow^* M'$.

Proof. By induction on the lenght of traces.

1. The application of β (or β_v) is encoded by applying App_{cbn} (or one of $App_{cbv\text{-}var}$ and $App_{cbv\text{-}lam}$) on the correct sub-bigraph, i.e. the one which encodes the right side of the rule. Next we use Sub_{var} for every occurrence of x in M, finally we apply $Sub_{dispose}$ to eliminate the unnecessary controls sub and def.

2. First of all note that, by definition of $[\![\cdot]\!]$, $[\![M']\!]$ has no sub or def controls. If $[\![M]\!] \rightarrow^* [\![M']\!]$, in the trace there are one or more application of App_{cbn} (or $App_{cbv\text{-}var}$ and $App_{cbv\text{-}lam}$), so we use the β (or β_v) rule on the corresponding λ-subterm. We can ignore the Sub_{var} and $Sub_{dispose}$ rules because the substitutions in λ-calculus are performed instantaneously. □

8 Conclusions

In this work, 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.

Moreover, a sound and complete axiomatization of the precategory of directed bigraphs has been proposed. We have used this axiomatization for encoding the λ-calculus, both in call-by-name and call-by-value variants. It is interesting to notice that no further extensions (such as binding signatures) are needed.

Future work We plan to apply directed bigraphs for representing other calculi, in particular calculi with resources, locations, etc., which can be represented by edges. An interesting candidate could be the ν-calculus [13], whose bound names can be represented by edges, and variables by names. We think that also the Fusion calculus [12] can be represented easily in directed bigraphs. It is interesting to investigate the corresponding wide transition systems.

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.

References

[1] H. Ehrig and B. König. Deriving bisimulation congruences in the dpo approach to graph rewriting. In Walukiewicz [18], pages 151–166.

[2] O. H. Jensen and R. Milner. Bigraphs and transitions. In *Proc. POPL*, pages 38–49, 2003.

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

[4] B. Klin, V. Sassone, and P. Sobociński. Labels from reductions: Towards a general theory. In J. L. Fiadeiro, N. Harman, M. Roggenbach, and J. J. M. M. Rutten, editors, *Proc. CALCO*, volume 3629 of *Lecture Notes in Computer Science*, pages 30–50. Springer, 2005.

[5] S. Lack and P. Sobociński. Adhesive categories. In Walukiewicz [18], pages 273–288.

[6] J. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In C. Palamidessi, editor, *Proc. CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2000.

[7] M. Merro and F. Zappa Nardelli. Behavioral theory for mobile ambients. *J. ACM*, 52(6):961–1023, 2005.

[8] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[9] R. Milner. Bigraphical reactive systems. In K. G. Larsen and M. Nielsen, editors, *Proc. 12th CONCUR*, volume 2154 of *Lecture Notes in Computer Science*, pages 16–35. Springer, 2001.

[10] R. Milner. Pure bigraphs: Structure and dynamics. *Inf. Comput.*, 204(1):60–122, 2006.

[11] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. *Inform. and Comput.*, 100(1):1–77, 1992.

[12] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, pages 176–185. IEEE, Computer Society Press, July 1998.

[13] A. M. Pitts and I. D. B. Stark. Observable properties of higher order functions that dynamically create local names, or what's new? In A. M. Borzyszkowski and S. Sokolowski, editors, *MFCS*, volume 711 of *Lecture Notes in Computer Science*, pages 122–141. Springer, 1993.

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

[15] V. Sassone and P. Sobociński. Reactive systems over cospans. In *Proc. LICS*, pages 311–320. IEEE Computer Society, 2005.

[16] P. Sewell. From rewrite rules to bisimulation congruences. *Theor. Comput. Sci.*, 274(1-2):183–230, 2002.

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

[18] I. Walukiewicz, editor. *Proceedings of Foundations of Software Science and Computation Structures*, volume 2987 of *Lecture Notes in Computer Science*. Springer, 2004.