Fundamenta Informaticae XX (2014) 1–31
DOI 10.3233/FI-2012-0000
IOS Press

On hierarchical graphs: reconciling bigraphs, gs-monoidal theories and gs-graphs[*]

Roberto Bruni [C]
Computer Science Department
University of Pisa, Italy
bruni@di.unipi.it

Ugo Montanari
Computer Science Department
University of Pisa, Italy

Gordon Plotkin
LFCS, School of Informatics
University of Edinburgh, UK

Daniele Terreni
Computer Science Department
University of Pisa, Italy

Abstract. Compositional graph models for global computing systems must account for two relevant dimensions, namely structural containment and communication linking. In Milner's *bigraphs* the two dimensions are made explicit and represented as two loosely coupled structures: the *place graph* and the *link graph*. Here, bigraphs are compared with an earlier model, *gs-graphs*, originally conceived for modelling the syntactical structure of agents with α-convertible declarations. We show that gs-graphs are quite convenient also for the new purpose, since the two above mentioned dimensions can be recovered by considering only a specific class of hyper-signatures. With respect to bigraphs, gs-graphs can be proved essentially equivalent, with minor differences at the interface level. We argue that gs-graphs offer a simpler and more standard algebraic structure, based on monoidal categories, for representing both states and transitions. Moreover, they can be equipped with a simple type system to check the well-formedness of *legal* gs-graphs that are shown to characterise *binding* bigraphs. Another advantage concerns a textual form in terms of sets of assignments, which can make implementation easier in rewriting frameworks like Maude.

Keywords: gs-monoidal theories, gs-graphs, pure bigraphs, binding bigraphs

[*]Research supported by the EU Integrated Project 257414 ASCENS and by the Italian MIUR Projects IPODS (PRIN 2008) and CINA (PRIN 2010/11).
[C]Corresponding author

1. Introduction

When modelling distributed systems, it is necessary to represent states and their evolution. Usually, states are seen as terms of an algebra equipped with certain structural axioms, and state transitions are defined via conditional term rewriting rules in SOS format. An alternative is to represent states as graphs and transitions as graph transformations. To have the best of the two approaches, it is sometimes possible to characterise terms up to structural axioms as graphs and the transitions as graph transformations. A good example is provided by arrows of *gs-monoidal theories* [6, 2], which can be seen as a particular type of graphs, called *gs-graphs*, and transitions, represented as 2/double cells of a 2/double category, which can be seen as gs-graph transformations. The approach has been applied to π-calculus [10], to CCS with localities [10] and causality [2], to logic programming [3] and to other models of computation.

Gs-monoidal theories are certain symmetric strict monoidal categories [19] whose objects are elements of the free monoid over a set S of symbols, with each such symbol carrying a commutative comonoid structure. As such they form a class of coloured PROduct and Permutation categories (PROPs) [18, 16, 13], where the term "coloured" means that they are taken over many objects instead of a single one. They can also be seen as a variation of cartesian (Lawvere) theories, but without the two naturality axioms that allow copying shared sub-terms and garbage-collecting unused terms. In addition, gs-monoidal theories are built out of symbols taken from a hyper-signature instead of an ordinary signature. The difference is that the codomains of symbols in a hyper-signature are not constrained to be single-sorted, but, like domains, can be a tuple of sorts. The corresponding gs-graphs are some kind of dags where substructures can be shared via ordinary duplicators, as in term graphs, or hyper-signature symbols (see, e.g., Fig. 3(a), where x_5 is shared by f and g and where both p_7 and x_3 are shared by the two h's). A useful feature of gs-graphs is that they can be represented in textual form as sets of assignments of the form $x, y := f(u, v, w)$, where f is a hyper-signature symbol with input arity three and output arity two, with x, y, u, v, w α-convertible names in the set of assignments.

Recent developments in the area of open-ended systems for global computing have emphasised the need for hierarchical graph models: they have two relevant dimensions, namely structural containment and communication linking. The former has to do with the structural design of processes (e.g., the scoping of a transaction, a compensation, or a session, or the nesting of an ambient, a membrane, or an environment); it induces a tree-like hierarchy on nodes. The latter concerns interaction capabilities (e.g., for communication, handshaking, or connectivity) that are flat, and may connect any tree nodes. This is the *pure* case. The situation is more complex in the *binding* case, where a name used for communication is declared at some level in the tree hierarchy and is usable only below its declaration point. Inspired by Cardelli and Gordon's ambients calculus [4] and aiming at defining a general, flexible and easy to grasp model, Milner defined *bigraphs*, where the two dimensions are made explicit and represented as essentially orthogonal structures, called *place graph* and *link graph*. Bigraphs [24, 23, 15, 21] have been studied in depth from several points of view, in particular as a basis of the reactive system approach, where a labelled transition system, over which bisimilarity is a congruence, is synthesised from a reduction semantics. This part of the theory is not covered for gs-graphs.

In this paper we aim to expose the gs-monoidal structure of bigraphs. Technically, we take advantage of the work on gs-graphs to trace a correspondence between bigraphs and gs-monoidal theories, because gs-graphs are closer to bigraphs. Gs-graphs were originally conceived with the syntactical structure of agents with α-convertible declarations in mind, not the hierarchical structure of global computing systems. However it turns out that they are also quite convenient for this new purpose. The nesting and linking structure can be recovered by considering two types of nodes: one for representing intermediate places in the tree-like hierarchy, here coloured in black, and one for representing communication channels, here colored in white.

We start by comparing gs-graphs with (lean support equivalent) bigraphs. Our first correspondence result states that the two models essentially coincide in the pure case. The only difference is in the interface: gs-graphs have an ordered tuple of ports and all the names are α-convertible, while bigraph interfaces are decorated with names. The "names versus strings in a free monoid" dichotomy can already be found in the simple case of equational theories versus Lawvere theories, where one has to translate between variables and numbers. On the one hand, named connectors facilitate the direct representation in bigraphs of standard process calculi operations; on the other hand, they make the algebraic structure of bigraphs more complex and less standard: e.g., parallel composition is partial and sequential composition is associative only up to isomorphism. Also the rewriting structure does not generate composable cells. On the contrary, gs-graphs inherit from gs-monoidal theories a variety of well-behaved operations. Our first result exposes the gs-monoidal structure of bigraphs, when their interface are equipped with a choice of suitable "shuffling" bijections.

Our second correspondence result is concerned with (lean support equivalent) binding bigraphs. In the binding case, it is necessary to constrain the compositional structure to generate only *legal* gs-graphs. For bigraphs [9], additional information is inserted in the interface to allow for a complete axiomatisation. In our approach, we introduce a type system which recognises legal binding gs-graphs. On gs-monoidal theories the type system is represented by membership sentences in membership equational logic [20], while on gs-graphs we exploit a quite simple relational type system. When the gs-graph is pure, no pairs are generated; parallel composition does not add any pair; and for sequential composition existing pairs are preserved, but new pairs, possibly leading to inconsistency, are generated. The complexity of the proposed typing algorithm is quadratic in the number of nodes.

The analogies and differences between the two different proposals, together with the transformations that move from one framework to the other, lead us to conclude that: (i) support-equivalent bigraphs can be presented at a suitable level of abstraction as arrows of a particular free coloured PROPS, in a perfectly standard way; and (ii) the gs-graphs representation seems to offer some advantages over the others.

Structure of the paper. Section 2 recaps the basics of the models we are comparing. Section 3 addresses the case of pure signatures, while the binding case is discussed in Section 4. Finally, Section 5 contains come concluding remarks.

All definitions and results in Sections 3 and 4.2, with special emphasis on the notion of legal gs-graph and corresponding type system, are original to this contribution, as well as the notion of ν-signatures and amended gs-graphs found in Section 2. Additional material concerned with the technical details of the main correspondence results is given in the Appendix.

2. Background on graph-based structures

Notation. For an ordinal n, we write $i \in n$ as a shorthand for $i \in \{0, \dots, n-1\}$ and let $[n, m]$ denote the set $\{i \mid n \leq i \leq m\}$. We let Id_m be the identity function on the ordinal m. We use the symbol $\uplus$ for disjoint union of sets. We let S^* denote the free monoid over the elements in S, whose product is

$$\text{(ops)}\ \frac{f \in \Sigma_{u,v}}{f : u \to v} \qquad \text{(ids)}\ \frac{u \in S^*}{id_u : u \to u} \qquad \text{(bang)}\ \frac{u \in S^*}{!_u : u \to \epsilon} \qquad \text{(dup)}\ \frac{u \in S^*}{\nabla_u : u \to uu}$$

$$\text{(sym)}\ \frac{u, v \in S^*}{\rho_{u,v} : uv \to vu} \qquad \text{(seq)}\ \frac{t : u \to v \quad t' : v \to w}{t; t' : u \to w} \qquad \text{(par)}\ \frac{t : u \to v \quad t' : u' \to v'}{t \otimes t' : uu' \to vv'}$$

Figure 1. Inference rules of gs-monoidal theories

juxtaposition and whose unit is denoted by ϵ. We abbreviate the juxtaposition of n consecutive objects u by u^n, with $u^0 = \epsilon$. Likewise, given a function $f : V \to V$, we denote by $f^n : T \to V$ the consecutive application of f for n times. We write $\emptyset_V$ for the unique function with empty domain and codomain V. We use subscripts to disambiguate elements when necessary, e.g., by writing V_G when referring to the set of vertices of the graph G. Finally, we overload $|\cdot|$ to denote the length of a string, the cardinality of a set and the support of place graphs, link graphs and bigraphs (see Definitions 2.23, 2.26 and 2.29).

2.1. Gs-monoidal theories

The gs-monoidal approach[1] is based on representing resources as nodes, computational entities as hyper-edges and interaction capabilities by the way in which hyper-edges are connected to nodes. We call a tentacle each connection from an hyper-edge to a node. For example, nodes can model communication channels and tentacles can express capabilities to perform i/o operations on them.

The idea is to consider a particular class of graphs obtained by selecting a few basic shapes for edges (i.e., to fix a signature), to equip them with suitable source and target interfaces and to freely compose them in series and in parallel to build larger and more complex shapes.

Definition 2.1. (hyper-signature)
Given a set S of sorts, a *hyper-signature* Σ is a family $\{\Sigma_{u,v}\}_{u,v \in S^*}$ of sets of operators such that each $f \in \Sigma_{u,v}$, also written $f : u \to v$, has input arity $|u|$ and output arity $|v|$.

For simplicity, in the rest of the paper we shall always omit the adjective 'hyper', and leave it implicit that we are dealing with hyper-structures (signature, graphs, edges, etc.).

When building larger shapes, one is further allowed: 1) to rearrange the order of nodes in the interface (e.g., to make adjacent the nodes that must be connected to the same edge) via so-called symmetries ρ; 2) to make some nodes be private to a certain sub-graph (no further edges can be attached to them) thanks to so-called dischargers !; and 3) to attach more than two tentacles to the same node via duplicators ∇.

Definition 2.2. (gs-monoidal theory)
Given a signature Σ over a set of sorts S, the *gs-monoidal theory* $\mathbf{GS}(\Sigma)$ is the symmetric, strict monoidal category whose objects are the elements of S^* and whose arrows $t\colon u \to v$ are equivalence classes of gs-monoidal terms, i.e., terms generated by the inference rules in Fig. 1 subject to the following conditions (where we tacitly assume all symbols to be universally quantified unless otherwise specified):

- identities and sequential composition satisfy the axioms of categories:
  [identity] $id_u \; ; t = t = t \; ; id_v$ for all $t : u \to v$;
  [associativity] $t_1 \; ; (t_2 \; ; t_3) = (t_1 \; ; t_2) \; ; t_3$ whenever any side is defined;

[1]The name *gs* stands for "graph structure".

- $\otimes$ is a monoidal functor with unit id_ϵ, i.e., it satisfies:
  [monoid] $t \otimes id_\epsilon = t = id_\epsilon \otimes t \qquad t_1 \otimes (t_2 \otimes t_3) = (t_1 \otimes t_2) \otimes t_3$;
  [functoriality] $id_{uv} = id_u \otimes id_v$, and
  $(t_1 \otimes t_2) \mathbin{;} (t'_1 \otimes t'_2) = (t_1 \mathbin{;} t'_1) \otimes (t_2 \mathbin{;} t'_2)$ whenever both sides are defined;

- ρ is a symmetric monoidal natural transformation, i.e., it satisfies:
  [naturality] $(t \otimes t') \mathbin{;} \rho_{v,v'} = \rho_{u,u'} \mathbin{;} (t' \otimes t)$ for all $t : u \to v$, $t' : u' \to v'$;
  [symmetry] $(id_u \otimes \rho_{v,w}) \mathbin{;} (\rho_{u,w} \otimes id_v) = \rho_{uv,w} \quad \rho_{\epsilon,u} = \rho_{u,\epsilon} = id_u \quad \rho_{u,v} \mathbin{;} \rho_{v,u} = id_{uv}$;

- ∇ and ! satisfy the following axioms:
  [monoidality] $\nabla_{uv} \mathbin{;} (id_u \otimes \rho_{v,u} \otimes id_v) = \nabla_u \otimes \nabla_v \qquad !_{uv} = !_u \otimes !_v$;
  [unit and duplication] $!_\epsilon = \nabla_\epsilon = id_\epsilon \qquad \nabla_u \mathbin{;} (id_u \otimes !_u) = id_u \qquad \nabla_u \mathbin{;} \rho_{u,u} = \nabla_u$
  $\nabla_u \mathbin{;} (id_u \otimes \nabla_u) = \nabla_u \mathbin{;} (\nabla_u \otimes id_u)$.

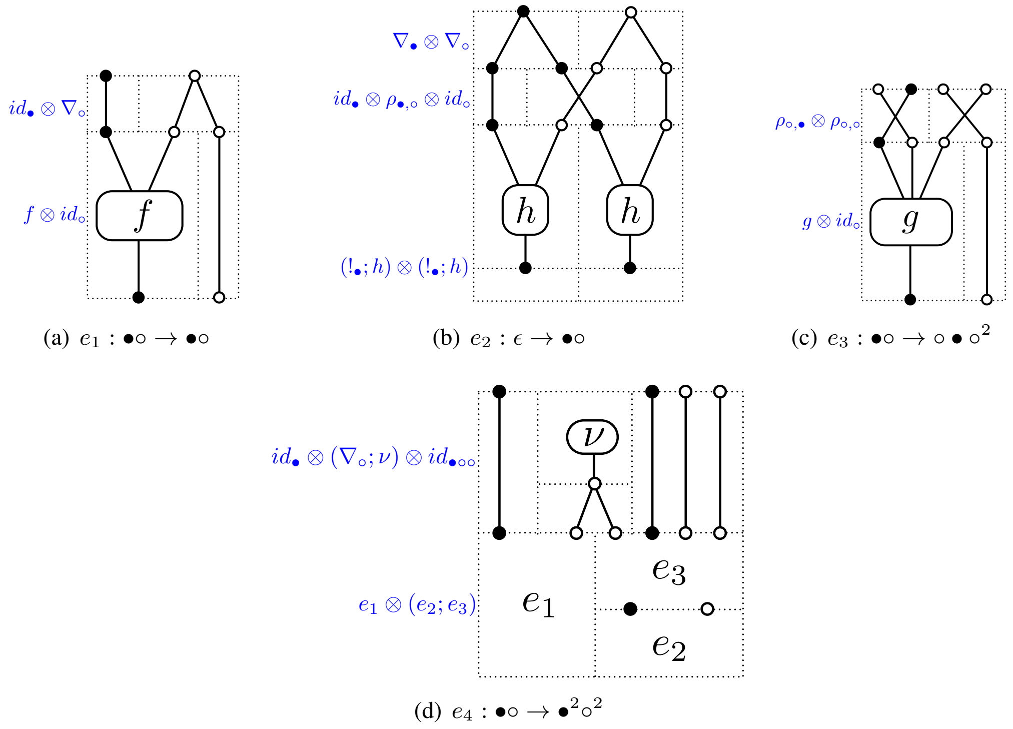
By rule (ops), the basic terms include one generator for each operator of the signature. All other basic terms define the wiring of tentacles that can be used to build larger graphs: the identities (ids), the dischargers (bang), the duplicators (dup) and the symmetries (sym). These are the elementary bricks of our terms, and we get the remaining ones by closing them with respect to sequential (seq) and parallel (par) composition. Every term $t\colon u \to v$ generated by the inference rules is typed by two sequences of sorts: a source u and a target v. They are relevant for the sequential composition, which is a partial operation. To limit the number of parentheses in expressions t, we assume $\otimes$ has precedence over ;.

Remark 2.3. (gs-monoidal theories and PROPs)
Inspired by the seminal work on flownomial algebras [8], a sound and complete axiomatisation of (acyclic) ranked term graphs[2] has been proposed in [6]. In particular, it has been shown that there is a bijective correspondence between term graphs over a signature Σ of rank (u, v) and the homset $\mathbf{GS}(\Sigma)[u, v]$ of arrows of the gs-monoidal theory of Σ from u to v. This result is analogous to the characterisation of (tuples of) terms over a signature Σ as arrows of the Lawvere theory of Σ, considered as the free cartesian category generated by Σ. When Σ is a standard one-sorted theory the result can be recast in the theory of PROPs [18, 17], analogously to what has been recently done for dags in [11]. Here we are interested in a slightly more general setting than that of [6] and of PROPs, because we shall consider two-sorted signatures. Concerning the axioms in Definition 2.2, analogous axiomatisations of various families of graphs are available, and most of them can be seen as enrichments of symmetric monoidal categories where (node) sharing is handled explicitly. The axioms allow one to represent graphs as expressions such that equivalence classes of expressions with respect to the axioms are one-to-one with equivalence classes graphs taken up to isomorphism. We refer the interested reader to the interdisciplinary survey [26] that relates various notions of monoidal categories and their associated string diagrams.

Remark 2.4. (cogs-monoidal structure)
For ease of modelling bigraphs, we reverse the sense of direction for composing arrows, i.e., we take cogs-monoidal theories. As a matter of notation we swap implicitly the source and target of each arrow, e.g., letting

$$\rho_{u,v} : vu \to uv \qquad \nabla_u : u^2 \to u \qquad !_u : \epsilon \to u.$$

[2]Term graphs are defined as isomorphism classes of ranked directed graphs, where the rank is a pair of natural numbers denoting the number of variables and of roots in the term graph, to be used for composition.


$id_\bullet \otimes \nabla_\circ$
$f \otimes id_\circ$
f
$\nabla_\bullet \otimes \nabla_\circ$
$id_\bullet \otimes \rho_{\bullet,\circ} \otimes id_\circ$
h
h
$(!_\bullet; h) \otimes (!_\bullet; h)$
$\rho_{\circ,\bullet} \otimes \rho_{\circ,\circ}$
$g \otimes id_\circ$
g
$id_\bullet \otimes (\nabla_\circ; \nu) \otimes id_{\bullet\circ\circ}$
ν
$e_1 \otimes (e_2; e_3)$
e_1
e_3
e_2

(a) $e_1 : \bullet\circ \to \bullet\circ$ (b) $e_2 : \epsilon \to \bullet\circ$ (c) $e_3 : \bullet\circ \to \circ \bullet \circ^2$

(d) $e_4 : \bullet\circ \to \bullet^2\circ^2$

Figure 2. Graphical representation of the terms from Example 2.7.

Definition 2.5. (ν-signature)
We shall focus on two-sorted signatures over $S = \{\bullet, \circ\}$, where $\bullet$ nodes are used for locations, while $\circ$ nodes for links. Moreover, we leave implicit that all signatures include the special operator $\nu : \circ \to \epsilon$ and all remaining operators are of the form $f : \bullet\circ^k \to \bullet\circ^n$. We call them *$\nu$-signatures*.

A symbol $f : \bullet\circ^k \to \bullet\circ^n$ can be regarded as a site constructor[3] that takes an existing location together with n names from the environment and build a new sub-location where k fresh names are available. While the term ν-signature is coined here, free gs-monoidal theories over suitable ν-signatures have been already exploited, e.g., in [10, 2] for representing different kinds of process calculi.

Remark 2.6. (idle edge: the term $!_\circ\, ;\, \nu$)
We will see later that the term $!_\circ\, ;\, \nu : \epsilon \to \epsilon$ denotes a special arrow that is the counterpart of so-called *idle edges* in bigraphs jargon. While lean support equivalent bigraphs abstract away from idle edges, the corresponding axiom $!_\circ\, ;\, \nu = id_\epsilon$ is not standard for gs-monoidal theories. This point is further discussed in the rest of the paper, when we consider amended gs-graphs, lean support equivalent bigraphs and their correspondence. In the following, we call νgs-monoidal theory the free coloured PROP over the axiom of gs-monoidal theories and the additional axiom $!_\circ\, ;\, \nu = id_\epsilon$.

[3]Remind that we are considering arrows in the opposite category, whose direction is swapped w.r.t. that of gs-monoidal theories.

Example 2.7. Let us consider a ν-signature with three operators $f, h : \bullet \to \bullet\circ$ and $g : \bullet \to \bullet\circ^2$, representing, e.g., firewall, host and gateway components within networking systems. Then we can compose, e.g., the terms e_1, e_2, e_3, e_4 below. To help the intuition, their graphical representation can be found in Figures 2(a)–2(d), where dotted lines are used to emphasise the basic building blocks of each expression.

$$\begin{array}{llll}
e_1 & \triangleq & f \otimes id_\circ \,;\, id_\bullet \otimes \nabla_\circ & : \bullet\circ \to \bullet\circ \\
e_2 & \triangleq & (!_\bullet \,;\, h) \otimes (!_\bullet \,;\, h) \,;\, id_\bullet \otimes \rho_{\bullet,\circ} \otimes id_\circ \,;\, \nabla_\bullet \otimes \nabla_\circ & : \epsilon \to \bullet\circ \\
e_3 & \triangleq & g \otimes id_\circ \,;\, \rho_{\circ,\bullet} \otimes \rho_{\circ,\circ} & : \bullet\circ \to \circ\bullet\circ^2 \\
e_4 & \triangleq & e_1 \otimes (e_2 \,;\, e_3) \,;\, id_\bullet \otimes (\nabla_\circ \,;\, \nu) \otimes id_{\bullet\circ\circ} & : \bullet\circ \to \bullet^2\circ^2
\end{array}$$

2.2. Gs-graphs

The algebraic structure of gs-monoidal theories finds suitable realisation in graph-based modelling: arrows can be interpreted as concrete acyclic directed graphs with interfaces, taken up to renaming of their nodes; all such graphs are represented by some arrow; any two isomorphic graphs whose interfaces match are represented by the same arrow and are thus equivalent abstract graphs.

We find it convenient to represent concrete gs-graphs as sets of formal assignments. We assume $\mathcal{N}$ is a denumerable set of S-sorted names, equipped with a total order $\leq$ and such that there are infinitely many names for each sort. Names are denoted by $n_1 : s_1, n_2 : s_2, \ldots$ or simply by $n_1, n_2, \ldots$ when the sort is clear from the context. A name substitution is a sort-preserving morphism $\sigma : \mathcal{N} \to \mathcal{N}$.

Remark 2.8. When the sort $S = \{\bullet, \circ\}$ is considered, we use $p, q, \ldots$ for names of sort $\bullet$ and $x, y, z, \ldots$ for names of sort $\circ$. Often we assume the order $\leq$ is induced by subscripts, i.e., that $n_i \leq n_j$ if $i \leq j$.

Definition 2.9. (proper assignment, auxiliary assignment)
Let $n'_i : s'_i$ for $i \in [1, k]$, $n_j : s_j$ for $j \in [1, h]$, $u = s'_1 \ldots s'_k$ and $v = s_1 \ldots s_h$. A *proper assignment* is written $n'_1 \ldots n'_k := f(n_1, \ldots, n_h)$ where $f \in \Sigma_{u,v}$.[4] An *auxiliary assignment* has either the form $n := n'$ (aliasing), with n and n' having the same sort, or $!(n)$ (name disposal).

Definition 2.10. (assigned name, used name)
Let G be a set of assignments. When a name appears in the left member of an assignment we say that it is *assigned*, when it appears in the right member we say that it is *used*. We write $n \sqsubset_G n'$ if G contains an assignment where n is used and n' is assigned.

Proper assignments define the edges of the graphs, whose tentacles are attached to nodes named according to their assigned and used names. Node sharing is realised by using the same name more than once. Auxiliary assignments allow to expose more references to the same node in the interface or to prevent certain nodes from appearing in the interface. A gs-graph is a set of assignments that satisfy some additional constraints, for which we need to introduce the following terminology and notation.

Definition 2.11. (inner connection, outer connection)
The *outer connections* of G are all names that are used but not assigned. For an auxiliary assignment $n := n'$ we say n is an *inner connection*. We denote with $ic(G)$ (resp. $oc(G)$) the list of the inner (resp. outer) connections of a set of assignments G (ordered according to $\leq$).

[4] As a matter of notation, if $f \in \Sigma_{u,\epsilon}$ we write $n'_1, \ldots, n'_k := f$, and if $f \in \Sigma_{\epsilon,v}$ we write just $f(n_1, \ldots, n_h)$.

Definition 2.12. (concrete gs-graph)
A *concrete gs-graph* is a set of assignments G s.t.: (1) every name is assigned at most once; (2) the transitive closure $\sqsubset_G^+$ of $\sqsubset_G$ is irreflexive; (3) every $n \in ic(G)$ is a maximal element of $\sqsubset_G^+$; (4) for each name $n \notin ic(G)$ (exactly) one assignment $!(n)$ is present.

Remark 2.13. (idle edge: the assignments $x := \nu$ and $!(x)$)
In the case of ν-signatures, the gs-graph counterpart of idle edges is given by pairs of assignments $x := \nu$ and $!(x)$ such that x is not used elsewhere in G.

Definition 2.14. (abstract gs-graph)
Two concrete gs-graphs G and H are *isomorphic* if H can be obtained from G by applying an injective name substitution that respects the total ordering $\leq$ of the inner and outer connections. An *abstract gs-graph* (or simply *gs-graph*) is the equivalence class of a concrete gs-graph modulo isomorphism.

Since gs-graphs are identified up to isomorphism, the exact choice of names is immaterial.

Remark 2.15. (concise representation of gs-graphs)
The definition of gs-graph allows us to introduce a more concise representation by assuming that: (i) an auxiliary assignment of the form $!(n)$ is omitted whenever n is used in some other assignment; and (ii) an auxiliary assignment $n := n'$ is omitted if n is not an outer connection and it is a maximal element, except for n, of the partial order $\sqsubseteq^+$.

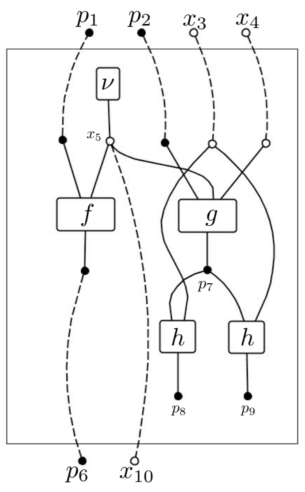
It has been shown that the gs-graphs defined over a signature Σ form a symmetric monoidal category that is naturally isomorphic to the gs-monoidal theory of Σ (see [10]). The idea is that a gs-graph G can be regarded as an arrow $G : ic(G) \to oc(G)$. Then we can fix *atomic* gs-graphs for the basic building blocks of gs-monoidal theories and define how to compose gs-graphs in sequence and in parallel.

Definition 2.16. (atomic gs-graph)
Atomic gs-graphs are defined as follows (for a consistent choice of sorted names):

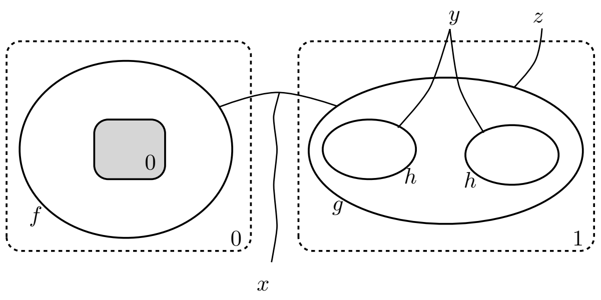
$$\begin{array}{rrcl rrcl}
\text{(ops)} & f & \triangleq & \{n'_1, \ldots, n'_{|u|} := f(n_1, \ldots n_{|v|})\} \quad \text{if } f \in \Sigma_{u,v} & & & & \\
\text{(sym)} & \rho_{s,s'} & \triangleq & \{n_3 := n_2\,,\, n_4 := n_1\} & \text{(ids)} & id_s & \triangleq & \{n_2 := n_1\} \\
\text{(dup)} & \nabla_s & \triangleq & \{n_2 := n_1\,,\, n_3 := n_1\} & \text{(bang)} & !_s & \triangleq & !(n)
\end{array}$$

Definition 2.17. (sequential composition of gs-graphs)
Let $G : u \to v$ and $H : v \to w$ be such that $oc(G) = ic(H)$ and no other names are shared between G and H. Let A be the set of assignments in H of the form $n := n'$ (aliasing) and let σ the corresponding name substitution. Their *sequential composition* is the gs-graph: $G; H \triangleq (G\sigma) \cup (H \backslash A) : u \to w$ with $ic(G; H) = ic(G)$ and $oc(G; H) = oc(H)$.

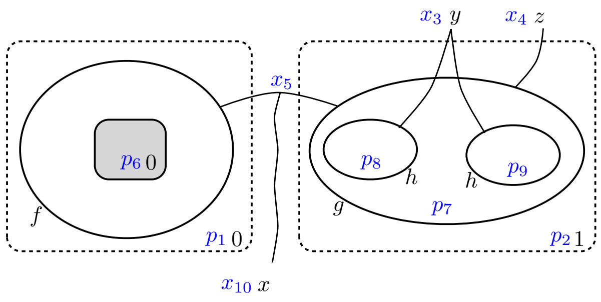
Definition 2.18. (parallel composition of gs-graphs)
Let $G_0 : u_0 \to v_0$ and $G_1 : v_1 \to w_1$ be such that for every name n in G_0 and n' in G_1 we have $n < n'$. Their *parallel composition* is the gs-graph $G_0 \otimes G_1 : u_0u_1 \to v_0v_1 = G_0 \cup G_1$ for which we have $ic(G_0 \otimes G_1) = ic(G_0)ic(G_1)$ and $oc(G_0 \otimes G_1) = oc(G_0)oc(G_1)$.



(a) A gs-graph $G : \bullet\circ \to \bullet^2\circ^2$



(b) A pure bigraph $G : \langle 1, \{x\}\rangle \to \langle 2, \{y, z\}\rangle$



(c) Correspondence between Fig. 3(a) and 3(b) made explicit

Figure 3. Different graphical models for the same structure

Note that, if a composition of abstract bigraphs is sound at the level of sorts, it is always possible to find suitable concrete gs-graphs in their equivalence classes that are composable, so that the notion of sequential and parallel composition extend to abstract gs-graphs. For example, if $G_0 : u_0 \to v_0$ and $G_1 : v_1 \to w_1$ we can always find suitable concrete gs-graphs G'_0 and G'_1 in their equivalence classes such that $G'_0 \otimes G'_1$ is defined. Moreover, any other choice of composable concrete gs-graphs G''_0 and G''_1 in the equivalence classes of G_0 and G_1 is such that $G''_0 \otimes G''_1$ is isomorphic to $G'_0 \otimes G'_1$.

Example 2.19. The arrow e_4 from Example 2.7 corresponds to the (concisely represented) gs-graph $\{\ x_5 := \nu\ ,\ p_6 := f(p_1, x_5)\ ,\ p_7 := g(p_2, x_5, x_4)\ ,\ p_8 := h(p_7, x_3)\ ,\ p_9 := h(p_7, x_3)\ ,\ x_{10} := x_5\ ,\ !(p_8)\ ,\ !(p_9)\ \}$, which is illustrated in Fig. 3(a) (see Fig. 2 for comparison).

To deal with idle edges we introduce the notion of amended gs-graph for ν-signatures.

Definition 2.20. (amended gs-graphs)
A concrete gs-graph over a ν-signature is an *amended gs-graph* if whenever there is a name x such that G contains both the assignments $x := \nu$ and $!(x)$, then there is at least another assignment using x.

It is immediate to see that any atomic gs-graph is also an amended gs-graph and that the parallel composition of amended gs-graphs is also an amended gs-graph. For the sequential composition of amended gs-graphs we introduce a simplification that is analogous to the axiom $!_\circ \ ; \ \nu = id_\epsilon$: we remove from $G; H$ all the assignments of the form $x := \nu$ and $!(x)$ such that x is not used in $(G; H) \setminus \{x := \nu, !(x)\}$. To avoid ambiguity, we denote by $G; ; H$ the sequential composition of amended gs-graphs.

Proposition 2.21. Amended gs-graphs over a ν-signature Σ form the arrows of the (freely generated) νgs-monoidal theory over Σ.

Proof:
It is a routine check to verify that amended gs-graphs over Σ form the arrows of a νgs-monoidal theory over Σ.

To prove that amended gs-graphs form an initial model, we need to show that any two different arrows $t, t' : u \to v$ of the νgs-monoidal theory over Σ are mapped to different amended gs-graphs. By the result in [10] we know that gs-graphs over a ν-signature Σ form the arrows of the freely generated gs-monoidal theory over Σ. By the property of νgs-monoidal theories we know that any such term $t : u \to v$ can be decomposed as $t = t_0; (id_v \otimes t_1)$, where t_0 is a term (in gs-monoidal normal form) with no occurrence of ν and t_1 has the form $\nu \otimes \cdots \otimes \nu$. Let us consider t_0 and t_1 as terms of gs-monoidal theories and let G_0 be the gs-graph associated with t_0 and G_1 the gs-graph associated with $id_v \otimes t_1$. It is immediate to check that G_0 and G_1 are amended gs-graphs and therefore $G_0; ; G_1$ is also an amended gs-graph.

Now let us consider $t, t' : u \to v$ as above and their decompositions $t = t_0; (id_v \otimes t_1)$ and $t' = t'_0; (id_v \otimes t'_1)$ and suppose they are mapped to the same amended gs-graph $G = G_0; ; G_1 = G'_0; ; G'_1$, while either $G_0 \neq G'_0$ or $G_1 \neq G'_1$ or both. Then $G_0; G_1 \neq G'_0; G'_1$, but $G_0; ; G_1 = G'_0; ; G'_1$, meaning that during the composition we have removed one or more assignments $x := \nu, !(x)$. But then $t = t'$ by applying the corresponding axioms $!_\circ; \nu = id_\epsilon$ that leads us to a contradiction.

For proving surjectivity, we notice that every amended gs-graph is also a gs-graph. As such, it is expressible as the sequential composition of certain atomic assignments, which are all constants of the algebra. Thus also their composition is expressible as a term of the algebra.

□

2.3. From pure signatures to pure bigraphs

Two main classes of bigraphs have been developed: pure bigraphs [14] and binding bigraphs [22]. In pure bigraphs a node is not allowed to declare a local name, and nodes communicate using only their global ports. Binding bigraphs are discussed in Section 4.

Definition 2.22. (pure signature)
A *pure signature* consists of a set $\mathcal{K}$ whose elements, called *controls*, specify the role of system nodes and a function $\mathsf{ar} : \mathcal{K} \to \mathbb{N}$ that assigns an arity to each control, i.e., the number of its ports.

2.3.1. Place graphs

A place graph is essentially a forest of unordered trees; it represents the locality of nodes.

Definition 2.23. (place graph)
Let $\mathcal{K}$ be a pure signature and m, n be a pair of ordinals, then a place graph $P : m \to n$ is a triple $(V, \mathsf{cl}, \mathsf{pt})$ where V is a finite set of nodes called the *support* of P (also denoted $|P|$), $\mathsf{cl} : V \to \mathcal{K}$ is the *control map* assigning a control to each node and $\mathsf{pt} : m \uplus V \to V \uplus n$ is the *parent map* that describes the location of the nodes. The parent map is acyclic in the sense that for each $v \in V$, $\mathsf{pt}^k(v) = v$ implies $k = 0$ (where we remind that pt^k denotes the application of pt for k times).

Two special cases of place graphs are identities and symmetries. The identity place graph at m is $id_m \triangleq (\emptyset, \emptyset_{\mathcal{K}}, Id_m) : m \to m$. Symmetries $\gamma_{m,n}$ have empty support and their effect is to swap sites:

$$\gamma_{m,n} \triangleq (\emptyset, \emptyset_{\mathcal{K}}, \mathsf{pt}(j) = \begin{cases} j + n & \text{if } j < m \\ j - m & \text{if } j \geq m \end{cases})$$

Definition 2.24. (place graph: composition)
Let $P : m \to n$ and $Q : n \to l$ two place graphs with $V_P \cap V_Q = \emptyset$, then their composition is defined as $Q \circ P \triangleq (V_P \uplus V_Q, \mathsf{cl}_P \uplus \mathsf{cl}_Q, \mathsf{pt}) : m \to l$, where for each $v \in m \uplus V_P \uplus V_Q$,

$$\mathsf{pt}(v) \triangleq \begin{cases} \mathsf{pt}_P(v) & \text{if } v \in m \uplus V_P \text{ and } \mathsf{pt}_P(v) \in V_P \\ \mathsf{pt}_Q(i) & \text{if } v \in m \uplus V_P \text{ and } \mathsf{pt}_P(v) = i \in n \\ \mathsf{pt}_Q(v) & \text{if } v \in V_Q \end{cases}$$

Definition 2.25. (place graph: tensor product)
The tensor product $\otimes$ on interfaces is the addition of ordinals and the unit object is 0. For $i \in \{0, 1\}$ let $P_i = (V_i, \mathsf{cl}_i, \mathsf{pt}_i) : m_i \to n_i$ two place graphs having disjoint supports. Then

$$P_0 \otimes P_1 \triangleq (V_0 \uplus V_1, \mathsf{cl}_0 \uplus \mathsf{cl}_1, \mathsf{pt}) : m_0 + m_1 \to n_0 + n_1$$

where for each $j \in m_0 + m_1$

$$\mathsf{pt}(j) \triangleq \begin{cases} \mathsf{pt}_0(j) & \text{if } j < m_0 \\ \mathsf{pt}_1(j - m_0) & \text{if } j \geq m_0 \text{ and } \mathsf{pt}_1(j - m_0) \in V_1 \\ \mathsf{pt}_1(j - m_0) + n_0 & \text{if } j \geq m_0 \text{ and } \mathsf{pt}_1(j - m_0) \in n_1 \end{cases}$$

and for each $v \in V_0 \uplus V_1$

$$\mathsf{pt}(v) \triangleq \begin{cases} \mathsf{pt}_0(v) & \text{if } v \in V_0 \\ \mathsf{pt}_1(v) & \text{if } v \in V_1 \text{ and } \mathsf{pt}_1(v) \in V_1 \\ \mathsf{pt}_1(v) + n_0 & \text{if } v \in V_1 \text{ and } \mathsf{pt}_1(v) \in n_1 \end{cases}$$

2.3.2. Link graphs

A link graph is a graph expressing connectivity: an edge represents, e.g., a communication medium.

Definition 2.26. (link graph)
Given a pure signature $\mathcal{K}$ a *link graph* $L = (V, E, \mathsf{cl}, \mathsf{lk}) : X \to Y$, where X and Y are the sets respectively of *inner* and *outer names*, has finite sets of nodes V and edges E, a control map $\mathsf{cl} : V \to \mathcal{K}$ and a *link map* $\mathsf{lk} : X \uplus \mathsf{Ps}_L \to E \uplus Y$ with $\mathsf{Ps}_L \triangleq \sum_{v \in V} \mathsf{ar}(\mathsf{cl}(v))$ the set of ports of L. The *support* of L is $|L| \triangleq V \uplus E$.

The identity link graph at X is $id_X \triangleq (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, Id_X) : X \to X$. The unit object is the empty set $\emptyset$ and the symmetries $\gamma_{X,Y}$ are just identities: $\gamma_{X,Y} \triangleq id_{X \uplus Y}$.

Definition 2.27. (link graph: composition)
Let $L : X \to Y$ and $M : Y \to Z$ be two link graphs such that $V_L \cap V_M = E_L \cap E_M = \emptyset$, then their composition is defined as: $M \circ L \triangleq (V_L \uplus V_M, E_L \uplus E_M, \mathsf{cl}_L \uplus \mathsf{cl}_M, \mathsf{lk}) : X \to Z$, where for each $p \in X \uplus \mathsf{Ps}_L \uplus \mathsf{Ps}_M$ of $M \circ L$,

$$\mathsf{lk}(p) \triangleq \begin{cases} \mathsf{lk}_L(p) & \text{if } p \in X \uplus \mathsf{Ps}_L \text{ and } \mathsf{lk}_L(p) \in E_L \\ \mathsf{lk}_M(y) & \text{if } p \in X \uplus \mathsf{Ps}_L \text{ and } \mathsf{lk}_L(p) = y \in Y \\ \mathsf{lk}_M(p) & \text{if } p \in \mathsf{Ps}_M \end{cases}$$

The product $\otimes$ is roughly disjoint set union.

Definition 2.28. (link graph: tensor product)
Suppose that for $i \in \{0, 1\}$, $L_i = (V_i, E_i, \mathsf{cl}_i, \mathsf{lk}_i) : X_i \to Y_i$ are link graphs with disjoint supports and with $X_0 \cap X_1 = Y_0 \cap Y_1 = \emptyset$. Their product is:

$$L_0 \otimes L_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, \mathsf{cl}_0 \uplus \mathsf{cl}_1, \mathsf{lk}_0 \uplus \mathsf{lk}_1) : X_0 \uplus X_1 \to Y_0 \uplus Y_1$$

2.3.3. Pure bigraphs

The key point of bigraphs is that their place and link graphs are constructed separately; therefore the locality of a node and its connectivity can not interfere.

Definition 2.29. (concrete pure bigraph)
A *concrete (pure) bigraph* $G = (V, E, \mathsf{cl}, \mathsf{pt}, \mathsf{lk}) : \langle m, X\rangle \to \langle n, Y\rangle$ consists of a place graph $G^P = (V, \mathsf{cl}, \mathsf{pt}) : m \to n$ and a link graph $G^L = (V, E, \mathsf{cl}, \mathsf{lk}) : X \to Y$.

We sometimes write $G = \langle G^P, G^L\rangle$ and denote the *support* of G by $|G| \triangleq V_G \uplus E_G$.

Example 2.30. An example of pure bigraph is in Fig. 3(b). The place graph is represented through the nesting of nodes, while the arcs pertain to the link graph. The interface is given by pairing the interfaces of the place graph and of the link graph. The outer interface of a place graph specifies the number of distinct components forming the bigraph; to each component corresponds a root displayed with an enclosing dashed box. In the example we have two roots (numbered 0 and 1). The inner interface consists of the holes of the place graph, called sites, that serve to compose with other place graphs. Our example has one hole (numbered 0), displayed with a grey box. For the link graph, outer names are displayed on the top (y and z), and inner names on the bottom (x).

The identity bigraph at $\langle m, X\rangle$ is $id_{\langle m,X\rangle} \triangleq \langle id_m, id_X\rangle$. The unit object $\epsilon \triangleq \langle id_0, \emptyset\rangle$ is the pairing of the unit objects of the place graph and link graph products, and the symmetry is $\gamma_{\langle m,X\rangle,\langle n,Y\rangle} \triangleq \langle \gamma_{m,n}, \gamma_{X,Y}\rangle$.

Definition 2.31. (bigraph: composition)
Given two concrete bigraphs $G : \langle m, X\rangle \to \langle n, Y\rangle$ and $H = \langle n, Y\rangle \to \langle l, Z\rangle$ with $V_G \cap V_H = E_G \cap E_H = \emptyset$, their composition is defined component-wise: $H \circ G \triangleq \langle H^P \circ G^P, H^L \circ G^L\rangle : \langle m, X\rangle \to \langle l, Z\rangle$.

Definition 2.32. (bigraph: tensor product)
Given two bigraph interfaces $\langle m, X\rangle$ and $\langle n, Y\rangle$ with $X \cap Y = \emptyset$, the product is defined componentwise: $\langle m, X\rangle \otimes \langle n, y\rangle = \langle m \otimes n, X \otimes Y\rangle$. The same happens on bigraphs: for $i \in \{0, 1\}$ let $G_i = \langle G^P, G^L\rangle : \langle m_i, X_i\rangle \to \langle n_i, Y_i\rangle$ be two bigraphs with disjoint supports and $X_0 \cap X_1 = Y_0 \cap Y_1 = \emptyset$, then

$$G_0 \otimes G_1 \triangleq \langle G_0^P \otimes G_1^P, G_0^L \otimes G_1^L\rangle : \langle m_0 + m_1, X_0 \uplus X_1\rangle \to \langle n_0 + n_1, Y_0 \uplus Y_1\rangle$$

Definition 2.33. (support equivalence for bigraphs)
Given two bigraphs $G, H : \langle m, X\rangle \to \langle n, Y\rangle$, a support equivalence $\rho : |G| \to |H|$ is a pair of bijections $\rho_V : V_G \to V_H$ and $\rho_E : E_G \to E_H$ such that: $\mathsf{cl}_H \circ \rho_V = \mathsf{cl}_G$, $\mathsf{pt}_H \circ (Id_m \uplus \rho_V) = (Id_n \uplus \rho_V) \circ \mathsf{pt}_G$ and $\mathsf{lk}_H \circ (Id_X \uplus \rho_{\mathsf{Ps}}) = (Id_Y \uplus \rho_E) \circ \mathsf{lk}_G$, where $\rho_{\mathsf{Ps}} : \mathsf{Ps}_G \to \mathsf{Ps}_H$ maps the ports of G in those of H and it is defined in terms of ρ_V: for each port $(v, i) \in \mathsf{Ps}_G$, $\rho_{\mathsf{Ps}}(v, i) = (\rho_V(v), i)$.

Definition 2.34. (lean bigraph)
A bigraph is *lean* if it has no idle edges (i.e., edges that are linked to nothing).

We write $G \doteq H$ when G and H are *support equivalent*, and $G \,\hat{\doteq}\, H$ if they are support equivalent ignoring their idle edges (*lean-support equivalence*). The lean-support quotient yields the (partial) symmetric monoidal category of *abstract bigraphs*, denoted by $BG(\mathcal{K})$, and the lean-support quotient functor $[\cdot]$ maps each concrete bigraph in its corresponding abstract bigraph.

Definition 2.35. (abstract bigraphs)
An *abstract bigraph* over $\mathcal{K}$ is a $\hat{\doteq}$-equivalence class $[G] : \langle m, X\rangle \to \langle n, Y\rangle$ of a concrete bigraph G.

Remark 2.36. Lean support equivalence over bigraphs roughly corresponds to garbage-collecting restricted names that are not used and it is convenient for representing process calculi whenever the structural axiom $(\nu\ x)\mathbf{nil} = \mathbf{nil}$ is considered, for $\mathbf{nil}$ the empty process and $(\nu\ x)$ the restriction operator for defining a private name x (whose scope is P in the expression $(\nu\ x)P$).

3. Characterising Pure Bigraphs

This section shows that pure bigraphs are essentially equivalent to a particular class of gs-graphs over ν-signatures. The word “essentially” means that there is a one-to-one homomorphism (w.r.t. ; and $\otimes$) between the objects of the two models, but only up to certain bijections between the interfaces. Indeed the main difference lies in how the two models view interfaces: a bigraph interface is a pair $\langle m, X\rangle$ of an ordinal and a set of names, a gs-graph interface is a string $u \in \{\bullet, \circ\}^*$. To make them comparable, we need to equip each model with some missing information which is present in the other model. In a bigraphical interface $\langle m, X\rangle$ we must form a list out of the elements in $\{0, \dots, m-1\}$ and the elements in X. For the gs-graphs instead, given a string $u \in \{\bullet, \circ\}^*$ we have to assign a name to each element of sort $\circ$.

Example 3.1. The relation between pure bigraphs and gs-graphs can be sketched by looking at Fig. 3. Places correspond to edges and their hierarchy is built in the gs-graph by exploiting the nodes of sort $\bullet$. Connectivity is represented by the sharing of nodes of sort $\circ$. Closed links are the $\circ$ nodes below a restriction ν. The dashed lines express which nodes are exported to the interfaces.

The correspondence between the gs-graphs in Fig. 3(a) and the bigraph in Fig. 3(b) is made explicit in Fig. 3(c), where the elements of the bigraph have been decorated with the corresponding names of the nodes in the gs-graph.

The difference between the two graphical representations resembles the distinction between the common views offered by file managers: the tree-like view is similar to the gs-graphs representation, while the icon view is similar to the bigraphs representation.

Signatures. Both bigraphs and gs-graphs have a signature giving the allowed operators. Therefore it is necessary to correlate the two kinds of signature. (In the rest of this section we understand that only pure signatures are considered and we use the symbol $\mathcal{K}$ also for gs-graph signatures.)

Definition 3.2. (equating signatures)
Consider a pure signature $\mathcal{K}$; the equivalent gs-graph ν-signature Σ has an operator $K : \bullet \to \bullet\circ^h$ if and only if the control K of arity h is in $\mathcal{K}$.

Interfaces. Given a bigraphical interface $\langle m, X\rangle$, every $i \in m$ is of sort $\bullet$ while the names in X are of sort $\circ$. Nevertheless if we had a gs-graph interface with exactly m elements of sort $\bullet$ and $|X|$ elements of sort $\circ$ we would not have an obvious way to map such interface in $\langle m, X\rangle$, because the $\circ$ elements are ordered unlike the names in X.

Example 3.3. Take for example the interfaces of our running example. We need to find a match between $u = \bullet\circ$ and $\langle 1, \{x\}\rangle$ as well as a match between $v = \bullet^2\circ^2$ and $\langle 2, \{y, z\}\rangle$. For the former there is a unique way to fix the correspondence. For the latter, assuming any ambiguity is solved at the level of locations because the order of locations is preserved, there are two possible bijections from $\circ^2$ to $\{y, z\}$ but only one allows to establish a correct correspondence (y must match the first $\circ$ and z the second $\circ$). On the contrary, if we knew that $y < z$, the natural way would be that of choosing the right bijection.

To improve the correspondence at the level of interfaces, we introduce a total order on the names used in the bigraphical interfaces.

Definition 3.4. Let $\mathscr{X}$ be a denumerable infinite totally ordered (by $\leq$) set of names. Given a pure signature $\mathcal{K}$, we take bigraphs in which the sets of names on the interfaces are replaced by lists of names ordered through $\leq$. Given a list L we denote by $L[j]$ the $(j + 1)^{th}$ element of the list.

The total order on names makes the two type interfaces more similar, but it is not sufficient to establish a bijective relation between them.

Example 3.5. Consider the previous example and suppose that in $\mathscr{X}$ we have $y < x < z$. The interface I can be associated, through the unique monotone bijection, to the bigraphical interface with the set $\{y, z\}$, but nothing prevents one from using $\{x, z\}$ or $\{x, y\}$ instead.

These considerations lead us to the following definition that embeds a particular set of names in a gs-graph interface.

Definition 3.6. (name choice)
Let $u \in \{\bullet, \circ\}^*$ and let $\#_u$ be the number of elements of sort $\circ$ in u. Then a *name choice* for u is an injective monotone map $\sigma_u : \#_u \to \mathscr{X}$. A gs-graph $G : u \to v$ can be equipped with two name choices σ_u, σ_v for the inner and the outer interfaces, written $G : (u, \sigma_u) \to (v, \sigma_v)$.

With the name choices, exactly one bigraphical interface corresponds to a string over $\{\bullet, \circ\}$, but the converse is false. Indeed a bigraphical interface over ordered names can be viewed as a concatenation of two ordered lists, the first with elements of sort $\bullet$ and the second with $\circ$-elements, while in a gs-graph interface such elements are mixed.

Definition 3.7. (shuffling function)
Let $\langle m, X\rangle$ a bigraphical interface. A *shuffling function* for $\langle m, X\rangle$ is a bijection, $\phi : m+|X| \to m+|X|$ that preserve the relative order of the elements with the same sort, i.e., $\forall i, j \in m+|X|$ if $0 \le i \le j < m$ or $m \le i \le j < m + |X|$ then $\phi_{in}(i) \le \phi_{in}(j)$.

Given a bigraph we add two bijections that "shuffle" the elements in its interface as stated below.

Definition 3.8. (shuffled bigraphs)
A *shuffled bigraph* $\langle G, \phi_{in}, \phi_{out}\rangle$ consists of a bigraph $G : \langle m, X\rangle \to \langle n, Y\rangle$ and two shuffling function, $\phi_{in} : m + |X| \to m + |X|$ and $\phi_{out} : n + |Y| \to n + |Y|$.

In the following, we denote a shuffled bigraph $\langle G, \phi_{in}, \phi_{out}\rangle$ with $G : \langle m, X\rangle \to \langle n, Y\rangle$ just by writing $G : \langle m, X, \phi_{in}\rangle \to \langle n, Y, \phi_{out}\rangle$.

3.1. From shuffled bigraphs to gs-graphs: the transformation $S[\![\cdot]\!]$

The first transformation $S[\![\cdot]\!]$ that we define takes a shuffled bigraph $G = \langle V, E, \mathsf{cl}, \mathsf{pt}, \mathsf{lk}\rangle : \langle m, X, \phi_{in}\rangle \to \langle l, Y, \phi_{out}\rangle$ and returns a gs-graph $H = S[\![G]\!]$ that represents it. The underlying idea is relatively simple: there is a proper assignment for each node and edge of the bigraph. In detail the edges cause the creation of assignments with the ν operator, while the nodes give assignments that describe the position of a control in the system and the interactions with the other controls, deriving it from the parent and the link map of the bigraph.

In the following we denote with $\mathcal{N}_H$ the set of all names appearing in H, which is partitioned in $\mathcal{N}_H^\bullet$ and $\mathcal{N}_H^\circ$, respectively the set of all names of sort $\bullet$ and of sort $\circ$, appearing in H. Let $\mathcal{N}_H^\bullet = V \uplus \{s_0, \ldots, s_{m-1}\} \uplus \{r_0, \ldots, r_{l-1}\}$ and $\mathcal{N}_H^\circ = E \uplus \{x_0, \ldots, x_{|X|-1}\} \uplus \{y_0, \ldots, y_{|Y|-1}\}$. Note that x_i and y_j are not the names in sets X and Y, but new names used only in the gs-graph. First we need two auxiliary functions $\overline{\mathsf{pt}} : m \uplus V \to \mathcal{N}_H^\bullet$ and $\overline{\mathsf{lk}} : \mathsf{Ps}_G \uplus X \to \mathcal{N}_H^\circ$ that translate the results of pt and lk into the names of the gs-graph:

$$\overline{\mathsf{pt}}(v) \triangleq \begin{cases} w & \text{if } \mathsf{pt}(v) = w \in V \\ r_i & \text{if } \mathsf{pt}(v) = i \in l \end{cases} \qquad \overline{\mathsf{lk}}(p) \triangleq \begin{cases} e & \text{if } \mathsf{lk}(p) = e \in E \\ y_i & \text{if } \mathsf{lk}(p) = Y[i] \end{cases}$$

Definition 3.9. (the transformation $S[\![\cdot]\!]$)
Using the notation introduced above, we let the gs-graph $S[\![G]\!]$ associated to the shuffled bigraph $G = \langle V, E, \mathsf{cl}, \mathsf{pt}, \mathsf{lk}\rangle : \langle m, X, \phi_{in}\rangle \to \langle l, Y, \phi_{out}\rangle$ be the set that includes all and only the following assignments:

1. $\forall v \in V$ we let $v := f(\overline{\mathsf{pt}}(v), \overline{\mathsf{lk}}(v, 0), \ldots, \overline{\mathsf{lk}}(v, h-1))$ where $f = \mathsf{cl}(v)$ and h is the arity of f;
2. $\forall e \in E$ we let $e := \nu$;
3. $\forall i \in m$ we let $s_i := \overline{\mathsf{pt}}(i)$;
4. $\forall i \in |X|$ we let $x_i := \overline{\mathsf{lk}}(X[i])$

together with the name choices $\sigma_{in}(i) \triangleq X[i]$ for $i \in |X|$ and $\sigma_{out}(i) \triangleq Y[i]$ for $i \in |Y|$. Note that $ic(H) = \{s_0, \ldots, s_{m-1}\} \cup \{x_0, \ldots, x_{|X|-1}\}$ and $oc(H) = \{r_0, \ldots, r_{l-1}\} \cup \{y_0, \ldots, y_{|Y|-1}\}$. The order of these names can be retrieved using the shuffling functions (see Definition A.1).

Proposition 3.10. (idle edges)
If G has no idle edges, then $S[\![G]\!]$ is an amended gs-graph.

Proof:
By contradiction, assume there is a name x such that $S[\![G]\!]$ contains both the assignments[5] $x := \nu$ and $!(x)$, but there is no other assignment using x. Then by definition of $S[\![G]\!]$ it must be the case that $x \in E_G$ is an idle edge of G, leading to a contradiction. □

Before proving that $S[\![\cdot]\!]$ preserves the operations for composing bigraphs (see Proposition 3.13), we enunciate a couple of technical lemmas that highlight some properties of the overlined maps, $\overline{\mathsf{pt}}$ and $\overline{\mathsf{lk}}$, defined above. The first one is effectively an alternative definition of the overlined maps. The second one is particularly useful in showing that $S[\![\cdot]\!]$ preserves composition.

Lemma 3.11. Let $\Upsilon^P : n \to \{r_0, \ldots, r_{n-1}\}$ such that $\Upsilon^P(i) = r_i$ and $\Upsilon^L : Y \to \{y_0, \ldots, y_{|Y|-1}\}$ such that $\Upsilon^L(i) = Y[i]$. Then, the overlined maps $\overline{\mathsf{pt}}$ and $\overline{\mathsf{lk}}$ can be decomposed as follows:

$$\overline{\mathsf{pt}} = (Id_V \uplus \Upsilon^P) \circ \mathsf{pt} \qquad \overline{\mathsf{lk}} = (Id_E \uplus \Upsilon^L) \circ \mathsf{lk}$$

With the second lemma an overlined map relative to a composition $G; H$ of two bigraphs can be expressed using the overlined map of H and the non-overlined map of G.

Lemma 3.12. Consider two bigraphs $G_0 : \langle m, X \rangle \to \langle n, Y \rangle$ and $G_1 : \langle n, Y \rangle \to \langle l, Z \rangle$ such that $G_1 \circ G_0$ is defined. Then:

$$\overline{\mathsf{pt}_{G_1 \circ G_0}} = (Id_{V_0} \uplus \overline{\mathsf{pt}_1}) \circ (\mathsf{pt}_0 \uplus Id_{V_1}) \qquad \overline{\mathsf{lk}_{G_1 \circ G_0}} = (Id_{E_0} \uplus \overline{\mathsf{lk}_1}) \circ (\mathsf{lk}_0 \uplus Id_{\mathsf{Ps}_1})$$

Proposition 3.13. (**$S[\![\cdot]\!]$ preserves composition and tensor product**)
Suppose that $G : \langle m, X, \phi_{in} \rangle \to \langle n, Y, \psi \rangle$, and $G' : \langle n, Y, \psi \rangle \to \langle l, Z, \phi_{out} \rangle$ are two shuffled bigraphs. We have that $S[\![G' \circ G]\!] = S[\![G]\!]; S[\![G']\!]$.

Now consider $G_0 : \langle m_0, X_0, \phi_0 \rangle \to \langle n_0, Y_0, \psi_0 \rangle$ and $G_1 : \langle m_1, X_1, \phi_1 \rangle \to \langle n_1, Y_1, \psi_1 \rangle$ with $X_0 < X_1$ and $Y_0 < Y_1$, then $S[\![G_0 \otimes G_1]\!] = S[\![G_0]\!] \otimes S[\![G_1]\!]$.

[5]Remind that by Remark 2.15 the assignment $!(x)$ is implicit for all x.

Proof:
The sequential composition of gs-graphs $S[\![G]\!]; S[\![G']\!]$ requires to choose the same names for the outer connection of $S[\![G]\!]$ and for the inner connection $S[\![G']\!]$. We denote such names with $t_0, \ldots, t_{n-1}$ and $b_0, \ldots, b_{|Y|-1}$ respectively of sort $\bullet$ and $\circ$.

The set of assignments A that we must remove from $S[\![G']\!]$ is defined as follows:

$$A = \{t_i := \overline{\mathsf{pt}_{G'}}(i) \mid i \in \{0, \ldots, n-1\}\} \;\cup\; \{b_i := \overline{\mathsf{lk}_{G'}}(Y[i]) \mid i \in \{0, \ldots, |Y|-1\}\}$$

Denoting with σ the substitution induced by A we have that

$$S[\![G]\!]; S[\![G']\!] = S[\![G]\!]\sigma \;\cup\; S[\![G']\!]\backslash A$$

From the previous relation we note that in $S[\![G]\!]$ there will be assignments in which the name used are defined through $\overline{\mathsf{pt}_{G'}}$ instead of $\overline{\mathsf{pt}_G}$, while in $S[\![G']\!]$ there are not such problems. Then take an assignment of $S[\![G]\!]$ that makes use of t_i: for simplicity we suppose that this assignment is an auxiliary one, $s_j := t_i$, since the case of a proper assignment is treated in exactly the same way. The name t_i is substituted in the composition by the name $\overline{\mathsf{pt}'_G}(i)$. Moreover since j is also a site of the bigraph $G; G'$, in $S[\![G; G']\!]$ we have an assignment $s_j := \overline{\mathsf{pt}_{G;G'}}(j)$. We need to show that $\overline{\mathsf{pt}_{G;G'}}(j) = \overline{\mathsf{pt}'_G}(i)$ and for this purpose we use the result of Lemma 3.12. We know that $s_j := t_i$ if and only if $\mathsf{pt}_G(j) = i$, thus we can rephrase the assignment in the composition as $s_j := \overline{\mathsf{pt}'_G}(\mathsf{pt}_G(j))$, but

$$\overline{\mathsf{pt}_{G;G'}}(j) = ((Id_{V_G} \uplus \overline{\mathsf{pt}_{G'}}) \circ (\mathsf{pt}_G \uplus Id_{V_{G'}}))(j) = (Id_{V_G} \uplus \overline{\mathsf{pt}_{G'}})(\mathsf{pt}_G(j)) = \overline{\mathsf{pt}_{G'}}(\mathsf{pt}_G(j))$$

as required.

The case of assignments that use names b_i is similar: instead of the parent map we have the link map and instead of the sites we have names. Finally we comment on the restricted assignments: if $e := \nu$ in $S[\![G]\!]; S[\![G']\!]$ then $e \in E_G \uplus E'_G$ and consequently $e := \nu$ is also an assignment of $S[\![G; G']\!]$. The reverse argument is still valid thus $S[\![G]\!]; S[\![G']\!]$ and $S[\![G; G']\!]$ have the same restricted assignments.

The proof for the case of tensor product can be found in the Appendix (see Section A.1). □

Remark 3.14. When G and G' have no idle edges, the proof carries over amended gs-graphs with sequential composition ;; only up to lean support equivalence, in the sense that $S[\![G]\!]; S[\![G']\!] = S[\![H]\!]$ with $H \mathrel{\hat{\approx}} G' \circ G$ (i.e., we take as H the bigraph obtained by removing all idle edges from $G; G'$).

3.2. From gs-graphs to shuffled bigraphs: the transformation $B[\![\cdot]\!]$

Let $H : (u, \sigma_u) \to (v, \sigma_v)$ be a gs-graph over $\mathcal{K}$ with name choices and let us take an instance in its isomorphism class. The first step in transforming the gs-graph H in the corresponding shuffled bigraph $G = B[\![H]\!]$ consists in defining the shuffle functions ϕ_{in} and ϕ_{out}. For this purpose let m and l be the number of elements of sort $\bullet$ in the lists u and v respectively; then for each list, for example u, define a function $u^\bullet : m \to |u|$ that tell us the positions in the list u of the $\bullet$ sort elements, and a similar function $u^\circ : (|u| - m) \to |u|$ that do the same thing on the elements of u of sort $\circ$. For example if $u = \bullet \circ \circ \bullet$, then $u^\bullet = \{0 \mapsto 0, 1 \mapsto 3\}$ and $u^\circ = \{0 \mapsto 1, 1 \mapsto 2\}$. With the aid of this functions we define $\phi_{in} : |u| \to |u|$ and $\phi_{out} : |v| \to |v|$ as:

$$\phi_{in}(i) \triangleq \begin{cases} u^\bullet(i) & \text{if } 0 \le i < m \\ u^\circ(i-m) & \text{otherwise} \end{cases} \qquad \phi_{out}(i) \triangleq \begin{cases} v^\bullet(i) & \text{if } 0 \le i < l \\ v^\circ(i-l) & \text{otherwise} \end{cases}$$

Now recall that in a pure ν-signature every operator, except ν, is of the form $f : \bullet \to \bullet\circ^h$ for some $h \in \mathbb{N}$, thus every proper assignment over those operators takes the form $n := f(n_\bullet, n_0, \ldots, n_{h-1})$ with $n, n_\bullet$ of sort $\bullet$ and the remaining names of sort $\circ$.

Definition 3.15. (the transformation $B[\![\cdot]\!]$)
The bigraph associated to the gs-graph $H : (u, \sigma_u) \to (v, \sigma_v)$ is $B[\![H]\!] = (V, E, \mathsf{cl}, \mathsf{pt}, \mathsf{lk}) : \langle m, X, \phi_{in}\rangle \to \langle l, Y, \phi_{out}\rangle$ where $m, l, \phi_{in}, \phi_{out}$ are defined as above, and:

1. $X[i] \triangleq \sigma_u(i)$ for each admissible i and $Y[j] \triangleq \sigma_v(j)$ for each admissible j.

2. $V \triangleq \{n \in \mathcal{N}_H^\bullet \mid n \notin ic(H) \text{ and } n \notin oc(H)\}$ is composed by the $\bullet$-names that are not visible outside the gs-graph. Thus the names in V are assigned exactly once with a proper assignment.

3. $E \triangleq \{n \in \mathcal{N}_H^\circ \mid n := \nu \in H\}$ comprises all "restricted" names of sort $\circ$.

4. The control map $\mathsf{cl} : V \to \mathcal{K}$ is defined as follows. Being $n \in V$ let $n := f(n_\bullet, n_0, \ldots, n_{h-1})$ the unique assignment of n in H, then $\mathsf{cl}(n) = f$

5. The parent map $\mathsf{pt} : m \uplus V \to V \uplus l$ is defined separately for the elements in V and m. For each $n \in V$ let $n := f(n_\bullet, n_0, \ldots, n_{h-1})$ the unique assignment of n in H, then:

$$\mathsf{pt}(n) = \begin{cases} n_\bullet & \text{if } n_\bullet \in V \\ \phi_{out}^{-1}(j) & \text{if } n_\bullet = oc(H)[j] \text{ for some } j \text{ in the list range} \end{cases}$$

   Take instead $i \in m$ and let $s_i = ic[\phi(i)]$. Since s_i is an inner connection, there exists in H a unique auxiliary assignment $s_i := n$.

$$\mathsf{pt}(i) = \begin{cases} n & \text{if } n \in V \\ \phi_{out}^{-1}(j) & \text{if } n = oc(H)[j] \text{ for some admissible } j \end{cases}$$

6. Finally, the link map $\mathsf{lk} : \mathsf{Ps}_{B[\![H]\!]} \uplus X \to E \uplus Y$ is defined as follows. Take a port (n, i) with $n \in V$ and let $n := f(n_\bullet, n_0, \ldots, n_{h-1})$ be the proper assignment of n, then

$$\mathsf{lk}((n,i)) = \begin{cases} n_i & \text{if } n_i \in E \\ Y[\phi_{out}^{-1}(j) - l] & \text{if } n_i = oc(H)[j] \text{ for some admissible } j \end{cases}$$

   Consider instead a name $x = X[i]$ and let $\overline{x} = ic(H)[\phi_{in}(m + i)]$. The auxiliary assignment associated to $\overline{x}$ is $\overline{x} := n$. Thus

$$\mathsf{lk}(x) = \begin{cases} n & \text{if } n \in E \\ Y[\phi_{out}^{-1}(i) - l] & \text{if } n = oc(H)[j] \text{ for some } j \end{cases}$$

Proposition 3.16. (idle edges)
If H is an amended gs-graph, then $B[\![H]\!]$ has no idle edges.

Proof:
Immediate, by definition of $B[\![H]\!]$. □

We can now present our first main correspondence result, whose proof shows shuffled support-equivalent bigraphs over a pure signature $\mathcal{K}$ are isomorphic to gs-graphs over $\mathcal{K}$ (seen as a ν-signature) with name choices.

Theorem 3.17. The transformation $S[\![\cdot]\!]$ is a bijection between shuffled support-equivalent bigraphs and gs-graphs over $\mathcal{K}$ and its inverse is $B[\![\cdot]\!]$.

Proof:
[$B[\![S[\![G]\!]]\!] = G$]. The first part of the proof shows that $B[\![S[\![\cdot]\!]]\!]$ is the identity function on shuffled bigraphs. Consider a general shuffled bigraph $G : \langle m, X, \phi_{in}, \rangle \to \langle n, Y, \phi_{out}\rangle$ and apply on it the transformation $H = S[\![G]\!]$. We let $G' = B[\![S[\![G]\!]]\!]$. Following the definition of $B[\![\cdot]\!]$, the first step consists in retrieving the shuffle functions ϕ'_{in} and ϕ'_{out}. Before doing that we must consider a little better what are the interfaces of the gs-graph H. We begin with the inner interface: $ic(H)$ is composed by exactly m $\bullet$-names $\{s_0, \dots, s_{m-1}\}$ and by the $\circ$-names $\{x_0, \dots, x_{|X|-1}\}$. The order of those names is dictated by ϕ_{in}. The inner interface u of H is actually formed taking in order the sorts of the names in $ic(H)$. The functions $u^\bullet$ and u° intuitively tell us where the i^{th} name of sort $\bullet$, or $\circ$, is positioned in $ic(H)$. Take for example s_i, the i^{th} $\bullet$-name, if $u^\bullet(i) = j$ then $ic(H)[j] = s_i$, but by the transformation $[\![\cdot]\!]$ we know that $ic(H)[j] = \overline{\phi}_{in}^{-1}(j) = s_i$ and this can happen by the definition of $\overline{\phi}_{in}^{-1}$ only if $\phi_{in}(i) = j$. Thus $u^\bullet = \phi_{in}|_m$. The same is true for $\circ$-names but we must remember to add m to the argument of ϕ_{in}. The final result is that:

$$\phi'_{in}(i) = \begin{cases} u^\bullet(i) = \phi_{in}(i) & \text{if } i < m \\ u^\circ(i-m) = \phi_{in}(i) & \text{otherwise} \end{cases}$$

thus $\phi'_{in} = \phi_{in}$. By a similar argument we have $\phi'_{out} = \phi_{out}$.

The next step is to check that the bigraphical interfaces of G and G' are effectively the same. As previously stated we have that $m' = m$, $l' = l$ and that the cardinality of the sets X and X' and those of Y and Y' coincide. The name choices σ_u and σ_v defined for H allow us to say that $X' = X$ and $Y' = Y$, indeed $X'[i] = \sigma_u(i) = X[i]$ and $Y'[j] = \sigma_v(j) = Y[j]$ for all i, j that are admissible for lists X and Y.

Now we can pass to examine the core of the bigraph G'. Recall that the set $\mathcal{N}_H = \mathcal{N}_H^\bullet \cup \mathcal{N}_H^\circ$ produced by the transformation $S[\![G]\!]$ is such that:

$$\mathcal{N}_H^\bullet = V_G \cup \{s_0, \dots, s_{m-1}\} \cup \{r_0, \dots, r_{n-1}\} \qquad \mathcal{N}_H^\circ = E_G \cup \{x_0, \dots, x_{|X|-1}\} \cup \{y_0, \dots, y_{|Y|-1}\}$$

Then:

1. The names of sort $\bullet$ that do not appear in inner or outer connection lists are exactly those in V_G. Hence $V_{G'} = V_G$

2. The gs-graph H has a proper assignment $n := \nu$ for each element n of E_G and no other one of this form. Thus $E_{G'} = E_G$

3. For each $n \in V_G$ we have a unique associated proper assignment $n := f(n_\bullet, n_0, \dots, n_{h-1})$; then $\mathsf{cl}_{G'}(n) = f$ but also $\mathsf{cl}_G(n) = f$ by definition. Hence $\mathsf{cl}_{G'} = \mathsf{cl}_G$.

4. As for the previous point let $n := f(n_\bullet, n_0, \ldots, n_{h-1})$ be the proper assignment that involves $n \in V_G$. We know by definition that $n_\bullet : \bullet$ and that $n_\bullet = \overline{\mathsf{pt}_G}(n)$. If $n_\bullet \in V_G$, then $\mathsf{pt}_{G'}(n) = n_\bullet$ but also by the definition of $\overline{\mathsf{pt}_G}$ we have $\mathsf{pt}_G(n) = n_\bullet$. Otherwise let $n_\bullet$ be an outer connection, say r_h. On one hand this implies that $\mathsf{pt}_G(n) = h \in m$, on the other hand there exists a j such that $oc(H)[j] = r_h$ and this corresponds to $\overline{\phi}_{out}^{-1}(j)$ if and only if $\phi_{out}^{-1}(j) = h$, since by definition $\mathsf{pt}_{G'} = \phi_{out}^{-1}(j) = h$, $\mathsf{pt}_G(n) = \mathsf{pt}_{G'}(n)$. The equality of the two parent maps on the sites is shown in a similar way. Hence, we have that $\mathsf{pt}_G = \mathsf{pt}_{G'}$.

5. Showing that $\mathsf{lk}_{G'} = \mathsf{lk}_G$ involves name choices, but is otherwise analogous to the previous point.

Since all the components of G' and G coincide, we conclude that $G' = B[\![S[\![G]\!]]\!] = G$.

[$\underline{S[\![B[\![H]\!]]\!] = H}$]. For the second part of the proof, consider a gs-graph with name choices $H : (u, \sigma_u) \to (v, \sigma_v)$. Let us take for simplicity an instance of the isomorphism class of H in which the names on the outer connection list are $\{r_0, \ldots, r_{l-1}\}$ for the $\bullet$ sort and $\{y_0, \ldots, y_h\}$ for the other sort and the names on the inner connection list are $\{s_0, \ldots, s_{m-1}\}$ and $\{x_0, \ldots, x_k\}$. Call $G = B[\![H]\!]$ with shuffle function ϕ_{in} and ϕ_{out}. These functions are defined as:

$$\phi_{in}(i) = \begin{cases} u^\bullet(i) & \text{if } 0 \leq i < m \\ u^\circ(i-m) & \text{otherwise} \end{cases} \qquad \phi_{out}(i) = \begin{cases} v^\bullet(i) & \text{if } 0 \leq i < l \\ v^\circ(i-l) & \text{otherwise} \end{cases}$$

Because u and v are the sorts of the lists $ic(H)$ and $oc(H)$ we can deduce that:

$$ic(H) = (\overline{\phi}_{in}^{-1}(0), \overline{\phi}_{in}^{-1}(1), \ldots, \overline{\phi}_{in}^{-1}(m+k-1)) \qquad oc(H) = (\overline{\phi}_{out}^{-1}(0), \overline{\phi}_{out}^{-1}(1), \ldots, \overline{\phi}_{out}^{-1}(l+h-1))$$

Now call $H' = S[\![G]\!]$; the first thing to note is that H' and H have exactly[6] the same ic and oc lists: indeed they are obtained using the shuffle functions. The next step consists in checking that the set of proper assignments and auxiliary assignments of the two gs-graphs coincide. The technical details of this part of the proof are in Appendix (see Section A.2). □

Corollary 3.18. The following two structures:

- shuffled lean support-equivalent bigraphs over a pure signature $\mathcal{K}$;
- amended gs-graphs over $\mathcal{K}$ (seen as a ν-signature) with name choices

are isomorphic as partial algebras equipped with sequential and parallel composition.

Proposition 3.19. (**$B[\![\cdot]\!]$ preserves sequential and parallel composition**)
Let $H : (u, \sigma_u) \to (v, \sigma_v)$ and $H' : (v, \sigma_v) \to (w, \sigma_w)$ be two gs-graphs with name choices, then $B[\![H; H']\!] = B[\![H']\!] \circ B[\![H]\!]$.

Consider instead $H_0 : (u_0, \sigma_{u_0}) \to (v_0, \sigma_{v_0})$ and $H_1 : (u_1, \sigma_{u_1}) \to (v_1, \sigma_{v_1})$ such that $Im(\sigma_{u_0}) < Im(\sigma_{u_1})$ and $Im(\sigma_{v_0}) < Im(\sigma_{v_1})$. We have: $B[\![H_0 \otimes H_1]\!] = B[\![H_0]\!] \otimes B[\![H_1]\!]$.

[6]Since we have chosen the names of H appropriately.

Proof:
Instead of giving a direct proof that works on the definition of $B[\![\cdot]\!]$, we use the fact that $B[\![\cdot]\!]$ is the inverse of $S[\![\cdot]\!]$ and that $S[\![\cdot]\!]$ preserves the operations as stated in Proposition 3.13. For H and H' gs-graphs with name choices call $G = B[\![H]\!]$ and $G' = B[\![H']\!]$ the two shuffled bigraphs associated. Since $B[\![\cdot]\!]$ and $S[\![\cdot]\!]$ are inverse to each other, we have $H = S[\![G]\!]$ and $H' = S[\![G']\!]$. Hence

$$B[\![H; H']\!] = B[\![S[\![G]\!]; S[\![G']\!]]\!] = B[\![S[\![G; G']\!]]\!] = G; G' = B[\![H]\!]; B[\![H']\!]$$

Preservation of parallel composition is proved analogously. □

Corollary 3.20. Let $H : (u, \sigma_u) \to (v, \sigma_v)$ and $H' : (v, \sigma_v) \to (w, \sigma_w)$ be two amended gs-graphs with name choices, then $B[\![H; ; H']\!] \Bumpeq B[\![H']\!] \circ B[\![H]\!]$.

As both the bijections $S[\![\cdot]\!]$ and $B[\![\cdot]\!]$ preserve composition and tensor, we can view bigraphs and gs-graphs not only as "essentially" equivalent formulations, but as "essentially" isomorphic algebras, thus exporting to bigraphs the gs-monoidal structure of gs-graphs, although equipped with partial operations.

4. Characterising Binding Bigraphs as Legal gs-graphs

In binding bigraphs, besides the possibility of having local names, there is added a scope discipline for limiting the visibility of such local names. In particular a control may declare some names that only its descendants can use, thus relaxing in part the assumption of independence of the two graphical structures.

While in the pure case the correspondence can be worked out smoothly, the case of binding signatures is more challenging. At the signature level, the idea is just to consider operators of the form $K : \bullet\circ^h \to \bullet\circ^k$ for h the binding arity of K and k the free arity of K. Then we can straightforwardly define an injective transformation $S_{bind}[\![\cdot]\!]$ from (shuffled) binding bigraphs to gs-graphs as a suitable extension of the one in Section 3.1 (see Appendix B). The main difference is that now the class of gs-graphs freely generated by the signature may contain some elements that do not correspond to any valid binding graphs, because the scope discipline is not enforced by the free construction.

4.1. Binding bigraphs

Definition 4.1. (binding signature)
A *binding signature* has a set of controls $\mathcal{K}$ and an arity function $\mathsf{ar} : \mathcal{K} \to \mathbb{N} \times \mathbb{N}$. If $\mathsf{ar}(K) = (h, k)$, we write $K : h \to k$ and we call, respectively, h and k the *binding arity* and the *free arity* of K and they index respectively the *binding ports* and the *free ports* of K.

Definition 4.2. (binding interface)
A *binding interface* is a triple $I = \langle m, loc, X\rangle$ where the ordinal m and the set of names X are as in pure bigraphs, and $loc \subseteq m \times X$ is the *locality* of the interface. If $(i, x) \in loc$ we say that i is a *place* of x. We denote by $I^u = \langle m, X\rangle$ the pure interface underlying I.

Example 4.3. The approach of the binding bigraphs for avoiding misleading compositions consists in enriching the interfaces with a locality relation loc that establishes to which places, if any, a name belongs to. Figure 4 denotes a simple binding bigraph with a single control with a local name and two sites in it; the locality relation on the inner interface associates the name to the first site. This restriction prevents controls in the second site from using this name.

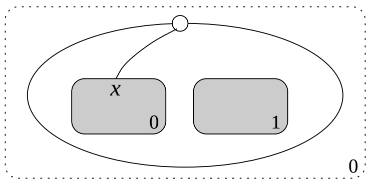


Figure 4. A binding bigraph

Definition 4.4. (locality relation)
Let $I = \langle m, loc_I, X\rangle$ and $J = \langle n, loc_J, Y\rangle$ be binding interfaces and consider a pure bigraph $G^u : I^u \to J^u$ on the pure underlying interfaces. Then the *locality relation* $loc_G \subseteq (m \uplus n \uplus V) \times (X \uplus Y \uplus P \uplus E)$, is the smallest relation such that:

- if $(i, x) \in loc_I$ then $(i, x) \in loc_G$ ($loc_I \subseteq loc_G$)
- if $(j, x) \in loc_J$ then $(j, x) \in loc_G$ ($loc_J \subseteq loc_G$)
- if p is a binding port of a node v then $(v, p) \in loc_G$
- if p is a free port of a node v then $(\mathsf{pt}(v), p) \in loc_G$
- if an edge e contains a binding port of v then $(v, e) \in loc_G$

Definition 4.5. (binding bigraph)
Given two binding interfaces I and J a *concrete binding bigraph* $G : I \to J$ consists of an underlying pure bigraph $G^u : I^u \to J^u$ such that: (a) any edge has at most one binding port, while an outer name has none; (b) if $\mathsf{lk}_G(q) = l$ is a local link then q is also local, and whenever $(w, q) \in loc_G$ then there exists w' such that $\mathsf{pt}^k_G(w) = w'$ for some $k \in \mathbb{N}$ and $(w', l) \in loc_G$. The condition (b) is called the scoping rule.

4.2. Legal gs-graphs

We start by showing that the ordinary transformation from binding bigraphs to gs-graphs is not surjective. To see this, we note that the set of gs-graphs that are images of bigraphs is closed under monoidal product, but not under sequential composition.

Example 4.6. Consider the gs-graphs in Fig. 5: the two gs-graphs on the left trivially respect the scope rule, but their sequential composition links h to the local port x of g despite h and g are siblings.

The main result we present here is the characterisation of "well-scoped" gs-graphs in terms of a relational type system. We start by rephrasing the location principle for the scoping rule in the context of gs-graphs.

Definition 4.7. (bound name, free name and location, exploited name)
We say that a name n is *bound* to location p if p and n appears together in the left hand side of some proper assignment in G (i.e., G contains an assignment of the form $p, ..., n, ... := ...$). Sometimes we

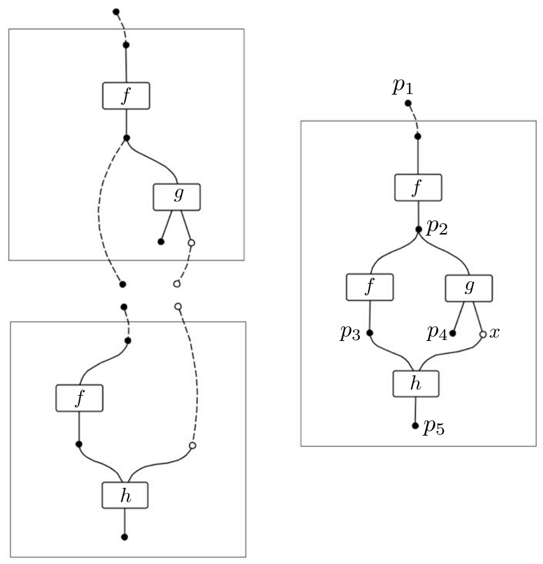


Figure 5. An example of composition that does not respect the scope rule

say just that n is bound, omitting the location p to which it is bound. If a name n is not bound then it is *free*.[7] If a location p is not assigned then we say it is *free*. We say that q *exploits* n if q and n appears together in the right hand side of some proper assignment in G (i.e., G contains an assignment of the form ... $:= f(q, ..., n, ...)$).

Note that free locations are part of the outer connections and therefore they are exposed in the interface of the gs-graph.

Definition 4.8. (legal gs-graph)
A gs-graph G is *legal* iff for any p, q, n such that n is bound to p and q exploits n then $q \sqsubset_G^* p$, where $\sqsubset_G^*$ is the reflexive and transitive closure of $\sqsubset_G$.

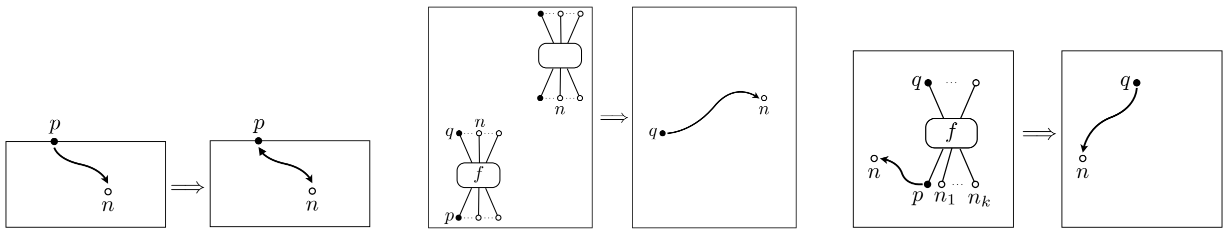
As a special case, any "pure" gs-graphs is legal because pure signatures forbid the presence of names bound to locations. Our second main correspondence result is:

Theorem 4.9. A gs-graph represents a binding bigraph iff it is legal.

Proof:
The proof goes by showing that the image of a binding bigraph via the transformation defined in Appendix B is a legal gs-graph (by contradiction, if it was not legal, then the scoping rule would have been violated) and then by giving a converse transformation from legal gs-graphs to binding bigraphs. □

Example 4.10. Let us consider a binding signature with three operators $f : \bullet \to \bullet$, $g : \bullet\circ \to \bullet\circ^2$ and $h : \bullet \to \bullet\circ$. The gs-graph in Fig. 5 can be defined as $G = \{\ p_2 := f(p_1)\ ,\ p_3 := f(p_2)\ ,\ p_4, x :=$

[7]For the purpose of this section, if n is assigned by $n := \nu$ then it is not bound and thus it is said free.



(a) rule (M) (b) rule (U1) (c) rule (U2)

Figure 6. Graphical illustration of the type system for legal gs-graphs

$g(p_2)$, $p_5 := h(p_3, x)$, $!(p_4)$, $!(p_5)$ }, which is not legal because p_3 exploits the node x (by the assignment $p_5 := h(p_3, x)$), which is bound to p_4 (by $p_4, x := g(p_2)$) and p_4 is not an ancestor of p_3 (i.e., $p_3 \not\angle^*_G p_4$).

We can conveniently characterise the class of legal gs-graphs by exploiting an elegant type system. The typing relations we are interested in are of the form "p uses n" and "p misuses n". We need just three inference rules:

$$(M)\frac{p \text{ free} \quad p \text{ uses } n}{p \text{ misuses } n} \qquad (U1)\frac{p, ... := f(q, ..., n, ...) \quad n \text{ bound}}{q \text{ uses } n}$$

$$(U2)\frac{p, n_1, ..., n_k := f(q, ...) \quad p \text{ uses } n \quad \forall i.\, n \neq n_i}{q \text{ uses } n}$$

The rules can be illustrated pictorially. Let us draw an arrow between p and n when "p uses n" and a double headed arrow when "p misuses n". Then the three rules can be informally sketched as in Fig. 6. Roughly the rules state that if q exploits n and n is bound to some other location, say p', then we must check that q be a descendant of p'. This task is accomplished by introducing dependencies of the form "p uses n" with rule (U1), then by propagating "upward" such dependencies through the location hierarchy via rule (U2) until either we discover that p' is an ancestor of q (in which case the propagation stops) or we reach the (free) root location p, in which case the scoping rule is violated and we assert that "p misuses n" thanks to rule (M).

In the above running example we derive the typing relation

$$\{\, p_3 \text{ uses } x \,,\, p_2 \text{ uses } x \,,\, p_1 \text{ uses } x \,,\, p_1 \text{ misuses } x \,\}$$

Proposition 4.11. A gs-graph is legal iff it induces an empty "misuses" relation.

Proof:
The proof is divided in two parts. First we show that if the "misuses" relation is not empty then the gs-graph is not legal. Conversely, we show that if a gs-graph is not legal, then the "misuses" relation is not empty. □

Theorem 4.12. The typing relation of $G_1 \otimes G_2$ is the union of the typing relations of G_1 and G_2. The typing relation of $G_1; G_2$ is a superset of the union of the typing relations of G_1 and of G_2.

Proof:
The proof of the first statement is trivial, because the assignments in $G_1 \otimes G_2$ are the union of the assignments in G_1 and G_2. The proof of the second statement follows by noting that all proper assignments of G_1 and G_2 are also present in $G_1; G_2$ and that the sequential composition may induce additional dependencies for the backward propagation of the "use" relation. □

Note that for computing the typing relation of $G_1; G_2$ it is enough to close the union of the typing relations of G_1 and G_2 w.r.t. the type inference rules (i.e., the reasoning is monotonic). The typing rules induce a straightforward quadratic algorithm for checking if a gs-graph is legal or not (the complexity is $O(B_G \cdot W_G)$ for B_G the number of $\bullet$ nodes in G and W_G the number of bound $\circ$ nodes in G).

Corollary 4.13. The parallel composition of two gs-graphs is legal iff both are legal. If a non legal gs-graph is used in a sequential composition the result is non legal.

5. Concluding Remarks

In conclusion, while bigraphs and gs-graphs are equally expressive, we claim that the presentation of gs-graphs in terms of sets of assignments combines the expressiveness of name links with the simpler and more standard algebraic structure of gs-monoidal theories. We believe that the relational type systems used above to check binding gs-graphs well-formedness may also be useful for establishing important properties of systems represented as gs-graphs.

A few observations are in place that deserves some future work. First, the research on bigraphs finds a main motivation in the reactive system approach mentioned in the introduction, which is based on the existence of so-called relative push-out (RPO) and idem push-out (IPO) in the category of bigraphs. RPOs/IPOs serve to distil the labelled transitions from the reduction rules and derive a bisimilarity equivalence that is guaranteed to be a congruence. Some preliminary work on extending the RPO approach to the case of term graphs is reported in [7]. We conjecture that the variant of the reactive approach based on so-called groupoidal RPOs [25] can be applied to the category of shuffled bigraphs and hence of gs-graphs. Moreover, we would like to exploit the gs-monoidal structure and 2-categorical rewriting techniques, along the lines of [5], to define a reference theory of concurrent rewrites for bigraphs and gs-graphs; this is currently missing.

Second, the fact that legal gs-graphs do not compose may suggest that their interfaces lack some additional information. In fact, while we can always represent binding bigraphs as legal gs-graphs, the interfaces of gs-graphs are still as defined in the encoding for pure bigraphs and thus are not able to pair names and the locations to which they are bound. One possible solution could be to fix some convention from which the needed binding information can be automatically inferred. For example, we might assume that the names ($\circ$) listed in the interface are bound to the rightmost location ($\bullet$) appearing on their right, if any (and free otherwise) and then check whether or not the gs-graph is legal. However, the information about name-sharing between locations would still get lost. We discarded this approach because, e.g., it would forbid the composition of legal gs-graphs with many arrows (i.e., symmetries like $\rho_{\circ,\bullet}$) that change the interface without affecting the structure of the graph, but may result in a non-legal gs-graph. We plan to investigate this issue in more detail, as we think it has still many advantages over other proposals, like [12], which resort to the introduction of a much more powerful closed monoidal structure.

Third, we would like to extend the comparison between binding bigraphs and legal gs-graphs to the algebra of graphs with nesting proposed in [1].

References

[1] R. Bruni, A. Corradini, F. Gadducci, A. Lluch-Lafuente, and U. Montanari. On gs-monoidal theories for graphs with nesting. In *Graph Transformations and Model-Driven Engineering*, volume 5765 of *LNCS*, pages 59–86. Springer, 2010.

[2] R. Bruni, F. Gadducci, and U. Montanari. Normal forms for algebras of connection. *Theor. Comput. Sci.*, 286(2):247–292, 2002.

[3] R. Bruni, U. Montanari, and F. Rossi. An interactive semantics of logic programming. *TPLP*, 1(6):647–690, 2001.

[4] L. Cardelli and A. D. Gordon. Mobile ambients. *Theor. Comput. Sci.*, 240(1):177–213, 2000.

[5] A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *CTCS'97*, volume 1290 of *LNCS*, pages 87–105. Springer, 1997.

[6] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 7(4):299–331, 1999.

[7] A. Corradini and F. Gadducci. On term graphs as an adhesive category. *Electr. Notes Theor. Comput. Sci.*, 127(5):43–56, 2005.

[8] V.-E. Căzănescu and G. Ştefănescu. A general result on abstract flowchart schemes with applications to the study of accessibility, reduction and minimization. *Theoretical Comput. Sci.*, 99(1):1–63, 1992.

[9] T. C. Damgaard and L. Birkedal. Axiomatizing binding bigraphs. *Nord. J. Comput.*, 13(1-2):58–77, 2006.

[10] G. L. Ferrari and U. Montanari. Tile formats for located and mobile systems. *Inf. Comput.*, 156(1-2):173–235, 2000.

[11] M. P. Fiore and M. D. Campos. The algebra of directed acyclic graphs. In B. Coecke, L. Ong, and P. Panangaden, editors, *Computation, Logic, Games, and Quantum Foundations*, volume 7860 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 2013.

[12] R. H. G. Garner, T. Hirschowitz, and A. Pardon. Variable binding, symmetric monoidal closed theories, and bigraphs. In *CONCUR'09*, volume 5710 of *LNCS*, pages 321–337. Springer, 2009.

[13] P. Hackney and M. Robertson. On the category of props, 2012. arXiv:1207.2773.

[14] O. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report UCAM-CL-TR-580, University of Cambridge, 2004.

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

[16] M. Johnson and D. Yau. On homotopy invariance for algebras over colored PROPs. *Journal of Homotopy and Related Structures*, 4:275–315, 2009.

[17] S. Lack. Composing PROPS. *Theory and Applications of Categories*, 13(9):147–163, 2004.

[18] S. MacLane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71(1):40–106, 1965.

[19] S. MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5. Springer, 2nd edition, 1998.

[20] J. Meseguer. Membership algebra as a logical framework for equational specification. In *WADT'97*, volume 1376 of *LNCS*, pages 18–61. Springer, 1997.

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

[22] R. Milner. Bigraphs whose names have multiple locality. Technical Report UCAM-CL-TR-603, University of Cambridge, 2004.

[23] R. Milner. Bigraphs and their algebra. *Electr. Notes Theor. Comput. Sci.*, 209:5–19, 2008.

[24] R. Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.

[25] V. Sassone and P. Sobocinski. Locating reaction with 2-categories. *Theor. Comput. Sci.*, 333(1-2):297–327, 2005.

[26] P. Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, pages 289–355. Springer, 2011.

A. Relating pure bigraphs and gs-graphs

In this section we collect some technical details about the correspondence result between pure bigraphs and gs-graphs.

Definition A.1. ($\overline{\phi}_{in}^{-1}$ and $\overline{\phi}_{out}^{-1}$)
Let $G = \langle V, E, \mathsf{cl}, \mathsf{pt}, \mathsf{lk}\rangle : \langle m, X, \phi_{in}\rangle \to \langle l, Y, \phi_{out}\rangle$ be a bigraph and let $\mathcal{N}_H^\bullet = V \uplus \{s_0, \ldots, s_{m-1}\} \uplus \{r_0, \ldots, r_{l-1}\}$ and $\mathcal{N}_H^\circ = E \uplus \{x_0, \ldots, x_{|X|-1}\} \uplus \{y_0, \ldots, y_{|Y|-1}\}$. We define $\overline{\phi}_{in}^{-1} : m + |X| \to \mathcal{N}_H$ and $\overline{\phi}_{out}^{-1} : l + |Y| \to \mathcal{N}_H$ as

$$\overline{\phi}_{in}^{-1}(j) \triangleq \begin{cases} s_i & \text{if } \phi_{in}^{-1}(j) = i < m \\ x_{i-m} & \text{if } \phi_{in}^{-1}(j) = i \geq m \end{cases} \qquad \overline{\phi}_{out}^{-1}(j) \triangleq \begin{cases} r_i & \text{if } \phi_{out}^{-1}(j) = i < l \\ y_{i-l} & \text{if } \phi_{out}^{-1}(j) = i \geq l \end{cases}$$

This way, we ensure that $ic(S[\![G]\!]) = (\overline{\phi}_{in}^{-1}(0), \overline{\phi}_{in}^{-1}(1), \ldots, \overline{\phi}_{in}^{-1}(m + |X| - 1))$ and $oc(S[\![G]\!]) = (\overline{\phi}_{out}^{-1}(0), \overline{\phi}_{out}^{-1}(1), \ldots, \overline{\phi}_{out}^{-1}(l + |Y| - 1))$.

A.1. Proof of Proposition 3.13: preservation of tensor product

We provide here some missing details about the the proof that $S[\![\cdot]\!]$ preserves the tensor product. The most difficult part is to show that $S[\![G_0 \otimes G_1]\!]$ and $S[\![G_0]\!] \otimes S[\![G_1]\!]$ have the same interfaces. Here we consider only the inner interfaces since the same reasoning can be applied to the outer interfaces. Moreover we assume for simplicity and without loss of generality, that the inner connections of $[\![G_1]\!]$ are $\{s_{m_o}, \ldots, s_{m_0+m_1-1}\}$ and $\{x_{|X|_0}, \ldots, x_{|X|_0+|X|_1-1}\}$. By definition, it follows that

$$ic(S[\![G_0 \otimes G_1]\!]) = [(\overline{\phi_0 \otimes \phi_1})^{-1}(0), \ldots, (\overline{\phi_0 \otimes \phi_1})^{-1}(m_0 + m_1 + |X_0| + |X_1| - 1)]$$

where

$$(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = \begin{cases} s_i & \text{if } (\phi_0 \otimes \phi_1)^{-1}(j) = i < m_0 + m_1 \\ x_(i - m_0 - m_1) & \text{if } (\phi_0 \otimes \phi_1)^{-1}(j) = i \geq m_0 + m_1 \end{cases}$$

The parallel product of gs-graphs instead tell us that the inner connections are formed juxtaposing the lists representing the inner connections of G_0 and G_1.

$$ic(S[\![G_0]\!] \otimes S[\![G_1]\!]) = [\overline{\phi_0}^{-1}(0), \ldots, \overline{\phi_0}^{-1}(m_0 + |X_0| - 1), \overline{\phi_1}^{-1}(0), \ldots, \overline{\phi_1}^{-1}(m_1 + |X_1| - 1)]$$

Thus it sufficient to show that $ic(S[\![G_0 \otimes G_1]\!]) = ic(S[\![G_0]\!] \otimes S[\![G_1]\!])$. We can split this problem in two subproblems:

- $ic(S[\![G_0 \otimes G_1]\!])[0 \ldots m_0 + |X_0| - 1] = ic(S[\![G_0]\!])$
- $ic(S[\![G_0 \otimes G_1]\!])[m_0 + |X_0| \ldots m_0 + m_1 + |X_0| + |X_1|] = ic(S[\![G_1]\!])$

Note that $(\phi_0 \otimes \phi_1) = (\phi_0 \oplus \phi_1) \circ (Id_{m_0} \oplus \rho_{m_1,|X_0|} \oplus Id_{|X_1|})$ and thus its inverse is $(\phi_0 \otimes \phi_1)^{-1} = (Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|}) \circ (\phi_0^{-1} \oplus \phi_1^{-1})$.

Now we approach the first subproblem. Consider $j \in [0, m_0 + |X_0| - 1]$ and $(\phi_0 \otimes \phi_1)^{-1}(j) = (Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_0(j))$. There are two possibilities:

- if $\phi_0(j) < m_0$, then $(Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_0(j)) = \phi_0(j) < m_0$ and $(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = s_{\phi_o(j)} = \overline{\phi_0}^{-1}(j)$.
- if $m_0 \leq \phi_0(j) < m_0 + |X_0| - 1$, then $(Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_0(j)) = \phi_0(j) + m_1 \geq m_0 + m_1$ and therefore $(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = x_{\phi_0(j)+m_1-m_0-m_1} = x_{\phi_0(j)-m_0} = \overline{\phi_0}^{-1}(j)$.

The second subproblem consists in showing that for each j such that $m_0 + |X_0| \leq j \leq m_0 + m_1 + |X_0| + |X_1| - 1$, we have $(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = \overline{\phi_1}^{-1}(j - m_0 - |X|_0)$. The left member of the equation can be rewritten as: $(\phi_0 \otimes \phi_1)^{-1}(j) = (Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_1^{-1}(i) + m_0 + |X|_0)$, where $i = j - m_0 - |X_0|$. Again, we have two possibilities:

- If $\phi_1^{-1}(i) < m_1$, then

  $$(Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_1^{-1}(i) + m_0 + |X|_0) = \phi_1^{-1}(i) + m_0 + |X_0| - |X_0| = \phi_1^{-1}(i) + m_0$$

  because $\phi_1^{-1}(i)$ is in the range of the permutation $\rho_{|X_0|,m_1}$ and it is shifted by $|X_0|$ backward. Then, we have $(\phi_0 \otimes \phi_1)^{-1}(j) < m_0 + m_1$ and thus $(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = s_{(\phi_0 \otimes \phi_1)^{-1}(j)} = s_{\phi_1^{-1}(i)+m_0} = \overline{\phi_1}^{-1}(i)$, where the last equality follows from the choice that we have made for the names on names of the inner connection of $S[\![G_1]\!]$.
- If $\phi_1^{-1}(i) \geq m_1$, then $(Id_{m_0} \oplus \rho_{|X_0|,m_1} \oplus Id_{|X_1|})(\phi_1^{-1}(i) + m_0 + |X|_0) = \phi_1^{-1}(i) + m_0 + |X_0|$ and $(\overline{\phi_0 \otimes \phi_1})^{-1}(j) = x_{(\phi_0 \otimes \phi_1)^{-1}(j)-m_0-m_1} = x_{(\phi_1^{-1}(i)+m_0+|X_0|-m_0-m_1)} = x_{(\phi_1^{-1}(i)+|X_0|-m_1)} = \overline{\phi_1}^{-1}(i)$. Indeed the names returned by $\overline{\phi_1}^{-1}$ follows those of $S[\![G_0]\!]$.

Having shown that the interfaces are the same, the equality of the two gs-graphs is easy. In fact we know that the product of two bigraphs consists in the union of the components that are supposed to be disjoint. Therefore taking for example $v \in V_{G_0}$ we have an assignment in $S[\![G_0]\!] \otimes S[\![G_1]\!]$ formed using the parent map and the link map of G_0. We find the same assignment in $S[\![G_0 \otimes G_1]\!]$ since $v \in V_{G_0} \uplus V_{G_1}$ and the mappings of $G_0 \otimes G_1$ behave as those of G_0 when applied on places and points of G_0. Clearly also the vice versa is true because given $v \in V_{G_0} \uplus V_{G_1}$ it must be an element of either V_{G_0} or V_{G_1}. Thus the assignment associated to v must be either in $S[\![G_0]\!]$ or in $S[\![G_1]\!]$ and consequently it belongs to $S[\![G_0]\!] \otimes S[\![G_1]\!]$. The same reasoning can be easily extended to edges, sites and inner names of the bigraphs, concluding the proof.

A.2. Proof of Theorem 3.17: $H = S[\![B[\![H]\!]]\!]$

We provide here the missing details about the second part of the proof of Theorem 3.17. Remind that we started with a gs-graph with name choices $H : (u, \sigma_u) \to (v, \sigma_v)$, that G denotes $B[\![H]\!]$ and that it is left to show that the set of assignments in H and $S[\![B[\![H]\!]]\!]$ do coincide. Let $n := f(n_\bullet, n_0, \dots, n_h)$ be a proper assignment in H with f clearly different from ν. The name n is thus put in the set V_G, and in H' we have therefore an assignment

$$n := \mathsf{cl}_G(n)(\overline{\mathsf{pt}_G}(n), \overline{\mathsf{lk}_G}(n, 0), \dots \overline{\mathsf{lk}_G}(n, h-1))$$

Next we prove that the two assignments are actually the same. The operators coincide, because transformation $B[\![G]\!]$ imposes that $\mathsf{cl}_G(n) = f$. For $\overline{\mathsf{pt}_G}$ and $\overline{\mathsf{lk}_G}$ we proceed by case analysis. We begin with the parent map.

If $n_\bullet$ is an element of V_G there are no problems: indeed we have that $\mathsf{pt}_G(n) = n_\bullet$ and consequently $\overline{\mathsf{pt}_G}(n) = n_\bullet$. If instead $n_\bullet$ is part of the outer interface, then $n_\bullet = oc(H)[j] = \overline{\phi}^{-1}_{out}(j) = r_{\phi^{-1}_{out}(j)}$ for some index j, but $\mathsf{pt}_G(n) = \phi^{-1}_{out}(j)$ and $\overline{\mathsf{pt}_G}(n) = r_{\phi^{-1}_{out}(j)}$; thus $n_\bullet = \overline{\mathsf{pt}_G}(n)$.

For the $\circ$-names in the assignment we must check that $n_i = \overline{\mathsf{lk}_G}((n, i))$. If $n_i \in E_G$ the result is immediate because $n_i = \mathsf{lk}_G((n, i)) = \overline{\mathsf{lk}_G}((n, i))$. If instead there exists j such that $n_i = oc(H)[j] = \overline{\phi} - out^{-1}(j) = y_{\phi^{-1}_{out}(j)-l}$, but also $\overline{\mathsf{lk}_G}((n, i)) = y_{\phi^{-1}_{out}(j)-l}$ since $\mathsf{lk}_G((n, i)) = Y[\phi^{-1}_{out}(j) - l]$ by definition. Thus for every $i \in \{0, \dots, h-1\}$ we have $n_i = \overline{\mathsf{lk}_G}((n, i))$ and we have therefore shown that each proper assignment in H (with $f \neq \nu$) is also in H' and vice versa.

Now consider the proper assignments in H involving the restriction operator ν. Let $n := \nu$ be one of such assignments, then in G we have that $n \in E_G$ and that each element of E_G is assigned the ν operator in H'. Hence we have $n := \nu$ in H'. Clearly the reasoning can be reversed to show that for each of such assignments in H' there is an equal one in H; indeed in H' the assignments with ν are in one-to-one correspondence with the elements of E_G.

Finally we have to take into account the auxiliary assignments: the arguments that leads to conclude that there are the same auxiliary assignments in H and H' are similar to those used in the case of proper assignments with the control different form ν.

Since all assignments coincide, we can conclude that $H = S[\![B[\![H]\!]]\!] = H'$.

B. Transforming binding bigraphs in gs-graphs

The construction of the gs-graph representing a certain binding bigraph is not so different from that relative to pure bigraphs. In fact, as previously mentioned, we can not represent the locality relations in the context of gs-graphs and the transformation is forced to ignore them. Therefore we have to deal only with the added possibility for controls to declare names. Likewise the pure case, we work with gs-graphs equipped with name choices on the interfaces and with shuffled binding bigraphs. The latter are defined exactly like shuffled pure bigraphs (see Definition 3.8), except that in place of a pure bigraph we have clearly a binding bigraph. Let $G = (V_G, E_G, \mathsf{cl}_G, \mathsf{pt}_G, \mathsf{lk}_G) : \langle m, loc_{in}, X, \phi_{in}\rangle \to \langle l, loc_{out}, Y, \phi_{out}\rangle$ be a shuffled binding bigraph on a signature $\mathcal{K}$ and denote with $\mathsf{Ps}^{loc}_G \subseteq \mathsf{Ps}_G$ the set of all its local ports. In particular given a node $v \in V_G$ such that its associated control $\mathsf{cl}_G(v)$ has a positive internal arity, we call $(v, local_i)$ the $(i+1)^{th}$ local port declared by v.

In the gs-graph $H = S_{bind}[\![G]\!]$ the following names will appear:

$$\mathcal{N}_H^\bullet = V_G \uplus \{s_0, \ldots, s_{m-1}\} \uplus \{r_0, \ldots, r_{l-1}\} \qquad \mathcal{N}_H^\circ = E_G \uplus \{x_0, \ldots, x_{|X|-1}\} \uplus \{y_0, \ldots, y_{|Y|-1}\}$$

Note that such sets of names are the same of those defined by the analogous transformation for the pure case. The difference is in the role played by the edges since in binding bigraphs they can be attached to local ports. In the corresponding gs-graph such local edges are not assigned with the restriction operator ν, but within the proper assignment of the node that declares the local port to which it is linked. Since every edge can be attached to at most one local port (see Definition 4.5) we are guaranteed that each local edge is assigned exactly once. In the following we denote with E_G^{local} the set of all local edges belonging to the bigraph G.

As in Section 3.1 the overlined maps $\overline{\mathsf{pt}} : m \uplus V \to \mathcal{N}_H^\bullet$ and $\overline{\mathsf{lk}} : \mathsf{Ps}_G \uplus X \to \mathcal{N}_H^\circ$ will help us in having a more concise representation of the assignments of H. (They are defined in the same way as in Section 3.1.)

Then we give the definitions of the assignments in H.

1. $\forall v \in V_G$ with $\mathsf{cl}_G(v) = f$ we add the assignment

$$v\ \overline{\mathsf{lk}}(v, local_0)\ \ldots \overline{\mathsf{lk}}(v, local_{k-1}) := f(\overline{\mathsf{pt}}(v), \overline{\mathsf{lk}}(v, 0), \ldots, \overline{\mathsf{lk}}(v, k-1))$$

   where k and h are respectively the binding and the free arity of f. Note that since in binding bigraph a local port can be linked to an outer name, the link map and its overlined version applied on a local port return always an edge (that clearly is local).

2. $\forall e \in E_G \backslash E_G^{local}$ we add $e := \nu$
3. $\forall i \in m$ we add $s_i := \overline{\mathsf{pt}}(i)$
4. $\forall x \in \{x_0, \ldots, x_{|X|-1}\}$ we add $x_i := \overline{\mathsf{lk}}(X[i])$

The inner and the outer connections are $\{s_0, \ldots, s_{m-1}\} \cup \{x_0, \ldots, x_{|X|-1}\}$ and $\{r_0, \ldots, r_{l-1}\} \cup \{y_0, \ldots, y_{|Y|-1}\}$ respectively and their order is obtained through the shuffle functions. In particular

$$\begin{aligned} ic(H) &= (\overline{\phi}_{in}^{-1}(0), \overline{\phi}_{in}^{-1}(1), \ldots, \overline{\phi}_{in}^{-1}(m + |X| - 1)) \\ oc(H) &= (\overline{\phi}_{out}^{-1}(0), \overline{\phi}_{out}^{-1}(1), \ldots, \overline{\phi}_{out}^{-1}(l + |Y| - 1)) \end{aligned}$$

where $\overline{\phi}_{in}^{-1}$ and $\overline{\phi}_{out}^{-1}$ are defined as in Definition A.1.

In conclusion the name choices for the interfaces of $S_{bind}[\![G]\!]$ are $\sigma_{in}(i) \triangleq X[i]$ for $i \in |X|$ and $\sigma_{out}(i) \triangleq Y[i]$ for $i \in |Y|$.

We can now prove that the gs-graphs produced by $S_{bind}[\![\cdot]\!]$ effectively represent binding bigraphs, or better, their bodies. We could not indeed encode the locality relation on the gs-graph interfaces, but we can show that the internal structure of a binding bigraph is faithfully represented.

For this purpose we abstract from the locality relation put on the interfaces and we identify all the binding bigraphs that differ only in these relations. We write $G \approx G'$ if G and G' are such two bigraphs and it follows immediately that $\approx$ is an equivalence relation. Furthermore, since $S_{bind}[\![\cdot]\!]$ does not take into account the locality relation, we have that $G \approx G'$ implies $S_{bind}[\![G]\!] = S_{bind}[\![G']\!]$ and therefore we can give a well-defined mapping on equivalence classes $S_{bind}\backslash_{\approx}[\![\cdot]\!]$ such that $S_{bind}\backslash_{\approx}[\![[G]]\!] = S_{bind}[\![G]\!]$, where $[G]$ denote the equivalence class of G.

Proposition B.1. The mapping $S_{bind}\backslash_{\approx}[\![\cdot]\!]$ is injective.

Unfortunately this mapping is not surjective. Next proposition guarantees that in the image of the mapping $S_{bind}[\![\cdot]\!]$ there are no unacceptable gs-graphs.

Proposition B.2. Let G be a binding bigraph. If in $S_{bind}[\![G]\!]$ there are an assignment $n \ldots := f(v, \ldots, e, \ldots)$ and an assignment $w \ldots e \ldots := g(\ldots)$ then $w \sqsubset^+ v$.

Proof:
First, we prove by induction that whenever we have $\mathsf{pt}^k_G(v) = w$ with $v, w \in V_G$ for some $k > 0$, then in the gs-graph $S_{bind}[\![G]\!]$ it results that $w \sqsubset^+ v$.

Then we make use of the injective property of the transformation for knowing what the two assignments mean in bigraphical terms. Having the assignment

$$w \ldots e \ldots := g(\ldots)$$

implies that $w \in V_G$ and that e is a local edge linked to a local port of w. Then we can say, using the notation introduced in Section 4 that $(w, e) \in loc_G$. The other assignment

$$n \ldots := f(v, \ldots, e, \ldots)$$

means that the one port, say the $(i+1)^{th}$ port, of $n \in V_G$ is linked to the local edge e. Then $(v, (n, i)) \in loc_G$. The scope rule imposes that $v \leq_G w$, thus v can not be an outer connection, but it must be a node of the bigraph. In terms of the parent map we have that $w = \mathsf{pt}^k(v)$ for some $k > 0$. Since $v, w \in V_G$ we can deduce that $w \sqsubset^+ v$. □

Finally we show that $S_{bind}[\![\cdot]\!]$ preserves the operations.

Proposition B.3. ($S_{bind}[\![\cdot]\!]$ preserves operations)
Let $G : \langle m, loc_I, X, \phi_{in}\rangle \to \langle n, loc_J, Y, \psi\rangle$ and $G' : \langle n, loc_J, Y, \psi\rangle \to \langle l, loc_H, Z, \phi_{out}\rangle$, be shuffled binding bigraphs. Then $S_{bind}[\![G' \circ G]\!] = S_{bind}[\![G]\!]; S_{bind}[\![G']\!]$

Suppose that $(G_0 : \langle m_0, loc_{I_0}, X_0, \phi_0\rangle \to \langle n_0, loc_{J_0}, Y_0, \psi_0\rangle)$ and $(G_1 : \langle m_1, loc_{I_1}, X_1, \phi_1\rangle \to \langle n_1, loc_{J_1}, Y_1, \psi_1\rangle)$ are shuffled binding bigraphs with $X_0 < X_1$ and $Y_0 < Y_1$. Then $S_{bind}[\![G_0 \otimes G_1]\!] = S_{bind}[\![G_0]\!] \otimes S_{bind}[\![G_1]\!]$