



**HAL**  
open science

## Inverse monoids of higher-dimensional strings

David Janin

► **To cite this version:**

David Janin. Inverse monoids of higher-dimensional strings. [Research Report] LaBRI, Université de Bordeaux. 2015. hal-01165724v1

**HAL Id: hal-01165724**

**<https://hal.science/hal-01165724v1>**

Submitted on 19 Jun 2015 (v1), last revised 5 Aug 2015 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Inverse monoids of higher-dimensional strings

David Janin

Université de Bordeaux,  
Bordeaux INP,  
LaBRI CNRS UMR 5800,  
INRIA Bordeaux Sud-Ouest  
F-33405 Talence, FRANCE  
`janin@labri.fr`

**Abstract.** Halfway between graph transformation theory and inverse semigroup theory, we define higher dimensional strings as bi-deterministic graphs with distinguished sets of input roots and output roots. We show that these generalized strings can be equipped with an associative product so that the resulting algebraic structure is an inverse semigroup. Its natural order is shown to capture existence of root preserving graph morphism. A simple set of generators is characterized. As a subsemigroup example, we show how all finite grids are finitely generated. Last, simple additional restrictions on products lead to the definition of subclasses with decidable Monadic Second Order (MSO) language theory.

## 1 Introduction

A never-ending challenge faced by computer science is to provide modeling concepts and tools that, on the one hand, allow for representing data and computations in a more and more abstract and richly structured way, but, on the other hand, remain simple enough to be taught to and used by application designers and software engineers [32].

A possible approach to this goal consists in generalizing to graphs the techniques that have already been developed for strings or trees such as the notion of recognizable languages and the associated notion of recognizers. In these directions, an enormous amount of techniques and works has been developed ranging from Lewis' graph compositions techniques [25] and Courcelle's developments of recognizability to graph languages [8] (see also [9]) up to more recent advances based on category theoretical development (see [14, 6] to name but a few).

Despite numerous successes in theoretical computer science, there is still room for polishing these techniques towards application to computer engineerings. The ideal balance to be reached between usage simplicity and mathematical coherence is a long-term goal [32]. While the underlying frameworks (the back end) of such tools can (and probably should) be based on robust mathematical tools, the interface (the front end) of these techniques and tools must be kept simple enough to remain teachable and usable.

Keeping in mind that strings, free monoids and related automata techniques are among the simplest and the most robust available models and are already

and successfully put in practice is system modeling methods like *event B* [2] we develop in this paper a notion generalized strings, called *higher dimensional strings*, in such a way that:

1. higher dimensional strings are simple: they are finitely generated from elementary graphs composed via a single and associative product that generalizes string concatenation in free monoids (Theorem 32),
2. the resulting classes of generalized strings include large classes of finite graphs such as, in particular, hypercubes, hence the name higher dimensional (Section 5 for the case of grids),
3. the resulting semigroups are inverse semigroups (Theorems 24 and 26) henceforth mathematically rich enough to provide algebraic characterization of graph-based concepts such as, for instance, graph morphisms by natural orders (Theorem 30) or acyclicity by quotient with ideal (Lemma 41),
4. some well-defined and rich subclasses of these generalized strings still enjoy an efficient, expressive and decidable language theory (Theorem 40).

Technically, our proposal amounts to combining concepts and results arising from the theory of inverse semigroups [24] and, beyond, group theory [27] and graph transformation approaches [25, 14, 6, 9].

Of course, various research developments have already shown that inverse semigroup theory is applicable to computer science, be it for data, computation, language or system modeling.

For data modeling, experiments in theoretical physics have already shown that structured data as complex as quasi-crystals can be described by means of some notion of (inverse) tiling semigroup [20–22]. Inverse semigroup theory has also been used to study reversible computations [10, 1]. More recently, various modeling experiments have been conducted in computational music [3, 18]. These last experiments also led to the definition of a Domain Specific (Programming) Language (DSL) which semantics is based on concepts arising from inverse semigroup theory [19, 15].

More closely related with the present paper, various developments of formal language theory linked with inverse semigroups have already been conducted [26, 29]. These (inverse) formal language theoretical approaches could also be conducted over higher dimensional strings when seen as (inverse) automata with multiple initial and terminal states. Such a point of view is however left for further studies.

Among recent works, let us also mention the study of the word problem on virtually free inverse semigroups [12] that already relates (elements of) inverse semigroups with graphs of bounded tree width. Let us also mention the lifting and the study of partially commutative trace theory (developed for concurrent systems modeling) into inverse semigroups [11].

## 2 Preliminaries

Let  $A = \{a, b, c, \dots\}$  be a finite alphabet of graph edge labels. Every concept defined in the sequel could be extended to hypergraphs, that is, graphs with edges that possibly relate more than two vertices. However, restricting our presentation to standard (relational) graph structures allows us to keep statements (and proofs) simpler.

**Relational graphs.** A (relational) graph on the (binary symbols) alphabet  $A$ , simply called  $A$ -*graph* or even *graph* when  $A$  is clear from the context, is a pair

$$G = \langle V, \{E_a\}_{a \in A} \rangle$$

with set of vertices  $V$  and  $a$ -labeled edge relation  $E_a \subseteq V \times V$  for every  $a \in A$ .

**Back and forth path labels.** Let  $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$  be a disjoint copy of the alphabet  $A$ . A back and forth path label (or simply path label) is a word from the free monoid  $(A + \bar{A})^*$  on the alphabet  $A + \bar{A}$ , with empty word denoted by 1 and the product of two words  $u$  and  $v \in (A + \bar{A})^*$  denoted by  $u \cdot v$  or simply  $uv$ . Then, the reverse mapping  $w \mapsto \bar{w}$  from  $(A + \bar{A})^*$  into itself is inductively defined by

$$\bar{1} = 1, \overline{a \cdot v} = \bar{v} \cdot \bar{a} \text{ and } \overline{\bar{a} \cdot v} = \bar{v} \cdot a$$

for every  $a \in A$ ,  $x \in A + \bar{A}$  and  $v \in (A + \bar{A})^*$ . It is an easy observation that the reverse mapping is an involutive monoid anti-isomorphism, that is, we have  $\overline{u \cdot v} = \bar{v} \cdot \bar{u}$  and  $\overline{\bar{w}} = w$  for every  $u, v, w \in (A + \bar{A})^*$ .

**Back and forth path actions.** For every  $X \subseteq V$  and  $w \in (A + \bar{A})^*$ , the set  $X \cdot w \subseteq V$  of vertices reachable from  $X$  following  $w$  is inductively defined by

$$\begin{aligned} X \cdot 1 &= X, \quad X \cdot av = \{y \in V : \exists x \in X, (x, y) \in E_a\} \cdot v \\ &\text{and } X \cdot \bar{a}v = \{y \in V : \exists x \in X, (y, x) \in E_a\} \cdot v \end{aligned}$$

for every letter  $a \in A$  and every string  $v \in (A + \bar{A})^*$ . In other words,  $X \cdot w$  is the set of vertices that can be reached from a vertex in  $X$  along a path labeled by  $w$ , where  $a$  (resp.  $\bar{a}$ ) denotes the forward (resp. backward) traversal of an  $a$ -labeled edge in the graph  $G$ .

One can check that  $X \cdot 1 = X$  and  $X \cdot (u \cdot v) = (X \cdot u) \cdot v$  for every  $X \subseteq V$  and every string  $u, v \in (A + \bar{A})^*$ . Rephrased in semigroup theoretical term, the edge relations of the graph  $G$  induce an *action* of the monoid  $(A + \bar{A})^*$  on the (powerset of the sets of) vertices of the graph  $G$ . It follows that the parenthesis can be removed without ambiguity.

**Notation for the singleton case.** When  $X$  is a singleton  $\{x\}$ , we may simply write  $x \cdot w$  instead of  $\{x\} \cdot w$ . Similarly, when  $x \cdot w$  itself is a singleton we may also treat it just as the element it contains. In other words, we may simply write

$$x \cdot w = y$$

instead of  $\{x\} \cdot w = \{y\}$ , to denote both the fact that there exists a (back and forth) path from vertex  $x$  to vertex  $y$  labeled by  $w$  and the fact that this path is unique. Similarly, we may say that  $x \cdot w$  is undefined (as a vertex) in the case  $x \cdot w = \emptyset$  (as a set).

**Graph morphism.** The usual notion of graph morphism can then be (re)defined via path actions as follows. Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  and  $G' = \langle V', \{E'_a\}_{a \in A} \rangle$  be two graphs on the alphabet  $A$ . A morphism  $f$  from  $G$  to  $G'$ , denoted by  $f : G \rightarrow G'$ , is a mapping  $f : V \rightarrow V'$  such that we have  $f(x \cdot a) \subseteq f(x) \cdot a$  and  $f(x \cdot \bar{a}) \subseteq f(x) \cdot \bar{a}$  for every  $x \in V$  and every  $a \in A$ . Then, by induction, we can easily prove that  $f(x \cdot w) \subseteq f(x) \cdot w$  for every  $x \in V$  and every  $w \in (A + \bar{A})^*$ .

**Graph quotient.** Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  be a graph. Let  $\simeq$  be an equivalence relation over the set  $V$ , that is, a reflexive and transitive relation. Let  $V/\simeq$  be the set of equivalence classes  $\{[x]_{\simeq} \subseteq V : x \in V\}$  where  $[x]_{\simeq} = \{x' \in V : x \simeq x'\}$ . Then, the quotient of the graph  $G$  by the equivalence  $\simeq$  is defined to be the graph  $G/\simeq = \langle V', \{E'_a\}_{a \in A} \rangle$  with set of vertices  $V' = V/\simeq_G$  and set of edges  $E'_a = \{([x], [y]) \in V' \times V' : ([x] \times [y]) \cap E_a \neq \emptyset\}$ . The mapping  $\eta_{\simeq} : V \rightarrow V/\simeq$  defined by  $\eta_{\simeq}(x) = [x]_{\simeq}$  for every  $x \in V$  is a surjective morphism called the canonical morphism from the graph  $G$  onto the quotient graph  $G/\simeq$ .

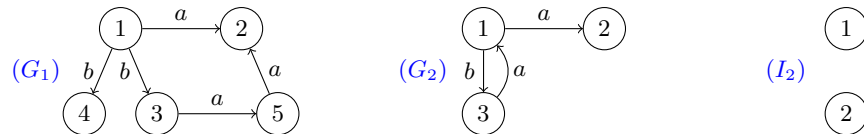
### 3 Unambiguous graphs and connecting morphisms

We define and study in this section the category of unambiguous graphs and connecting morphisms. Though fairly simple, this study is quite detailed for it constitutes the foundation of the notion of birooted graphs define in the next section.

**Definition 1 (Unambiguous graphs).** A graph  $G = \langle V, \{E_a\}_{a \in A} \rangle$  is *unambiguous* when, for every vertex  $x \in V$ , for every path  $w \in A + \bar{A}$ , there is at most on vertex  $y$  such that  $x \cdot w = \{y\}$ .

Clearly, by simple inductive argument,  $G$  is unambiguous as soon as the above condition is satisfied for every one letter path.

**Examples.** Graphs examples are depicted in Figure 1 with ambiguous graph  $G_1$  and unambiguous graphs  $I_2$  and  $G_2$ . In this figure, vertices are named only for illustrative purposes. These vertex names should not be understood as labels. Only edges are labeled in relational graphs.



**Fig. 1.** Ambiguous graph  $G_1$  and unambiguous graph  $G_2$ .

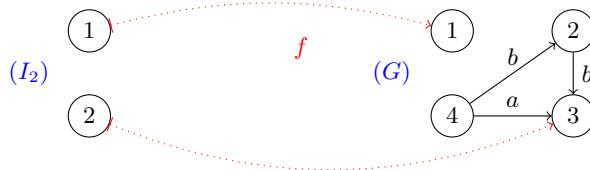
One can observe that graph  $G_1$  is ambiguous for two reasons. First, the upper left vertex 1 is the source of two edges labeled by  $b$ . Second, the upper right vertex 2 is the target of two edges labeled by  $a$ .

**Remark.** Observe that when a graph  $G$  is seen as a graph automaton on the alphabet  $A$ , it is unambiguous when it is both deterministic and co-deterministic. In the connected case, these unambiguous graphs are the Schützenberger graphs studied and used in [30].

**Definition 2 (Connecting morphisms).** Let  $f : G \rightarrow G'$  be a graph morphism between two graphs  $G = \langle V, \{E_a\}_{a \in A} \rangle$  and let  $G' = \langle V', \{E'_a\}_{a \in A} \rangle$ . The morphism  $f$  is a *connecting morphism* when for every  $x' \in V'$  there exist  $x \in V$  and  $w \in (A + \bar{A})^*$  such that  $x' \in f(x) \cdot w$ .

In other words, a morphism  $f : G \rightarrow G'$  is a connecting morphism when every vertex of graph  $G'$  is connected to the image of a vertex of  $G$  in graph  $G'$ .

**Examples.** Clearly, every surjective (i.e. onto) morphism is a connecting morphism. Another example of (non surjective) connecting morphism  $f : I_2 \rightarrow G$  is depicted in Figure 2.



**Fig. 2.** A connecting morphism  $\varphi : I_2 \rightarrow G$  with  $\varphi(1) = 1$  and  $\varphi(2) = 3$ .

**Remark.** Observe that when both  $G$  and  $G'$  are unambiguous, then, for every  $x \in V$ , every  $w \in (A + \bar{A})^*$ , if  $x \cdot w$  is not empty then so is  $f(x) \cdot w$  and we have  $f(x \cdot w) = f(x) \cdot w$ . This leads us to the following Lemma.

**Lemma 3 (Unique morphism completion).** Let  $G, G_1$  and  $G_2$  be three graphs. Let  $f_1 : G \rightarrow G_1$  and  $f_2 : G \rightarrow G_2$  be two graph morphisms. Assume that  $f_1$  is connecting and that both  $G_1$  and  $G_2$  are unambiguous. Then there exists at most one morphism  $g : G_1 \rightarrow G_2$  such that  $g \circ f_1 = f_2$ . Moreover, if  $f_2$  is connecting, then so is  $g$ .

*Proof.* Let  $g : G_1 \rightarrow G_2$  be a morphism such that  $g \circ f_1 = f_2$ . Let  $x_1$  be a vertex in  $G_1$ . Since  $f_1$  is connecting, there exist a vertex  $x$  of  $G$  and a path  $w \in (A + \bar{A})^*$  such that  $x_1 \in f_1(x) \cdot w$ . Since  $G_1$  is unambiguous, we have  $x_1 = f_1(x) \cdot w$ . Since  $g$  is a morphism, this implies that  $g(x_1) \in g(f_1(x) \cdot w)$ . Since  $G_2$  is unambiguous, this implies  $g(x_1) = g(f_1(x)) \cdot w$  hence, by applying the equality  $g \circ f_1 = f_2$ , we have  $g(x_1) = f_2(x) \cdot w$ . In other words,  $g(x_1)$  is uniquely determined by  $f_2(x)$  and the structure of  $G_2$ .

Last, assume that  $f_2$  is a connecting morphism. Then every vertex of  $G_2$  can be reached from a vertex is the image of  $f_2$  henceforth, thanks to the equality  $g \circ f_1 = f_2$ , it can also be reached from a vertex in the image of  $g$ . In other words,  $g$  is also a connecting morphism.  $\square$

Clearly, the composition of two connecting morphisms is a connecting morphism. Since the identity mapping over a graph is also a connecting morphism, this allows us to define the following categories.

**Definition 4 (Induced categories).** Let  $\mathbf{CGrph}(A)$  (resp.  $\mathbf{UCGrph}(A)$ ) be the category defined by finite graphs (resp. by finite unambiguous graphs) as objects and connecting morphisms as arrows.

We aim now at studying the properties of both category  $\mathbf{CGrph}(A)$  and category  $\mathbf{UCGrph}(A)$  and, especially, the way they are related. The notion of unambiguous congruence defined below allows to transform any graphs into its greatest unambiguous image. In group theory, this generalizes the notion of Stallings foldings [27].

**Definition 5 (Unambiguous congruence).** Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  be a graph on the alphabet  $A$ . A relation  $\simeq \subseteq V \times V$  over the vertices of  $G$  is an *unambiguous congruence* when it is an equivalence relation such that, for every  $a \in A$ , for every  $x, y \in V$ , if  $x \simeq y$  then we have both  $x \cdot a \times y \cdot a \subseteq \simeq$  and  $x \cdot \bar{a} \times y \cdot \bar{a} \subseteq \simeq$ .

The next lemma states in which sense unambiguous congruences can indeed be understood as congruences induced by morphisms via quotients.

**Lemma 6 (Soundness).** *Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  be a graph. Let  $\simeq$  be a unambiguous congruence. Then, the quotient graph  $G / \simeq$  is an unambiguous graph and the canonical graph morphism  $\eta_{\simeq} : G \rightarrow G / \simeq$  is a surjective henceforth connecting morphism.*

*Conversely, let  $f : G \rightarrow H$  be a morphism with unambiguous graph  $H$ . Then, the canonical equivalence  $\simeq_f$  induced by  $f$  over the vertices of  $G$  is an unambiguous congruence, we have  $G / \simeq_f \subseteq H$  with usual vertex set and edge set inclusions, and  $f$  restricted to  $G / \simeq_f$  is a surjective henceforth connecting morphism.*

*Proof.* Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  be a graph, and let  $\simeq$  be an unambiguous congruence over  $G$ . Let  $G / \simeq = \langle V', \{E'_a\}_{a \in A} \rangle$  be the quotient of  $G$  by  $\simeq$ . We have first to check that  $G / \simeq$  is unambiguous.

Assume that there exists  $x, y, z \in V$  such that both  $([x], [y]) \in E'_a$  and  $([x], [z]) \in E'_a$ . By definition, this means that there exist  $(x', y') \in E_a$  with  $x \simeq x'$  and  $y \simeq y'$ , and there exists  $(x'', z') \in E_a$  with  $x \simeq x''$  and  $z \simeq z'$ . But, since relation  $\simeq$  is an equivalence, this implies that  $x' \simeq x''$ . Then, by definition of unambiguous congruence, this implies that  $y' \simeq z'$  henceforth  $[y] = [z]$ . A symmetrical argument proves the symmetric case when both  $([y], [x]) \in E'_a$  and  $([z], [x]) \in E'_a$ .

Let then  $\eta_{\simeq} : V \rightarrow V/\simeq$  be the canonical onto mapping defined by  $\eta_{\simeq}(x) = [x]_{\simeq}$  for every  $x \in V$ . The fact  $\eta_{\simeq}$  is a surjective morphism immediately follows from the definition.

Conversely, let  $f : G \rightarrow H$  be a morphism with unambiguous graph  $H$ . Let  $\simeq_f$  be the canonical equivalence induced by  $f$  defined by  $x \simeq_f y$  when  $f(x) = f(y)$  for every  $x, y \in V$ .

Let us prove that  $\simeq_f$  is an unambiguous congruence. Let  $x, y \in V$  such that  $x \simeq_f y$ . Let  $a \in A + \bar{A}$ . In the case  $x \cdot a$  or  $y \cdot a$  is the empty set, nothing has to be proved. Assume that both  $x \cdot a$  and  $y \cdot a$  are non empty. Since  $f$  is a morphism we have  $\emptyset \subset f(x \cdot a) \subseteq f(x) \cdot a$  and  $\emptyset \subset f(y \cdot a) \subseteq f(y) \cdot a$ . Since  $H$  is unambiguous this implies that  $f(x \cdot a) = f(x) \cdot a$  and  $f(y \cdot a) = f(y) \cdot a$  and thus, because  $f(x) = f(y)$ , we have  $f(x \cdot a) = f(y \cdot a)$  hence  $(x \cdot a) \times (y \cdot a) \subseteq \simeq_f$ .

A symmetrical argument proves the symmetric case with  $x \cdot \bar{a}$  and  $y \cdot \bar{a}$ .  $\square$

The existence of a least congruence is stated in Lemma 7 and the associated universality property is stated Lemma 8.

**Lemma 7 (Least unambiguous congruence).** *Let  $G$  be a graph, possibly ambiguous. Then there exists a least unambiguous congruence  $\simeq_G$  over  $G$ . Moreover, in the case  $G$  is unambiguous, then  $\simeq_G$  is the identity relation.*

*Proof.* The complete relation over the vertices of  $G$  is a unambiguous congruence. Since unambiguous congruences are closed under intersection, the relation  $\simeq_G$  can just be defined as the intersection of all unambiguous congruences over  $G$ . In the case  $G$  is unambiguous, we easily check that the equality relation  $=$  is an unambiguous congruence.  $\square$

The graph  $G/\simeq_G$  is called the *greatest unambiguous graph image* of the graph  $G$ . Its maximality is to be understood in the following sense.

**Lemma 8 (Maximal unambiguous image).** *Let  $G$  be a graph. Let  $\simeq_G$  be its least unambiguous congruence. Then, for every graph morphism  $f : G \rightarrow H$  with unambiguous graph  $H$ , there exists a unique morphism  $g : G/\simeq_G \rightarrow H$  such that  $f = g \circ \eta_{\simeq_G}$ . Moreover, if  $f$  is connecting then so it  $g$ .*

*Proof.* Let  $G = \langle V, \{E_a\}_{a \in A}$  be a graph, let  $H = \langle V', \{E'_a\}_{a \in A}$  be an unambiguous graph, let  $f : G \rightarrow H$  be a morphism and let  $\simeq_f$  be the induced equivalence.

By Lemma 6, the equivalence  $\simeq_f$  is an unambiguous congruence. Since  $\simeq_G$  is the least unambiguous congruence, we have  $\simeq_G \subseteq \simeq_f$ . Let then  $\eta : V/\simeq_G \rightarrow V/\simeq_f$  be the inclusion mapping defined by  $\eta([x]_{\simeq_G}) = [x]_{\simeq_f}$ . We easily check that  $\eta$  is a surjective henceforth connecting morphism from  $G/\simeq_G$  onto  $G/\simeq_f$  with  $\eta_{\simeq_f} = \eta \circ \eta_{\simeq_G}$ .

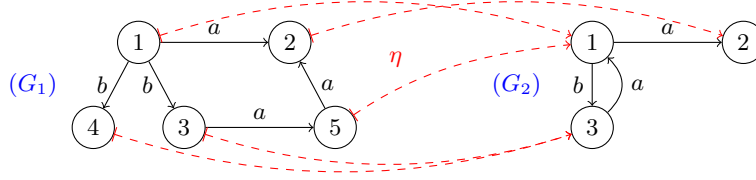
Now, we observe that  $G/\simeq_f \subseteq H$  with usual vertex set and edge sets inclusion. It follows that the inclusion mapping  $\theta : V/\simeq_f \rightarrow H$  is a graph morphism from  $G/\simeq_f$  to  $H$  such that  $f = \theta \circ \eta_{\simeq_f}$ .

It follows that  $g = \theta \circ \eta$  is a graph morphism from  $G/\simeq_G$  into  $H$  with  $f = g \circ \eta_{\simeq_G}$ . Since  $\eta_{\simeq_G}$  is connecting and both  $G/\simeq_G$  and  $H$  unambiguous, the unicity of  $g$  follows from Lemma 3.

Last, clearly, when  $f$  is a connecting morphism, then so is  $g$ .  $\square$



**Example.** An example of maximal graph image is provided by the graphs already depicted in Figure 1 where  $G_2$  has not been chosen at random since  $G_2 = G_1 / \simeq_{G_1}$ .



**Fig. 3.** Graph  $G_2$  is the maximal unambiguous image of graph  $G_2$ .

The canonical onto morphism  $\eta : G_1 \rightarrow G_1 / \simeq_{G_1} = G_2$  is depicted in Figure 3, encoding the least unambiguous congruence on  $G_1$  that glue 1 with 5, and 3 with 4.

**Remark.** The construction described above is a generalization of what is known in algebra as Stallings folding [27]. Observe that with  $G = \langle V, \{E_a\}_{a \in V} \rangle$ , the least unambiguous congruence  $\simeq_G$  equals the least fixpoint of the mapping  $F : V \times V \rightarrow V \times V$  defined by

$$F(R) = R \cup \bigcup \{(x \cdot a) \times (y \cdot a) \cup (x \cdot \bar{a}) \times (y \cdot \bar{a}) : (x, y) \in R, a \in A\}$$

that contains the equality. It follows, by applying classical fixpoint techniques, that  $\simeq_G = \bigcup_{n \geq 0} F^n(=)$ , henceforth it can be computed in quasi linear time. In other words, computing the maximal unambiguous image  $G / \simeq_G$  of the graph  $G$  can be done in time quasi linear in the size of the graph  $G$ .

**Remark.** Generalizing [30], provided some adequate sets of initial and terminal states are chosen, the graph  $G$  can be viewed as a non-deterministic automaton on the alphabet  $A + \bar{A}$ . Then computing the unambiguous graph  $G / \simeq_G$  associated to  $G$  just amounts to perform a minimization. However, such a formal language based approach, related with the study of inverse languages [26, 29, 27], can be detailed much more and goes out of the scope of the present paper.

Clearly, the category  $\mathbf{UCGrph}(A)$  is a subcategory of  $\mathbf{CGrph}(A)$ . The next lemma shows that maximal graph images extend to morphisms henceforth defining a projection functor from  $\mathbf{CGrph}(A)$  into  $\mathbf{UCGrph}(A)$ .

**Lemma 9 (Projected morphisms).** *Let  $G$  and  $H$  be two graphs. Let  $f : G \rightarrow H$  be a connecting morphism. Let  $\eta_G : G \rightarrow G / \simeq_G$  and  $\eta_H : H \rightarrow H / \simeq_H$  be the related canonical onto morphism. Then there exists a unique connecting morphism  $\varphi(f) : G / \simeq_G \rightarrow H / \simeq_H$  such that  $\varphi(f) \circ \eta_G = \eta_H \circ f$ .*

*Proof.* Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$  and  $H = \langle V', \{E'_a\}_{a \in A} \rangle$ . We aim at defining  $g = \varphi(f)$  as depicted in Figure 4.

$$\begin{array}{ccc}
G & \xrightarrow{f} & H \\
\eta_G \downarrow & & \downarrow \eta_H \\
G/\simeq_G & \xrightarrow{\varphi(f)} & H/\simeq_H
\end{array}$$

**Fig. 4.** Morphism induced by quotient.

Let  $\simeq$  be the canonical equivalence over  $G$  induced by the graph morphisms  $\eta_H \circ f$ . Since  $H/\simeq_H$  is unambiguous then, by Lemma 6, the relation  $\simeq$  is an unambiguous congruence over  $G$ , henceforth, by minimality,  $\simeq_G \subseteq \simeq$ . It follows that for every  $x, x' \in V$ , if  $x \simeq x'$  then  $\eta_H \circ f(x) = \eta_H \circ f(x')$ . Thus the mapping  $g : V/\simeq_G \rightarrow V/\simeq_H$  defined by  $g([x]_{\simeq_G}) = \eta_H \circ f(x)$  is well defined.

Let us prove it is a morphism. Let  $x \in V$  and  $a \in A$ . Since  $\eta_H \circ f$  is a morphism, we have  $\eta_H \circ f(x \cdot a) \subseteq \eta_H \circ f(x) \cdot a$ . Now, since  $G$  is unambiguous, we have  $[x]_{\simeq_G} \cdot a = [y]$  for any  $y \in x \cdot a$ . It follows that  $g([x]_{\simeq_G} \cdot a) = \eta_H \circ f(y)$  with  $y \in x \cdot a$  henceforth  $g([x]_{\simeq_G} \cdot a) \subseteq \eta_H \circ f(x \cdot a) \subseteq \eta_H \circ f(x) \cdot a$ . With  $g([x]_{\simeq_G}) = \eta_H \circ f(x)$  we thus have  $g([x]_{\simeq_G} \cdot a) \subseteq g([x]_{\simeq_G}) \cdot a \subseteq$ .

By construction,  $g \circ \eta_G = \eta_H \circ f$  and since  $\eta_G$  is connecting, by Lemma 3, the morphism  $g$  is unique. Moreover, since  $f$  and  $\eta_H$  are connecting morphisms hence so is  $\eta_H \circ f$  and so is  $g$ .  $\square$

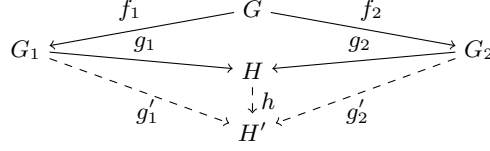
In other words, we can define the functor  $\varphi : \mathbf{CGrph}(A) \rightarrow \mathbf{UCGrph}(A)$  by  $\varphi(G) = G/\simeq_G$  for every graph  $G$  and by  $\varphi(f)$  as given by Lemma 9 for every connecting morphism  $f$ . Then, we have  $\varphi(G) = G$  for every unambiguous graph  $G$  and  $\varphi(f) = f$  for every connecting graph morphism  $f$  between unambiguous graphs. In other words,  $\varphi$  is a projection from  $\mathbf{CGrph}(A)$  into  $\mathbf{UCGrph}(A)$  henceforth a left inverse of the inclusion functor from  $\mathbf{UCGrph}(A)$  to  $\mathbf{CGrph}(A)$ .

We study a bit further the morphisms in these categories showing that they both admit pushouts. The following definition, classical in category theory, is given here for the sake of completeness.

**Definition 10 (Pushouts).** Let  $\langle f_1 : G \rightarrow G_1, f_2 : G \rightarrow G_2 \rangle$  be a pair of morphisms. A pair of morphisms  $\langle g_1 : G_1 \rightarrow H, g_2 : G_2 \rightarrow H \rangle$  is a pushout of the pair  $\langle f_1, f_2 \rangle$  when  $f_1 \circ g_1 = f_2 \circ g_2$ , and, for every other pair of morphisms  $\langle g'_1 : G_1 \rightarrow H', g'_2 : G_2 \rightarrow H' \rangle$ , if  $f_1 \circ g'_1 = f_2 \circ g'_2$  then there exists a unique morphism  $h : H \rightarrow H'$  such that  $g'_1 = h \circ g_1$  and  $g'_2 = h \circ g_2$ . Such a situation is depicted in Figure 5 below.

The first pushout lemma, in the category  $\mathbf{CGrph}(A)$ , is a slight generalization of the pushout in the category **Set**.

**Lemma 11 (Synchronization).** *In category  $\mathbf{CGrph}(A)$ , every pair of morphisms with common source has a pushout.*



**Fig. 5.** A pushout of two connecting morphisms

*Proof.* Though fairly classical, for the sake of completeness, we give here a stand alone proof.

Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$ ,  $G_1 = \langle V', \{E'_a\}_{a \in A} \rangle$  and  $G_2 = \langle V'', \{E''_a\}_{a \in A} \rangle$  be three graphs with connecting morphisms  $f_1 : G \rightarrow G_1$  and  $f_2 : G \rightarrow G_2$  as depicted in Figure 5.

Let  $G_1 + G_2$  be the graph defined as the disjoint sum of the two graphs  $G_1$  and  $G_2$ . Let  $\equiv_{f_1, f_2}$  be the equivalence relation over its set of vertices  $V' \uplus V''$  defined for every  $x, y \in V' \uplus V''$  by  $x \equiv_{f_1, f_2} y$  when  $f_x^{-1}(x) \cap f_y^{-1}(y) \neq \emptyset$  with  $f_z = f_1$  when  $z \in V'$  and  $f_z = f_2$  when  $z \in V''$ . Let then

$$H = (G_1 + G_2) / \equiv_{f_1, f_2}$$

be the quotient of the sum  $G_1 + G_2$  by this equivalence, and let

$$g_1 : G_1 \rightarrow H \text{ and } g_2 : G_2 \rightarrow H$$

be the related graph morphisms defined by  $g_1(x) = [x]_{\equiv_{f_1, f_2}}$  for every  $x \in V'$  and  $g_2(x) = [x]_{\equiv_{f_1, f_2}}$  for every  $x \in V''$ .

The fact these mapping are morphisms is obvious. One can also notice that they are one-to-one and, thanks to the gluing, they are also connecting since both  $f_1$  and  $f_2$  are connecting morphisms and vertices with common ancestors in  $G$  via  $f_1$  and/or via  $f_2$  have been connected by the quotient under relation  $\equiv_{f_1, f_2}$ . Clearly, we have  $g_1 \circ f_1 = g_2 \circ f_2$  since  $f_1(x) \equiv_{f_1, f_2} f_2(x)$  for every  $x \in V$ .

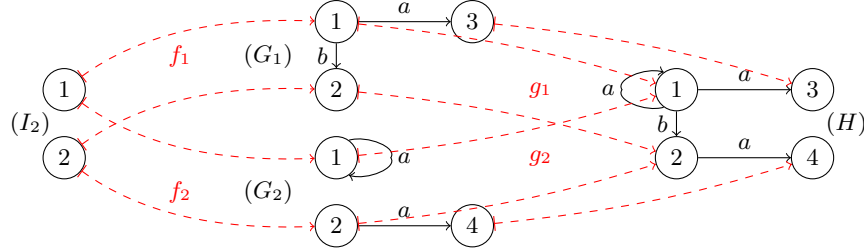
Let then  $\langle g'_1 : G_1 \rightarrow H', g'_2 : G_2 \rightarrow H' \rangle$  be another pair of connecting morphisms with graph  $H'$  such that  $g'_1 \circ f_1 = g'_2 \circ f_2$ . Let  $\simeq_{g'_1, g'_2}$  be the equivalence over  $V' \uplus V''$  defined for every  $x, y \in V' \uplus V''$  by  $x \simeq_{g'_1, g'_2} y$  whenever  $g'_x(x) = g'_y(y)$  where, for every  $z \in V' \uplus V''$ ,  $g'_z(z) = g'_1(z)$  when  $z \in V'$  and  $g'_z(z) = g'_2(z)$  when  $z \in V''$ .

Clearly,  $\equiv_{f_1, f_2} \subseteq \simeq_{g'_1, g'_2}$ . Indeed, let  $x, y \in V' \uplus V''$  such that  $x \equiv_{f_1, f_2} y$ . By definition, this means that there is  $z \in V$  such that  $x = f_x(z)$  and  $y = f_y(z)$ . But since  $g'_1 \circ f_1 = g'_2 \circ f_2$  this implies that  $g'_x \circ f_x(z) = g'_y \circ f_y(z)$  hence  $g'_x(x) = g'_y(y)$ , that is,  $x \simeq_{g'_1, g'_2} y$ .

It follows that the mapping  $h$  that maps every vertex  $[x]_{\equiv_{f_1, f_2}}$  of  $H$  to the vertex  $h([x]_{\equiv_{f_1, f_2}}) = g'_x(x)$  for every  $x \in V' \uplus V''$ , is well defined. It is then routine to check that  $h$  is a morphism, each edge of  $H$  being the image of an edge in either  $G_1$  or  $G_2$  henceforth simply propagated into  $H$  either via  $g'_1$  or via  $g'_2$ .

Then, the unicity of  $h$  just follows from the fact that, for every  $x \in V' \uplus V''$ , we must have  $h \circ g_x(x) = g'_x(x)$ .  $\square$

**Example.** An example of such a pushout in the category  $\mathbf{CGrph}(A)$  is depicted in Figure 6.



**Fig. 6.** A “synchronization” pushout example.

**Remark.** Existence of pushouts in  $\mathbf{CGrph}(A)$  essentially follows from the existence of pushouts in the category  $\mathbf{Set}$ . These pushouts are called synchronization (or glueing) pushouts since, the pushout of  $\langle f_1 : G \rightarrow G_1, f_2 : G \rightarrow G_2 \rangle$  essentially glues the vertices of  $G_1$  and  $G_2$  that have common ancestors in  $G$  either via  $f_1$  or via  $f_2$ .

The second pushout lemma, in the category  $\mathbf{UCGrph}(A)$ , is completed by a fusion phase (or glueing propagation) defined by taking the maximal unambiguous image of the graph resulting from the pushout in  $\mathbf{CGrph}(A)$ . It is called the synchronization and fusion since, building the pushout of a pair of morphisms  $\langle f_1 : G \rightarrow G_1, f_2 : G \rightarrow G_2 \rangle$  in  $\mathbf{UCGrph}(A)$  not only amount to synchronize or glue the vertices  $G_1$  and  $G_2$  that have common preimages in  $G$ , but, for the resulting graph to be unambiguous, it also amounts to perform the fusion of the resulting (possibly ambiguous) synchronized graphs by propagating this glueing.

**Lemma 12 (Synchronization and fusion).** *In category  $\mathbf{UCGrph}(A)$ , every pair of morphisms with common source has a pushout.*

*Proof.* Let  $G = \langle V, \{E_a\}_{a \in A} \rangle$ ,  $G_1 = \langle V', \{E'_a\}_{a \in A} \rangle$  and  $G_2 = \langle V'', \{E''_a\}_{a \in A} \rangle$  be three unambiguous graphs with connecting morphisms  $f_1 : G \rightarrow G_1$  and  $f_2 : G \rightarrow G_2$ .

In this case, the pushout construction goes in two steps: the synchronization steps provided by the synchronization Lemma 11 followed by the fusion step provided by the quotient under least unambiguous congruence. Then Lemma 8 allows us to conclude.

More in detail, let  $\langle g_1 : G_1 \rightarrow H, g_2 : G_2 \rightarrow H \rangle$  be the pushout in  $\mathbf{CGrph}(A)$  given by Lemma 11. Let  $U = H / \simeq_H$  be the quotient of  $H$  by the least unambiguous congruence (Lemma 7). Let  $\eta : H \rightarrow U$  be the surjective henceforth connecting morphism from  $H$  onto  $U$ . Let then  $h_1 = \eta \circ g_1$  and  $h_2 = \eta \circ g_2$ .

We claim that the pair of morphism  $\langle h_1 : G_1 \rightarrow U, h_2 : G_2 \rightarrow U \rangle$  is the pushout of  $\langle f_1, f_2 \rangle$  in the category  $\mathbf{UCGrph}(A)$ .

By Lemma 6, the graph  $H$  is unambiguous. Since  $f_1, f_2, g_1, g_2$  and  $\eta$  are connecting morphisms so are  $h_1$  and  $h_2$ , and, since  $g_1 \circ f_1 = g_2 \circ f_2$ , we have  $h_1 \circ f_1 = h_2 \circ f_2$ .

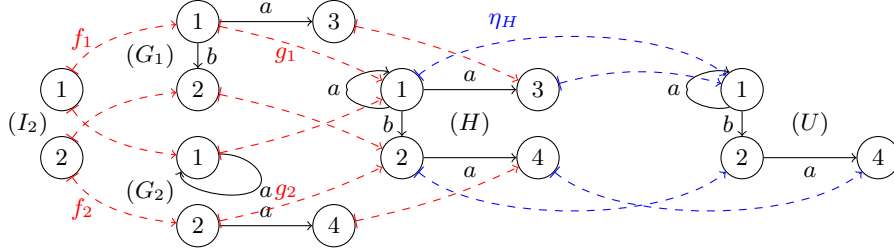
Let then  $\langle h'_1 : G_1 \rightarrow U', h'_2 : G_2 \rightarrow U' \rangle$  be another pair of connecting morphisms with unambiguous graph  $U'$  such that  $h'_1 \circ f_1 = h'_2 \circ f_2$ . Since  $\langle g_1, g_2 \rangle$  is a pushout in  $\mathbf{CGrph}(A)$  there exists a unique mapping  $h' : H \rightarrow U'$ .

We conclude by applying the maximal unambiguous graph property. Indeed, graph  $U'$  is unambiguous, with  $h' : H \rightarrow U'$  and  $\eta : H \rightarrow U = H / \simeq_H$ . It follows that, by applying Lemma 8, there exists a unique connecting morphism  $h : U \rightarrow U'$ .

The strictness of the pushout in category  $\mathbf{UCGrph}(A)$  immediately follows from the unique morphism property.

Indeed, assume that the pair  $\langle h'_1 : G_1 \rightarrow U', h'_2 : G_2 \rightarrow H'_2 \rangle$  is also a pushout. Then there exists a unique  $h' : U' \rightarrow U$ . But then, given the connecting morphisms  $f = h_1 \circ f_1 = h_2 \circ f_2$  we have  $f : G \rightarrow U$ ,  $h' \circ h : U \rightarrow U$  and  $h' \circ h \circ f = f$ . Then, by Lemma 3,  $h' \circ h$  is uniquely determined hence  $h' \circ h = id_H$ . By a similar argument, we have  $h \circ h' = id_{H'}$  and thus both  $h'$  and  $h$  are isomorphisms. This proves that  $f$  and  $f'$  are isomorphic.  $\square$

**Example.** Continuing the example given in Figure 6 we have an example of a synchronization + fusion example depicted in Figure 7.



**Fig. 7.** A “synchronization + fusion” pushout example.

## 4 The inverse monoid of birooted graphs

Before defining birooted graphs, we shall review some basic facts in category theory with pushout that lead the way towards our proposed definition of birooted graphs and related algebras. These details are given for the sake of completeness and require no prior knowledge of category theory.

More precisely, in a category that admits pushouts as above, there is a generic way to define a commutative monoid structure over (equivalence classes of) morphisms from a given object. Such a construction, that we call tensor monoids, is reviewed here. Applied to the category  $\mathbf{UCGrph}(A)$  it turns out that all morphisms commute and are, up to isomorphism, idempotents for the tensor product. The resulting algebraic is thus a meet semi-lattice.

The first step is to define the right notion of equivalence over morphisms with same domain. Unless we explicit mention a category, all definition and statement below are generic to any given category with pushouts.

**Definition 13 (“Nose” morphism of morphisms).** Let  $f_1 : G \rightarrow G_1$  and  $f_2 : G \rightarrow G_2$  be two morphisms. A nose morphism from  $f_1$  to  $f_2$  is a morphism  $k : G_1 \rightarrow G_2$  such that  $k \circ f_1 = f_2$ . This situation is denoted by  $k : f_1 \Rightarrow f_2$ .

Then, the two morphisms  $f_1$  and  $f_2$  are nose equivalent, which is denoted by  $f_1 \sim f_2$  when there exists an isomorphism  $k$  such that  $k \circ f_1 = f_2$ .

As a matter of fact, the existence of pushouts allows to define the notion of product of morphisms, via pushouts, stable under nose morphism and nose equivalence. Its definition is reviewed below.

**Definition 14 (Tensor product via pushouts).** Let  $\langle f_1, f_2 \rangle$  be a pair of morphisms with common domain. Let  $\langle g_1, g_2 \rangle$  be a pushout of the pair  $\langle f_1, f_2 \rangle$ . Then, the morphism  $g_1 \circ f_1 = g_2 \circ f_2$  is called the *tensor product* of  $f_1$  and  $f_2$  via  $\langle g_1, g_2 \rangle$ . It is denoted by  $f_1 \otimes_{g_1, g_2} f_2$ .

Then, we can prove the following property.

**Lemma 15 (Pushout stability w.r.t. nose morphisms).** Let  $f_1, f_2, f'_1$  and  $f'_2$  be four connecting morphisms with the same domain. Let

$$k_1 : f_1 \Rightarrow f'_1 \text{ and } k_2 : f_2 \Rightarrow f'_2$$

be two nose morphisms. Then, for every pushout  $\langle g_1, g_2 \rangle$  for  $\langle f_1, f_2 \rangle$  and  $\langle g'_1, g'_2 \rangle$  for  $\langle f'_1, f'_2 \rangle$  there exists a unique nose morphism

$$h : (f_1 \otimes_{g_1, g_2} f_2) \Rightarrow f'_1 \otimes_{g'_1, g'_2} f'_2$$

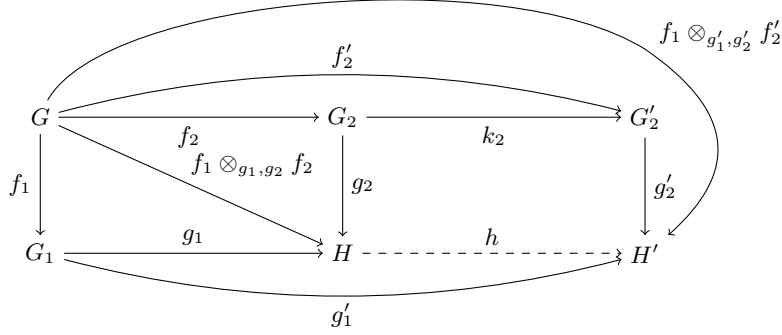
Moreover, if  $f_1 \sim f'_1$  and  $f_2 \sim f'_2$  then we have

$$f_1 \otimes_{g_1, g_2} f_2 \sim f'_1 \otimes_{g'_1, g'_2} f'_2$$

*Proof.* Let  $f_1 : G \rightarrow G_1, f_2 : G \rightarrow G_2, f'_1 : G \rightarrow G'_1$  and  $f'_2 : G \rightarrow G'_2$  be four morphisms from the same domain  $G$ . Let  $k_1 : G_1 \rightarrow G'_1$  and  $k_2 : G_2 \rightarrow G'_2$  be two morphisms such that

$$f_1 \circ k_1 = f'_1 \text{ and } f_2 \circ k_2 = f'_2$$

Let  $\langle g_1 : G_1 \rightarrow H, g_2 : G_2 \rightarrow H \rangle$  (resp.  $\langle g'_1 : G'_1 \rightarrow H', g'_2 : G'_2 \rightarrow H' \rangle$ ) be a pushout for the pair  $\langle f_1, f_2 \rangle$  (resp. the pair  $\langle f'_1, f'_2 \rangle$ ).



**Fig. 8.** Tensor product stability:  $h \circ f_1 \otimes_{g_1, g_2} f_2 = f_1 \otimes_{g'_1, g'_2} f'_2$ .

Without loss of generality, we may assume that  $G_1 = G'_1$  and  $k_1 = id_{G_1}$  henceforth  $f_1 = f'_1$ . Indeed, proving such a partial case will also prove the symmetrical (partial) case  $G_2 = G'_2$ ,  $k_2 = id_{G_2}$  and  $f_2 = f'_2$ . Then, applying these two partial results in sequence will give a proof of the complete case. Such a simplified situation depicted in Figure 8 below. Then, we observe that  $g'_1 \circ f_1 = g'_2 \circ k_2 \circ f_2$ . Since  $\langle g_1, g_2 \rangle$  is a pushout of  $\langle f_1, f_2 \rangle$ , there exists a unique morphism  $h : H \rightarrow H'$  such that  $h \circ f_1 \otimes_{g_1, g_2} f_2 = f_1 \otimes_{g'_1, g'_2} f'_2$ . This concludes the first part of the proof.

Assume now that  $f_1 \sim f'_1$  and  $f_2 \sim f'_2$ . Again, without loss of generality, we can restrict to the partial case where  $f_1 = f'_1$ . With the same notation as above, since  $f_2 \sim f'_2$  we are in the that case  $k_2$  is an isomorphism. Applying the result just obtained, this means that there exists two nose morphisms

$$h : H \rightrightarrows H' \text{ and } h' : H' \rightrightarrows H$$

such that

$$h \circ f_1 \otimes_{g_1, g_2} f_2 = f_1 \otimes_{g'_1, g'_2} f'_2 \text{ and } h' \circ f'_1 \otimes_{g_1, g_2} f_2 = f_1 \otimes_{g'_1, g'_2} f'_2$$

We aim now at proving that  $h' \circ h = id_H$  and  $h \circ h' = id_{H'}$  henceforth  $h$  is an isomorphism. By symmetry, it suffices to prove the first equality.

By combining the results stated above, we have

$$(h' \circ h) \circ f_1 \otimes_{g_1, g_2} f_2 = f_1 \otimes_{g_1, g_2} f_2$$

Let then  $g''_2 = (h' \circ h) \circ g_2$ . We have  $g''_2 \circ f_2 = (h' \circ h) \circ g_2 \circ f_2$ . Since  $g_1 \circ f_1 = g_2 \circ f_2 = f_1 \otimes_{g_1, g_2} f_2$  this implies that  $g''_2 \circ f_2 = (h' \circ h) \circ f_1 \otimes_{g_1, g_2} f_2$  hence  $g''_2 \circ f_2 = f_1 \otimes_{g_1, g_2} f_2$ , that is,  $g''_2 \circ f_2 = g_1 \circ f_1$ .

But  $\langle g_1, g_2 \rangle$  is a pushout of  $\langle f_1, f_2 \rangle$ . It follows that there is a unique morphism  $h'' : H \rightarrow H$  such that  $h'' \circ g_1 \circ f_1 = g_1 \circ f_1$  and  $h'' \circ g''_2 \circ f_2 = g_2 \circ f_2$ . But clearly  $id_H$  is such a morphism, as well as  $h' \circ h$ . It follows that  $id_H = h' \circ h$ .  $\square$

**Remark.** One can define a more general notion of (meta) morphism between two morphisms  $f_1$  to  $f_2$  as a pair  $\langle k_1, k_2 \rangle$  such that  $f_2 \circ k_1 = k_2 \circ f_1$ . However, the tensor product is no longer stable under such a more general notion of morphism. Indeed, as a counter example, let  $f_1 : I_2 \rightarrow I_2$  be the identity morphism on the two vertices graph, and let  $f_2 : I_2 \rightarrow I_2$  be the permutation of the same graph. Clearly, with  $k_1 = f_2$  and  $k_2 = f_1$  we have  $f_2 \circ k_1 = k_2 \circ f_1$ . However, while  $f_1 \otimes f_1 \sim f_1$ , we have  $f_1 \otimes f_2 \sim g$  where  $g : I_2 \rightarrow I_1$  is the (unique) mapping from the two vertex graph  $I_2$  to the one vertex graph  $I_1$ , henceforth we have  $f_1 \otimes f_1 \not\sim f_1 \otimes f_2$ .

The stability lemma (Lemma 15) allows us to define the tensor algebra of morphisms with same domain, up to nose equivalence, regardless of the pushouts that are taken.

**Definition 16 (Tensor algebras).** Let  $G$  be an unambiguous graph. Let  $M_G$  be the set of classes of connecting morphisms with domain  $G$  equivalent under nose equivalence  $\sim$ .

Then, the set  $M_G$  can be equipped with the product  $\otimes$  defined for all morphisms  $f_1$  and  $f_2$  by  $[f_1]_{\sim} \otimes [f_2]_{\sim} = [f_1 \otimes_{g_1, g_2} f_2]_{\sim}$  for some (any) pushout  $\langle g_1, g_2 \rangle$  of the pair  $\langle f_1, f_2 \rangle$ .

**Remark.** The stability property (Lemma 15) ensures the soundness of such a definition. It even allows us to view  $M_G$  as a set of representative of classes of morphisms emanating from  $G$ , simply denoting by  $f_1 \otimes f_2$  one (any) representative of the equivalence classes product  $[f_1]_{\simeq} \otimes [f_2]_{\simeq}$ . Then, thanks to Lemma 15, proving the validity of any equation in  $M_G$  over equivalent classes of morphisms just amounts to prove the validity of this equation over some of the representative of these classes.

**Theorem 17 (Semi-lattices of morphisms).** *In the category  $\mathbf{UCGrph}(A)$ , for every unambiguous graph  $G$ , the tensor algebra  $\langle M_G, \otimes \rangle$  is a commutative idempotent monoid with neutral element  $id_G$ .*

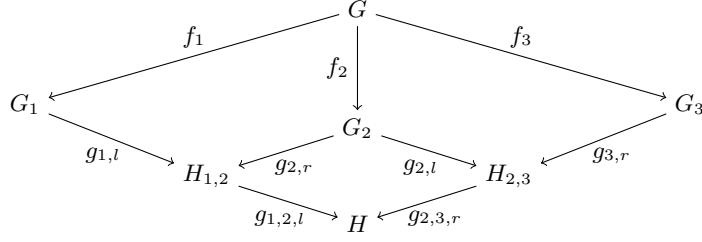
*Moreover, given the order induced  $\leq$  defined by  $f \leq g$  when  $f = g \otimes f$ , then the tensor product  $\otimes$  is the meet for the order  $\leq$  and we have  $g \leq f$  if and only there is a nose morphism  $k : f \Rightarrow g$ .*

*Proof.* Let  $G$  be an unambiguous graph. We first prove that the tensor algebra  $\langle M_G, \otimes, id \rangle$  is a commutative monoid with unit  $id_G$ . Clearly, the symmetry of pushouts shows that the tensor product is commutative. Let us prove its is associative.

Let  $f_1 : G \rightarrow G_1$ ,  $f_2 : G \rightarrow G_2$  and  $f_3 : G \rightarrow G_3$  three morphisms emanating from  $G$  with pushouts  $\langle g_{1,l} : G_1 \rightarrow H_{1,2}, g_{2,r} : G_2 \rightarrow H_{1,2} \rangle$  for the pair  $\langle f_1, f_2 \rangle$  and  $\langle g_{2,l} : G_2 \rightarrow H_{2,3}, g_{3,r} : G_3 \rightarrow H_{2,3} \rangle$  for the pair  $\langle f_2, f_3 \rangle$ . Let then  $\langle g_{1,2,l} : H_{1,2} \rightarrow H, g_{2,3,r} : H_{2,3,r} \rightarrow H \rangle$  be the pushout of the pair  $\langle g_{2,l}, g_{2,r} \rangle$ . This situation is depicted in Figure 9

It is then routine to check that  $\langle g_{1,2,l}, g_{2,3,r} \circ g_{3,r} \rangle$  is a pushout of  $\langle f_1 \otimes f_2, f_3 \rangle$  and that  $\langle g_{1,2,l} \circ g_{1,l}, g_{2,3,r} \rangle$  is a pushout of  $\langle f_1, f_2 \otimes f_3 \rangle$ , which conclude the proof that  $(f_1 \otimes f_2) \otimes f_3 = f_1 \otimes (f_2 \otimes f_3)$ .





**Fig. 9.** Associativity up to nose morphism.

Last, we prove that  $id_G$  is a neutral element for the tensor product. For such a purpose, let  $f : G \rightarrow H$  be a morphism. We claim that  $\langle id_H, f \rangle$  is the pushout of the pair  $\langle f, id_G \rangle$  from which we will have that  $f \otimes id_G = f$ .

The expected morphism equality is clearly satisfied. Let  $g'_1 : H \rightarrow H'$  and  $g'_2 : G \rightarrow H'$  such that  $g'_1 \circ f = g'_2 \circ id_G$ . Clearly taking  $h = g'_1$ , we have  $h : H \rightarrow H'$  with  $g'_1 = h \circ id_H$  and  $g'_2 = h \circ f$ . Moreover, since  $h \circ id_H = h$ , this implies that  $h = g'_1$  hence such a morphism is unique.

So far, our proof is generic to arbitrary category that has pushouts. In the category  $\mathbf{UCGrph}(A)$ , we furthermore can show that every morphism is, up to isomorphism, idempotent under the tensor product.

Let  $f : G \rightarrow H$  be a connecting morphism with unambiguous graphs  $G = \langle V, \{E_a\}_{a \in A} \rangle$  and  $H = \langle V', \{E'_a\}_{a \in A} \rangle$ . We want to prove that  $f \otimes f = f$ . It suffice to prove that the pair  $\langle id_H, id_H \rangle$  is the pushout of the pair  $\langle f, f \rangle$ . The expected morphism equality is satisfied. It remains to prove the universality property.

Let  $\langle g_1 : H \rightarrow H', g_2 : H \rightarrow H' \rangle$  be a pair of connecting morphism with unambiguous  $H'$  such that  $g_1 \circ f = g_2 \circ f$ . We want to prove that there exists a (unique) morphism  $h : H \rightarrow H'$  such that  $h \circ f = g_1 \circ f = g_2 \circ f$ . Thanks to Lemma 3, it suffice to prove that such a morphism exists.

Let  $x \in V'$  be a vertex in  $H$ . Since  $f$  is a connecting morphism, there exists  $z \in V$  and  $w \in (A + \bar{A})^*$  such that  $x = f(z) \cdot w$ . It follows that  $g_1(x) = g_1(f(z) \cdot w)$ . But since both  $g_1$  is connecting and  $H'$  is unambiguous, we have  $g_1(x) = g_1 \circ f(z) \cdot w$ . Similarly, we also have  $g_2(x) = g_2 \circ f(z) \cdot w$ . But since  $g_1 \circ f = g_2 \circ f$ , this means that  $g_1(x) = g_2(x)$  hence  $g_1 = g_2$ . Then we can take  $h = g_1 = g_2$  to conclude.

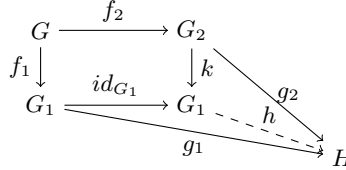
It is routine to check that the (associative, commutative and idempotent) tensor product is the meet of the order defined in  $M_G$  by  $f \leq g$  when  $f = f \otimes g$ .

It remains to prove the last statement: the correspondance between this order and the existence of morphism. Let  $f_1 : G \rightarrow H_1$  and  $f_2 : G \rightarrow H_2$  be two morphisms in  $\mathbf{UCGrph}(A)$ .

Assume that  $f_1 \leq f_2$ , that is, given a pushout  $\langle g_1 : G_1 \rightarrow H, g_2 : G_2 \rightarrow H \rangle$  of the pair  $\langle f_1, f_2 \rangle$ , there is a nose isomorphism  $k : f_1 \otimes_{g_1, g_2} f_2 \Rightarrow f_1$ . By definition,

we have  $f_1 \otimes_{g_1, g_2} f_2 = g_1 \circ f_1 = g_2 \circ f_2$  with  $k \circ (f_1 \otimes_{g_1, g_2} f_2) = f_1$ , it follows that  $k \circ g_2 \circ f_2 = f_1$ . In other words, there is the nose morphism  $k \circ g_2 : f_2 \Rightarrow f_1$ .

Conversely, assume that there is a nose morphism  $k : f_2 \Rightarrow f_1$ . This means that  $k : G_2 \rightarrow G_2$  with  $f_1 = k \circ f_2$ . Then we claim that the pair  $\langle id_{G_1}, k \rangle$  is a pushout of  $\langle f_1, f_2 \rangle$ . This situation is depicted Figure 10. Clearly, we have



**Fig. 10.** The case there is a nose morphism  $k : f_2 \rightarrow f_1$ .

$id_{G_1} \circ f_1 = k \circ f_2$ . Let then  $\langle g_1 : G_1 \rightarrow H, g_2 : G_2 \rightarrow H \rangle$  such that  $g_1 \circ f_1 = g_2 \circ f_2$ .

Then we can take  $h = g_1$ . Clearly, we have  $g_1 = h \circ id_{G_1}$  and  $h \circ k = g_2$ , and the first equality ensures that  $h$  is uniquely determined by  $g_1$ .  $\square$

We are now ready to define birooted graphs as certain cospans in the category  $\mathbf{UCGrph}(A)$ . For such a purpose, for every integer  $k > 0$ , let  $I_k$  be the unambiguous defined by  $k$  distinct vertices  $\{1, 2, \dots, k\}$  and empty edge relations, and let  $id_k : I_k \rightarrow I_k$  be the identify isomorphism.

**Definition 18 (Birooted graphs).** A birooted graph  $B$  is a pair of connecting morphisms

$$B = \langle in : I_p \rightarrow G, out : I_q \rightarrow G \rangle$$

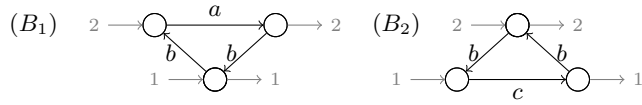
from two trivial graphs  $I_p$  and  $I_q$  to a common unambiguous graph  $G$ .

The morphism  $in$  is called the *input root morphism*, or, more simply, the *input root* of the birooted graph  $B$ . The morphism  $out$  is called the *output root morphism*, or, more simply, the *output root* of the birooted graph  $B$ .

The pair of positive integers  $(p, q)$  that defines the domains of root morphisms is called the *type* of the birooted graph. It is denoted by  $dom(B)$ . The underlying graph  $G$  is the codomain of the input and output morphisms. It is called the *graph* of  $B$  and it is also denoted by  $cod(B)$ .

**Remark.** A birooted graph of type  $(p, q)$  can simply be seen as a unambiguous graph  $G = \langle V, \{E_a\}_{a \in A} \rangle$  enriched with two tuples of distinguished vertices  $(x_1, x_2, \dots, x_p) \in V^p$  and  $(y_1, y_2, \dots, y_q) \in V^q$  that label the vertices marked by the input and the output roots of the birooted graph.

This point of view is depicted in Figure 11 with two birooted graphs  $B_1$  and  $B_2$  of type  $(2, 2)$ . In such a figure, vertices of input roots are marked by dangling input arrows, and vertices of output roots are marked by dangling output arrows.

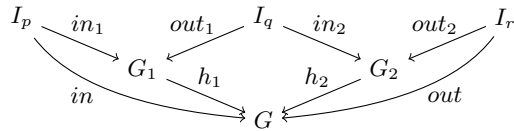


**Fig. 11.** Examples of  $(2, 2)$ -bierooted graphs.

**Remark.** The name “bierooted graphs” is borrowed from [30]. However, our definition is a clear generalization of the definition given in [30]. Indeed, Stephen’s bierooted graphs are only bierooted graphs of type  $(1, 1)$ .

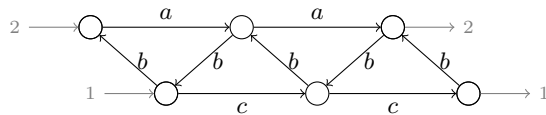
In category theoretical term, a bierooted graph is a cospan (see for instance [4]). The existence of pushouts in the category  $\mathbf{UCGrph}(A)$  allows us to define the product of bierooted graphs as the product of their cospan. However, such a product is (so far) not uniquely determined since, a priori, it may depend on the chosen pushout.

**Definition 19 (Bierooted graph product instance).** Let  $B_1 = \langle in_1, out_1 \rangle$  and let  $B_2 = \langle in_2, out_2 \rangle$  be two bierooted graphs. Assume that  $B_1$  is of type  $(p, q)$  and that  $B_2$  is of type  $(q, r)$ . Let  $\langle h_1, h_2 \rangle$  be a pushout of the pair  $\langle out_1, in_2 \rangle$ . Then, the *product instance* of bierooted graphs *via the pushout*  $\langle h_1, h_2 \rangle$  is defined to be the bierooted graphs  $\langle h_1 \circ in_1, h_2 \circ out_2 \rangle$ , and it is denoted by  $B_1 \cdot_{h_1, h_2} B_2$ . Such a situation is depicted in Figure 12.



**Fig. 12.** The pushout diagram of a product instance  $B_1 \cdot_{h_1, h_2} B_2$ .

A concrete example of a product instance built from the  $(2, 2)$ -bierooted graphs given in Figure 11 is depicted in Figure 13.



**Fig. 13.** A product instance of  $B_1 \cdot B_2 \cdot B_1 \cdot B_2$ .

We aim now at defining products of bierooted graphs up to some adequate notion of bierooted graph equivalence. This is done via the notion of bierooted graph

morphisms (Definition 20) and the proof that birooted graph product instances are stable under birooted graph morphisms (Lemma 22).

**Definition 20 (Birooted graph morphisms).** Let  $B_1 = \langle in_1, out_1 \rangle$  and  $B_2 = \langle in_2, out_2 \rangle$  be two birooted graphs. A birooted graph morphism from  $B_1$  to  $B_2$  is defined as root preserving graph morphism of their codomain, that is, a graph morphism  $h : cod(B_1) \rightarrow cod(B_2)$  such that  $in_2 = h \circ in_1$  and  $out_2 = h \circ out_1$ . Such a morphism is denoted by  $h : B_1 \Rightarrow B_2$ .

Two birooted graphs  $B_1$  and  $B_2$  are isomorphic when there is an isomorphism  $h : B_1 \Rightarrow B_2$ . Such a situation is denoted by  $B_1 \sim B_2$ .

**Lemma 21 (Unicity of birooted graph morphism).** *Let  $B_1$  and  $B_2$  be two birooted graphs. Then there exists at most one morphism  $h : B_1 \Rightarrow B_2$ . Moreover, in the case there exists a morphism  $h : B_1 \Rightarrow B_2$  and a morphism  $h' : B_2 \Rightarrow B_1$  then  $h$  and  $h'$  are isomorphisms and  $B_1 \sim B_2$ .*

*Proof.* Thanks to the definition, since roots are connecting morphisms, a birooted graph morphism is necessarily a connecting morphism. Then, the unicity of birooted graph morphisms follows from Lemma 3. In particular, when there exists  $h : B_1 \Rightarrow B_2$  and  $h' : B_2 \Rightarrow B_1$  then we have  $h' \circ h : B_1 \rightarrow B_1$  and  $h \circ h' : B_2 \rightarrow B_2$ . By unicity, with  $cod(B_1) = G_1$  and  $cod(B_2) = G_2$ , this implies that  $h' \circ h = id_{G_1}$  and  $h \circ h' = id_{G_2}$  hence  $B_1$  and  $B_2$  are isomorphic.  $\square$

**Lemma 22 (Product stability w.r.t. birooted graphs morphisms).** *Let  $f_1 : B_1 \Rightarrow C_1$  and  $f_2 : B_2 \Rightarrow C_2$  be two birooted graphs morphisms and let  $B_1 \cdot B_2$  and  $C_1 \cdot C_2$  be two product instances. Then, there exists a (unique) birooted graphs morphisms  $h : B_1 \cdot B_2 \Rightarrow C_1 \cdot C_2$ .*

*Proof.* Let then  $B_1 = \langle in_1, out_1 \rangle$ ,  $B_2 = \langle in_2, out_2 \rangle$ ,  $C_1 = \langle in'_1, out'_1 \rangle$  and  $C_2 = \langle in'_2, out'_2 \rangle$ . Let also  $f_1 : B_1 \Rightarrow C_1$ , that is,  $f_1 \circ in_1 = in'_1$  and  $f_1 \circ out_1 = out'_1$ , and,  $f_2 : B_2 \Rightarrow C_2$ , that is,  $f_2 \circ in_2 = in'_2$  and  $f_2 \circ out_2 = out'_2$ . This situation is depicted Figure 14. We aim at building  $h : G \rightarrow G'$  such that  $h \circ in = in'$  and  $h \circ out = out'$ . For such a purpose, let us consider

$$g_1 = h'_1 \circ f_1 : G_1 \rightarrow G' \text{ and } g_2 = h'_2 \circ f_2 : G_2 \rightarrow G'$$

Clearly, we have  $g_1 \circ out_1 = g_2 \circ in_2$ . But since  $\langle h_1, h_2 \rangle$  is a pushout of  $\langle out_1, in_2 \rangle$ , this implies that there is a unique morphism  $h : G \rightarrow G'$  such that factorizes  $g_1$  and  $g_2$  through  $h_1$  and  $h_2$ , that is, such that

$$g_1 = h \circ h_1 \text{ and } g_2 = h \circ h_2$$

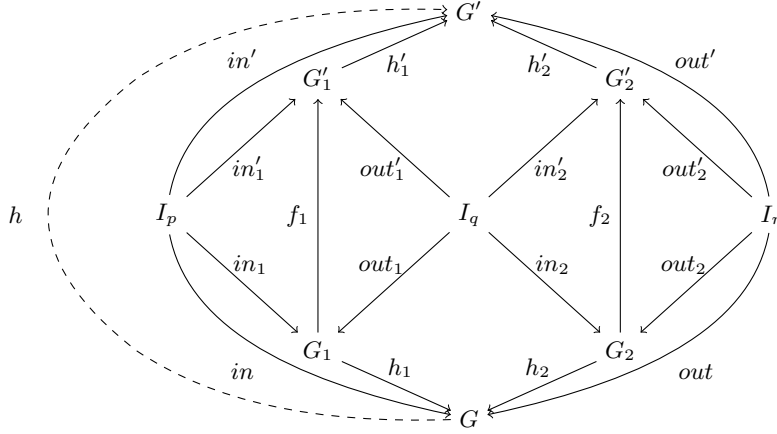
Replacing  $g_1$  and  $g_2$  by their definitions, we obtain

$$h'_1 \circ f_1 = h \circ h_1 \text{ and } h'_2 \circ f_2 = h \circ h_2$$

This implies that

$$\underbrace{h'_1 \circ f_1 \circ in_1}_{in'} = h \circ \underbrace{h_1 \circ in_1}_{in} \text{ and } \underbrace{h'_2 \circ f_2 \circ in_2}_{out'} = h \circ \underbrace{h_2 \circ in_2}_{out}$$

hence the desired results.  $\square$



**Fig. 14.** Building a morphism  $B_1 \cdot B_2 \Rightarrow C_1 \cdot B_2$ .

This stability property allows us to define the following birooted graph algebras.

**Definition 23 (Birooted graph algebras).** Let  $HS(A)$  be the set of classes of isomorphic birooted graphs extended with the empty set equipped with the product defined for every  $X, Y \in H(S)$  as follows. In the case there is  $B \in X$ ,  $C \in Y$  and a product instance  $B \cdot C$ , then we take  $X \cdot Y = [B]_{\sim} \cdot [Y]_{\sim} = [B \cdot Y]_{\sim}$  and we take  $X \cdot Y = \emptyset$  in all other cases.

**Notation.** In the sequel we shall simply write  $B$  (or  $C$ ) instead of  $[B]$  (or  $[C]$ ) and we shall simply write  $B \cdot C$  for the product  $[B]_{\sim} \cdot [C]_{\sim}$  of the corresponding classes of equivalent birooted graphs.

**Theorem 24 (Semigroup property).** *The algebra  $HS(A)$  is semigroup, that is, the product of birooted graphs is an associative operation.*

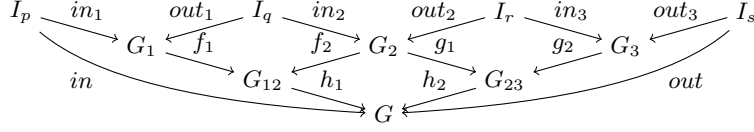
*Proof.* Associativity of the product follows from the existence of pushouts in  $\mathbf{UCGrph}(A)$  (Lemma 12) and unicity up to isomorphism thanks to unicity (Lemma 3).

More precisely, let  $B_i = \langle in_i, out_i \rangle$  for  $i = 1, 2, 3$  be three birooted graphs. First we easily observe that  $(B_1 \cdot B_2) \cdot B_3 = 0$  if and only if  $B_1 \cdot (B_2 \cdot B_3) = 0$  since this is just a matter of type compatibility. It thus remains to prove the case when both products are non-zero.

Let  $\langle f_1, f_2 \rangle$  (resp.  $\langle g_1, g_2 \rangle$ ) be a pushout of  $\langle out_1, in_2 \rangle$  (resp.,  $\langle out_2, in_3 \rangle$ ). Let  $B_{1,2} = B_1 \cdot B_2 = \langle f_1 \circ in_1, f_2 \circ out_2 \rangle$  and,  $B_{2,3} = B_2 \cdot B_3 = \langle g_1 \circ in_2, g_2 \circ out_3 \rangle$  be the corresponding product instances. Let also  $\langle h_1, h_2 \rangle$  be a pushout of  $\langle f_2, g_1 \rangle$  and let  $B = \langle h_1 \circ f_1 \circ in_1, h_2 \circ g_2 \circ out_3 \rangle$  be the resulting birooted graphs. This situation is depicted in Figure 15.

Now, we aim at proving that, up to isomorphism, we have:

$$B = B_{1,2} \cdot B_3 = B_1 \cdot B_{2,3}$$



**Fig. 15.** Associativity of the product (up to isomorphisms) from combinations of pushouts

thus proving the associativity of the product in  $HS(A)$ . For this purpose, we just check that:

$$\langle h_1, h_2 \circ g_2 \rangle \text{ is a pushout of } \underbrace{\langle f_2 \circ out_2, in_3 \rangle}_{out_{12}}$$

hence

$$\langle G_{12}, f_1 \circ in_1, f_2 \circ out_2 \rangle \cdot \langle G_3, in_3, out_3 \rangle = \underbrace{\langle h_1 \circ f_1 \circ in_1, h_2 \circ g_2 \circ out_3 \rangle}_{in \quad out}$$

and that

$$\langle h_1 \circ f_1, h_2 \rangle \text{ is a pushout of } \underbrace{\langle out_1, f_1 \circ in_2 \rangle}_{in_{23}}$$

hence

$$\langle G_1, in_1, out_1 \rangle \cdot \langle G_{12}, g_1 \circ in_2, g_2 \circ out_3 \rangle = \underbrace{\langle h_1 \circ f_1 \circ in_1, h_2 \circ g_2 \circ out_3 \rangle}_{in \quad out}$$

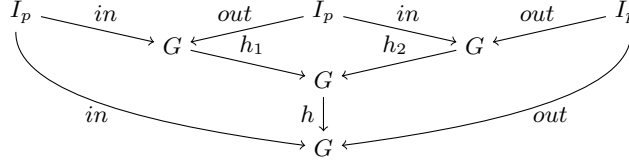
where, thanks to stability (Lemma 22) birooted graph equality should be understood up to isomorphisms.  $\square$

**Lemma 25 (Idempotent property).** *A non-zero birooted graph  $B$  of the form  $B = \langle in, out \rangle$  is idempotent, that is,  $B \cdot B = B$ , if and only if  $in = out$ . Moreover, idempotent birooted graphs commute henceforth form a subsemigroup.*

*Proof.* Let  $B = \langle in, out \rangle$  be a birooted graph with domain  $G$ , assumed to be idempotent (up to isomorphism). This implies that  $B$  is of type  $(p, p)$  for some  $p > 0$ . The situation is depicted in Figure 16 with isomorphism  $h : G \rightarrow G$ . where  $\langle h_1, h_2 \rangle$  is the pushout of  $\langle out, in \rangle$ .

Since  $h \circ h_1 \circ in = in$  this means that  $h \circ h_1$  agrees with  $id_G$  on the vertices of  $G$  in the image of  $in$ . But since  $in$  is connecting, by Lemma 3 we have  $h \circ h_1 = id_G$ . Quite similarly, since  $h \circ h_2 \circ out = out$  we also have  $h \circ h_2 = id_G$ . But then, since  $h_1 \circ out = h_2 \circ in$ , this implies that  $h \circ h_1 \circ out = h \circ h_2 \circ in$  henceforth  $in = out$ .

Conversely, when  $in = out$ , we know (Theorem 17) that  $in \otimes out \sim in \sim out$  hence a pushout of  $\langle in, in \rangle$  is given by  $\langle id_G, id_G \rangle$  henceforth  $B$  is idempotent.



**Fig. 16.** An idempotent product  $\langle G, in, out \rangle \cdot \langle G, in, out \rangle \sim \langle G, in, out \rangle$

As a consequence of this characterization, we also observe that the product of idempotent birooted graphs coincide with the (extension with zero) of the tensor product of morphisms. It follows that Theorem 17 applies proving that idempotent birooted graphs commute henceforth they form a subsemigroup of the semigroup of birooted graphs.  $\square$

**Theorem 26 (Inverse semigroup property).** *The semigroup  $HS(A)$  is an inverse semigroup, that is, for every element  $B$ , there is a unique element  $B^{-1}$  such that*

$$B \cdot B^{-1} \cdot B = B \text{ and } B^{-1} \cdot B \cdot B^{-1} = B^{-1}$$

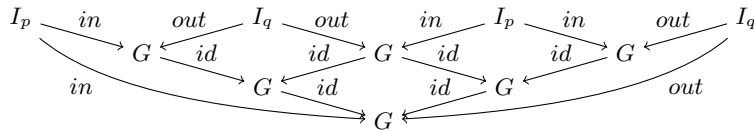
*The inverse  $B^{-1}$  of a non-zero birooted graph  $B = \langle in, out \rangle$  is simply given by  $B^{-1} = \langle out, in \rangle$ .*

*Proof.* It suffices to prove that every element  $B \in HS(A)$  has an inverse. Indeed, thanks to Lemma 25, we know that idempotent element commute and, following the theory [24], this implies the unicity of inverse.

Since 0 has itself has inverse, it remains to prove the case on a non-zero  $B \in HS(A)$ . Let then  $B = \langle in : I_p \rightarrow G, out : I_q \rightarrow G \rangle$ . We aim at proving that  $C = \langle out, in \rangle$  is an inverse of  $B$ . By symmetry, it suffices to prove the equality

$$B \cdot C \cdot B = B$$

However, thanks to Lemma 17, we know that  $in \otimes in = in$  and  $out \otimes out = out$ . Then, the expected equality follows from the definition, as depicted in Figure 17 where all diamond diagrams are clearly pushouts.



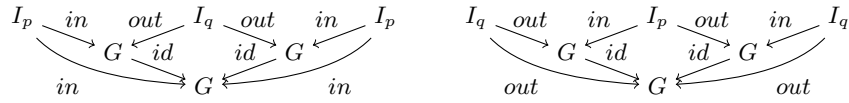
**Fig. 17.** A product instance  $\langle in, out \rangle \cdot \langle out, in \rangle \cdot \langle in, out \rangle$ .

Inverses allow us to define left and right projections that, following inverse semigroup theory, characterize left and right Green classes.

**Definition 27 (Left and right projection).** Let  $B \in HS(A)$  be a birooted graph. The left projection  $B^L$  of the birooted graph  $B$  is defined by  $B^L = B^{-1} \cdot B$ . The right projection  $B^R$  of the birooted graph  $B$  is defined by  $B^R = B \cdot B^{-1}$ .

**Lemma 28.** Let  $B = \langle in, out \rangle$  be a non-zero birooted graph. Then we have  $B^L = \langle out, out \rangle$  and  $B^R = \langle in, in \rangle$ .

*Proof.* This can easily be deduce from the pushout diagrams induced by these product with, again thanks to Theorem 17,  $in \otimes in = in$  and  $out \otimes out = out$  as depicted in Figure 18 below.  $\square$



**Fig. 18.** Right and left projection diagrams pushouts diagrams.

**Remark.** As in any inverse semigroup, it is an obvious observation that an element  $B$  is idempotent if and only if it is self-inverse, that is, when  $B = B^{-1}$ , henceforth, as already observed in Lemma 25, in the non-zero case, when it is of the form  $\langle f, f \rangle$ .

**Remark.** As a general matter of fact, the relation  $B \preceq C$  defined over birooted graphs when there exists a morphism  $h : C \Rightarrow B$  is a preorder (or quasi-order) relation. Thanks to identities, it is reflexive, and thanks to morphism composition it is transitive. Then, Lemma 21 above proves that the equivalence induced by such a preorder relation is the isomorphism relation over birooted graphs. In other words, the notion of morphisms induces an order relation on the set  $HS(A)$ . We shall see now that this order over  $HS(A)$  induced by morphisms has an algebraic characterization in inverse semigroup theory: it is the natural order [24].

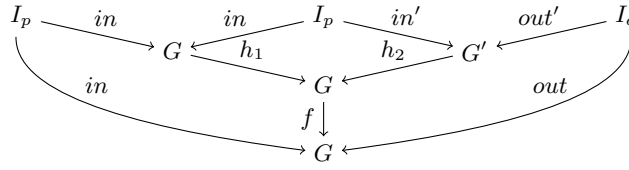
**Definition 29 (Natural order).** The natural order  $\leq$  is defined over birooted graphs by  $B \leq C$  when  $B = B^R \cdot C$  (or, equivalently,  $B = C \cdot B^L$ ).

**Theorem 30 (Natural order vs birooted graph morphisms).** In the inverse semigroup  $HS(A)$ , the absorbant element  $0$  is the least element under the natural order and, for every pair of non zero birooted graphs  $B$  and  $C$ ,  $B \leq C$  if, and only if, there is a birooted graph morphism  $h : C \Rightarrow B$ .

*Proof.* The fact  $0$  is the least element in the natural order is immediate. Let  $B = \langle in, out \rangle$  and  $C = \langle in', out' \rangle$  be two non-zero birooted graphs.

Assume that  $B \leq C$ , that is,  $B^R \cdot C$  is isomorphic with  $B$ . Let  $f : B^R \cdot C \Rightarrow B$  be such an isomorphism. By applying Lemma 28 that shows  $B^R = \langle in, in \rangle$ , the resulting situation is depicted Figure 19. Since  $f \circ h_1 \circ in = in$ , by applying the





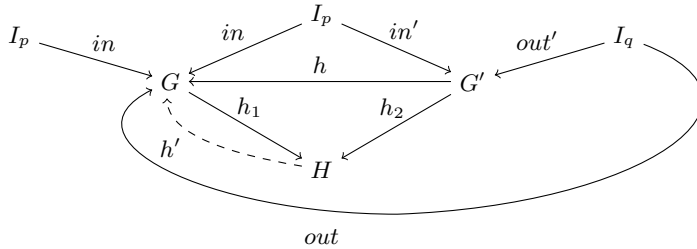
**Fig. 19.** The case  $B^R \cdot C \sim B$ .

unicity lemma (Lemma 8) we have  $f \circ h_1 = id_G$ .

Let then  $h = f \circ h_2$ . We have  $h \circ in' = f \circ h_2 \circ in'$ . But  $h_2 \circ in' = h_1 \circ in$  hence  $h \circ in' = f \circ h_1 \circ in$  and thus  $h \circ in' = in$ . Since  $h \circ out' = out$  we thus have proved that  $h$  is a morphism  $h : C \Rightarrow B$  from the birooted graph  $C$  to the birooted graph  $B$ .

Conversely, assume that there is a morphism  $h : C \rightarrow B$ . That is,  $h : G' \rightarrow G$  such that  $h \circ in' = in$  and  $h \circ out' = out$ .

This situation, together with the product of  $B^R \cdot C$  via a pushout  $\langle h_1, h_2 \rangle$ , is depicted Figure 20. We aim at proving that  $h_1 : B \Rightarrow B^R \cdot C$  is an isomorphism.



**Fig. 20.** The product  $B^R \cdot C$  with  $h : C \Rightarrow B$ .

Given that  $B^R \cdot C = \langle h_1 \circ in, h_1 \circ out' \rangle$ , this amounts to prove that  $h_1$  is a graph isomorphism and  $h_1 \circ in = h_1 \circ in$  (which is clear) and  $h_1 \circ out = h_2 \circ out'$  (which is also clear). In other words, it remains to prove that  $h_1$  is an isomorphism.

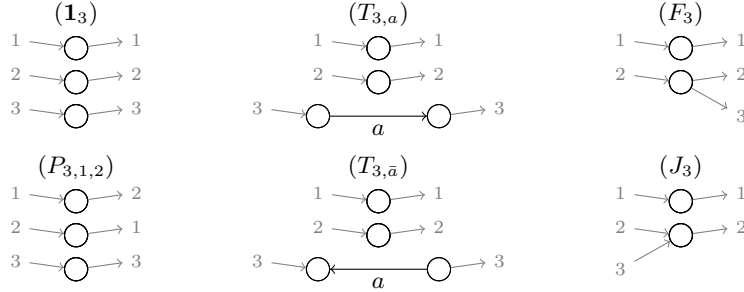
Let  $g_1 = id_G$  and  $g_2 = h$ . By construction, we have  $g_1 \circ in = g_2 \circ in'$ . Since  $\langle h_1, h_2 \rangle$  is a pushout of  $\langle in, in' \rangle$ , this implies that there is a unique morphism  $h' : H \rightarrow G$  such that  $g_1 = h' \cdot h_1$  and  $g_2 = h' \cdot h_2$ . This means in particular that  $id_G = h' \cdot h_1$ . But this also means that  $h_1 \circ h' \circ h_1 = h_1$  hence, since  $h_1$  is connecting, by Lemma 3, we also have  $id_H = h_1 \circ h'$ .  $\square$

The inverse semigroup of birooted graphs gives a fairly simple though mathematically robust way to compose birooted graphs one with the other. Now we aim at characterizing a simple set of generators for this semigroup.

Clearly, as the number of vertex in root morphisms domain may be arbitrarily big, there exists no finite set of generators. More precisely, root morphisms must

be connecting morphisms. Then, a simple connectivity argument shows that the birooted graph  $Id_n = \langle id_n, id_n \rangle$  component cannot be generated from birooted graph generator with root morphisms domain  $I_k$  with  $k < n$ .

**Definition 31 (Elementary birooted graphs).** A elementary birooted graph is either zero or any birooted graph among  $I_m, P_{m,i,j}, T_{m,a}, T_{m,\bar{a}}, F_m$  or  $J_m$  defined below. In the case  $m = 3$  these graphs are depicted in Figure 21.



**Fig. 21.** Elementary birooted graphs.

Formally, the birooted graph  $P_{m,i,j} = \langle id_m : I_m \rightarrow I_m, out : I_m \rightarrow I_m \rangle$  is defined for any  $m > 0$  and  $1 \leq i, j \leq m$  by  $out(i) = j, out(j) = i$  and  $out(k) = k$  for every other  $1 \leq k \leq m$ . It is called a *root permutation*. As a particular case, when  $i = j$ , since  $P_{m,i,i} = \langle id_m, id_m \rangle$ , the birooted graph  $P_{m,i,i}$  is denoted by  $\mathbf{1}_m$  instead and called a *root identity*.

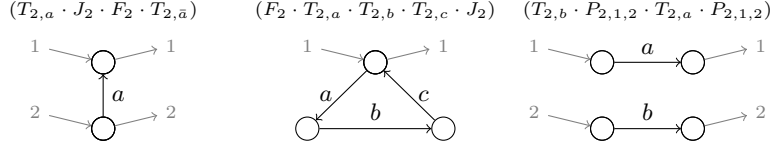
The birooted graphs  $F_m = \langle id_{m-1} : I_{m-1} \rightarrow I_{m-1}, out : I_m \rightarrow I_{m-1} \rangle$  and  $J_m = \langle in : I_m \rightarrow I_{m-1}, id_{m-1} : I_{m-1} \rightarrow I_{m-1} \rangle$  are defined for any  $m > 1$ , by  $in(m) = out(m) = m - 1$  and  $in(k) = out(k) = k$  for every  $1 \leq k \leq m - 1$ . They are called a *root fork* and a *root join*.

The birooted graph  $T_{m,a} = \langle in : I_m \rightarrow G_a, out : I_m \rightarrow G_a \rangle$  is defined for any  $m > 0$  and  $a \in A$ , by  $G_a$  being the  $m + 1$  vertex graph with set of vertices  $V = \{1, \dots, m, m + 1\}$  and sets of edges  $E_a = \{(m, m + 1)\}$  and  $E_b = \emptyset$  for every  $b \neq a$ , with  $in(m) = m, out(m) = m + 1$  and  $in(k) = out(k) = k$  for every other  $1 \leq k < m$ . It is called a *forward edge*. The birooted graph  $T_{m,\bar{a}} = T_{m,a}^{-1}$  is called a *backward edge*.

**Examples.** Some birooted graphs generated by elementary graphs are depicted in Figure 22.

**Theorem 32.** Every birooted graphs  $\langle in : I_p \rightarrow G, out : I_q \rightarrow G \rangle$  with  $n$  vertices in  $G$  is finitely generated from 0 and the elementary birooted graphs  $\mathbf{1}_k, P_{k,i,j}, T_{k,a}, T_{k,\bar{a}}, F_k$  and  $J_k$  with  $1 \leq k \leq \max(n, p + 1, q + 1)$ .

*Proof.* Let  $B = \langle in : I_p \rightarrow G, out : I_q \rightarrow G \in HS(A) \rangle$  with  $G = \langle V, \{E_a\}_{a \in A} \rangle$ . The proof that  $B$  can be built as a product of elementary birooted graphs goes



**Fig. 22.** Some elementary compositions.

by induction on the triple  $n(B)$  by the number of edges, the number of roots and the number of vertices

$$n(B) = (\sum_{a \in A} |E_a|, p + q, |V|)$$

ordered lexicographically.

In the smallest possible case, we have  $n(B) = (0, 1, 1)$ , that is,  $G = I_1$  and  $B = \mathbf{1}_1$ , nothing has to be done.

Assume now that the statement is true for all birooted graphs  $B'$  with  $n(B') < n(B)$ . Then three cases are possible.

We first examine the three possible cases of edges connecting an input or an output root to any other vertex.

In the first case, there is  $i \in \{1, \dots, p\}$  and  $x \in A + \bar{A}$  such that  $in(i) \cdot x = y$  is defined. Possibly taking  $P_{p,i,p} \cdot B$  instead of  $B$ , we may assume that  $i = p$ . Indeed, we easily check that  $B = P_{p,i,p} \cdot P_{p,i,p} \cdot B$  so the original  $B$  can be recovered by composing it with the same root permutation.

Let then  $B' = \langle in' : I_{p+1} \rightarrow G', out' : I_q \rightarrow G' \rangle$  be the birooted graph defined by:

- the graph  $G'$  obtained the graph  $G$  by keeping the same set of vertices and by removing the edge  $(x, y)$  from  $E_x$  when  $x \in A$  or the edge  $(y, x) \in E_{\bar{x}}$  when  $x \in \bar{A}$ ,
- taking  $in'(k) = in(k)$  for every  $1 \leq k \leq p$  and  $in(p+1) = y$ ,
- taking  $out' = out$ .

Since the number of edges is strictly smaller, we have  $n(B') < n(B)$  hence the induction hypothesis applies and we conclude by observing that we have  $B = F_{p+1} \cdot T_{p+1,x} \cdot B'$ .

In the second case, there is  $j \in \{1, \dots, q\}$  and  $x \in A + \bar{A}$  such that  $out(j) \cdot x = y$  is defined. Again, possibly taking  $B \cdot P_{q,j,q}$  we may assume that  $j = q$ .

Then, using a symmetrical argument, we can define  $B'$  with  $n(B') < n(B)$  such that we have  $B = B' \cdot T_{q+1,x} \cdot J_{q+1}$  and we conclude by applying the induction hypothesis.

In the third case, for every  $x \in A + \bar{A}$ , every  $i \in \{1, \dots, q\}$  and every  $j \in \{1, \dots, p\}$  we have  $in(i) = \emptyset = out(j)$ . Since both  $in$  and  $out$  are connecting morphisms this implies that  $E_a = \emptyset$  for every  $a \in A$ .

Then we examined if some vertex are double input or double output root. Three subcases are possible.

In the first subcase, there are  $i, j \in \{1, \dots, p\}$  with  $i \neq j$  such that  $in(i) = in(j)$ . Possibly taking  $P_{p,i,p} \cdot P_{p,j,p-1} \cdot B$  instead of  $B$  we may assume that  $i = p$  and  $j = p-1$ . Indeed, we may then recover the initial birooted graph just taking back  $P_{p,j,p-1} \cdot P_{p,i,p} \cdot B$  since, for all  $p, i, j$  and all  $B$ , we have  $P_{p,i,j} \cdot P_{p,i,j} = \mathbf{1}_p$  and  $\mathbf{1}_p \cdot B = B$ .

Let then  $B' = \langle in', out \rangle$  be the birooted graph obtained from  $B = \langle in, out \rangle$  just by taking the restriction  $in' : I_{p-1} \rightarrow G$  of  $in : I_p \rightarrow G$  to the set  $\{1, p-1\}$ . We easily observe that  $B = J_p \cdot B'$ . Since the number of edges is the same and the number of roots is strictly smaller, we have  $n(B') < n(B)$ . We conclude by observing that  $B = J_p \cdot B'$  and applying the induction hypothesis on  $B'$ .

In the second subcase there are  $i, j \in \{1, \dots, q\}$  with  $i \neq j$  such that  $out(i) = out(j)$ . By applying a symmetrical argument, we can show that there exists  $B'$  with  $n(B') < n(B)$  such that  $B = B' \cdot F_q$  and we conclude similarly by applying the induction hypothesis on  $B'$ .

In the third case, we may assume that both  $in$  and  $out$  are injective morphisms. Since there are no edges and both  $in$  and  $out$  are connecting morphisms, this implies that  $p = q = |V|$  and both  $in$  and  $out$  are isomorphisms from  $I_{|V|}$  into  $I_{|V|}$ . Thus there is a bijection  $p$  over  $\{1, 2, \dots, |V|\}$  such that  $out = p \circ in$ . As well know, every bijection over a finite set  $V$  can be generated by a sequence  $(i_k, j_k)_{1 \leq k < p}$  of  $p$  transpositions (possibly repeating the identity) over  $V$  such that  $p = (i_p, j_p) \circ \dots \circ (i_2, j_2) \circ (i_1, j_1)$ . It follows that

$$B = P_{p,i_p,j_p} \cdots P_{p,i_2,j_2} \cdot P_{p,i_1,j_1}$$

This concludes the case study and the induction proof argument.

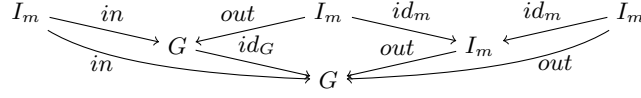
We conclude the proof by checking the correctness of the announced bound on the number of roots used in intermediate elementary birooted graphs.  $\square$

**Definition 33 (Bounded birooted graphs algebras).** For any given integer  $m > 0$ , let  $HS_m(A)$  (resp.  $HS_{\leq m}(A)$ ) be the algebraic structure defined as the subsemigroup of  $HS(A)$  generated by  $\mathbf{1}_m, P_{m,i,j}, T_{m,a}, T_{m,\bar{a}}$  (resp.  $\mathbf{1}_k, P_{k,i,j}, T_{k,a}, T_{k,\bar{a}}, F_k$  and  $J_k$  with  $1 \leq k \leq m$ ).

As an corollary of Theorems 24 and 26, we have:

**Theorem 34.** *For every integer  $m > 0$ , the algebra  $HS_m(A)$  is an inverse monoid with neutral element  $\mathbf{1}_m$ .*

*Proof.* We observe first that the birooted graphs of  $HS_m(A)$  are necessarily of type  $(m, m)$ . Since, by definition,  $HS_m(A)$  is closed under product, it suffices to prove that  $\mathbf{1}_m$  is a neutral element. For this, we observe that a pushout of a pair of morphisms of the form  $\langle id_m : I_m \rightarrow I_m, r : I_m \rightarrow G \rangle$  is the pair  $\langle r : I_m \rightarrow G, id_G : G \rightarrow G \rangle$ . From this fact we easily deduce that the birooted



**Fig. 23.** The birooted (right) unit case

graph  $\mathbf{1}_m = \langle id_m, id_m \rangle$  is, up to birooted graph isomorphism, a right unit for the product of  $(m, m)$ -birooted graphs as depicted in Figure 23.

A symmetrical argument shows that  $\langle id_m, id_m \rangle$  it is also a left unit for the product.  $\square$

**Remark.** As a particular case, it can be shown that  $HS_1(A)$  is the free inverse monoid  $FIM(A)$  generated by  $A$ . We shall see below that birooted grids of arbitrary size but of type  $(2, 2)$  belong to  $HS_{\leq 2}(A)$ . In other word, in Theorem 32, the bound given for  $k$ , depending on the number of vertices of  $G$  is not optimal.

**Remark.** In the category  $UCGrph(A)$  there is a notion of cospans, that is, pairs of morphisms of the form

$$\langle in : G_{in} \rightarrow G, out : G_{out} \rightarrow G \rangle$$

with arbitrary input graph  $G_{in}$  and output graph  $G_{out}$ . However, in our definition of birooted graphs, we restrict to trivial graphs  $G_{in}$  and  $G_{out}$  of the form  $I_k$ . One may wonder what is the loss induced by such a restriction.

It turns out that, up to compatibility criteria in the product, arbitrary cospans may be encoded into birooted graphs. More precisely, for every unambiguous graph  $G$ , given the number  $n$  of its vertices, let  $h_G : I_n \rightarrow G$  be a (fixed) canonical injection morphism from the trivial graph with  $I_n$  into  $G$ . Then, up to cospan isomorphism (defined just as for birooted graph), every cospan

$$C = \langle in : G_{in} \rightarrow G, ou : G_{out} \rightarrow G \rangle$$

can be represented, by the birooted graph

$$B(C) = \langle h_{G_{in}}, h_{G_{in}} \rangle \cdot \langle in \circ h_{G_{in}}, out \circ h_{G_{out}} \rangle \cdot \langle h_{G_{out}}, h_{G_{out}} \rangle$$

More precisely:

**Lemma 35 (Co-span encoding stability).** *Let  $C_1$  and  $C_2$  be two cospans. If  $C_1 \sim C_2$  then  $B(C_1) \sim B(C_2)$ , i.e. the encoding preserve isomorphism equivalence. However, it is false that if  $B(C_1) \sim B(C_2)$  then  $C_1 \sim C_2$ .*

*Denoting by  $C_1 \cdot C_2$  the isomorphic classes of the cospan product of  $C_1$  and  $C_2$ , if  $C_1 \cdot C_2$  is non-empty,  $B(C_1 \cdot C_2) = B(C_1) \cdot B(C_2)$ , i.e. the encoding preserve product.*

*Proof.* Immediate from the definitions. The morphism is not injective (one-to-one) since, through the proposed encoding  $B(C)$ , the interface  $\langle G_{in}, G_{out} \rangle$  that defines the cospan compatibility in product is partially lost. Both graphs  $G_1$  and  $G_2$  have been merged into (some morphic image of)  $G$ .

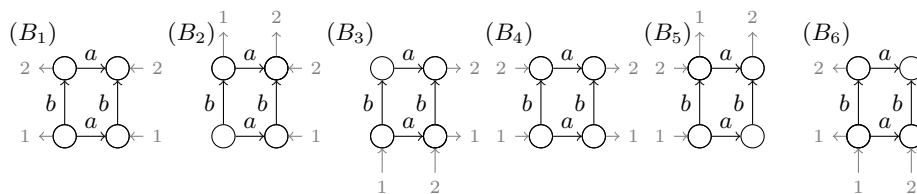
## 5 Languages of birooted graphs

Now we aim at developing the language theory of higher dimensional strings, that is to say, the study of the definability of subsets of  $HS(A)$ . For such a purpose, we consider the First Order ( $FO$ ) logic or the Monadic Second Order ( $MSO$ ) logic (see [9]) on birooted graphs. We refer the reader to the book [9] for a definition of  $MSO$  on graphs.

More precisely, we consider  $HS_{\leq m}(A)$  so that the number of input and output roots on graphs is bounded. Then, one can enrich the signature  $A$  by  $2 * m$  symbols, necessarily interpreted as singletons in order to describes these roots. Clearly, this is easily done within  $FO$  or  $MSO$  logic and we can thus consider the class of  $F$ -definable or  $MSO$ -definable languages of birooted graphs.

**Theorem 36 (Undecidability).** *When  $m \geq 2$ , the language emptiness problem for  $FO$ -definable (henceforth  $MSO$ -definable) languages of birooted graphs of  $HS_{\leq m}(A)$  is undecidable.*

*Proof (sketch of).* The undecidability of  $FO$  follows from the fact that, as soon as  $m \geq 2$ , as depicted in Figure 24, grids of arbitrary size can be finitely generated henceforth classical undecidability results applies [9].  $\square$

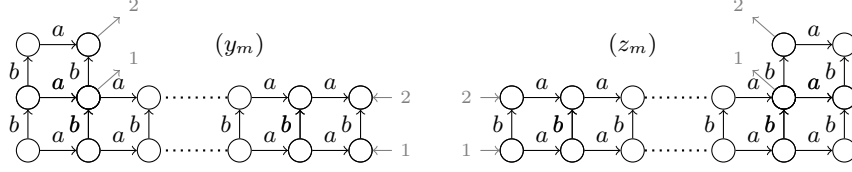


**Fig. 24.** A finite set of generators  $B_1, B_2, B_3, B_4, B_5$  and  $B_6$ .

We first check, following the examples depicted in Figure 22, that these generators can indeed be defined by means of  $P_{k,i,j}$ ,  $T_{k,a}$ ,  $T_{k,\bar{a}}$ ,  $F_k$  and  $J_k$  with  $1 \leq k \leq 2$ . For instance, we have  $B_5 = (T_{2,\bar{b}} \cdot J_2)^R \cdot T_{2,a} \cdot T_{2,b} \cdot (T_{2,a} \cdot J_2)^R \cdot P_{2,1,2}$ .

Then, as depicted in Figure 25, we can generate birooted grids of arbitrary size by taking the  $(2, 2)$ -birooted graph  $B_{m,n}$  defined by  $G_{m,n} = (Z_m \cdot Y_m)^n$ . Clearly,  $B_{mn}$  contains a grid of size  $m$  by  $2 * n$ .

One may ask how generating such graphs of unbounded tree-width can be avoided. It occurs that this can simply be done by restricting the overlaps that are allowed in product instances. More precisely, observing the examples above, one can notice that every product involved in generating  $Y_m$  or  $Z_m$  is fairly simple. They essentially consists in synchronizing the involved graphs. There is a trivial fusion phase as in Lemma 11.



**Fig. 25.** The  $(2, 2)$ -bierooted graphs  $Y_m = (B_1)^m \cdot B_2 \cdot B_3$  and  $Z_m = (B_4)^m \cdot B_5 \cdot B_6$ .

On the contrary, when defining the product  $y_m \cdot z_m$ , the gluing phase involved more than  $2 * m$  vertices, glued one by one. There is a non-trivial fusion phase in the sense of Lemma 12.

Recently introduced in the context of bierooted words [16] or trees [17] languages, the definition of the disjoint product, extended to bierooted graphs, makes this restriction of overlaps formal.

**Definition 37 (Disjoint product).** Let  $B_1 = \langle in_1, out_1 \rangle$  and  $B_2 = \langle in_2, out_2 \rangle$  be two bierooted graphs. Let  $\langle h_1, h_2 \rangle$  be a pushout of  $\langle out_1, in_2 \rangle$  in  $\mathbf{UCGrph}(A)$  and let  $B_1 \cdot B_2 \langle in, out \rangle$  with  $in = h_1 \circ in_1$  and  $out = h_2 \circ out_2$  be the resulting product. Then this product is a *disjoint product* when the pair  $\langle h_1, h_2 \rangle$  is also a pushout of  $\langle out_1, in_2 \rangle$  in the category  $\mathbf{CGrph}(A)$ . In this case, the disjoint product is denoted by  $B_1 \star B_2$ .

In other words, a bierooted graph product is a disjoint product when the fusion phase in the underlying pushout computation is trivial. Although partially defined, this disjoint product is still associative in the following sense.

**Lemma 38 (Partial associativity).** *For all bierooted graphs  $B_1, B_2, B_3$  the disjoint product  $B_1 \star (B_2 \star B_3)$  is defined if and only if the disjoint product  $(B_1 \star B_2) \star B_3$  is defined and, in that case, the products are equal.*

*Proof.* Let  $f : I_m \rightarrow G$  be a morphism in  $\mathbf{UCGrph}(A)$ . We define the interface type  $\tau(f)$  of the morphism  $f$  to be the mapping  $\tau(f) : \{1, \dots, m\} \rightarrow \mathcal{P}(A + \bar{A})$  defined by  $\tau(f)(k) = \{x \in A + \bar{A} : f(k) \cdot x \neq \emptyset\}$  for every  $1 \leq k \leq m$ . Then, two such a mapping  $f : I_m \rightarrow G_1$  and  $g : I_n \rightarrow G$  are said *disjoint compatible* when  $m = n$  and for every  $1 \leq k \leq m$  we have  $\tau(f)(k) \cap \tau(g)(k) = \emptyset$ .

Then, given any bierooted graphs  $B_1 = \langle in_1 : I_{p_1} \rightarrow G_1, out_1 : I_{q_1} \rightarrow G_1 \rangle$  and  $B_2 = \langle in_2 : I_{p_2} \rightarrow G_1, out_2 : I_{q_2} \rightarrow G_2 \rangle$ , we can easily check, thanks to the proof of the pushout Lemmas 11 and 12, that the disjoint product  $B_1 \star B_2$  is defined if and only if  $out_1$  and  $in_2$  are disjoint compatible.

Then, checking that such a compatibility condition is associative in the above sense is routine checking. □

Then, the closure under disjoint products and left and right projections are defined as follows.

**Definition 39 (Disjoint closure and decomposition).** Let  $X \subseteq HS(A)$  be a set of birooted graphs. The *disjoint closure* of the set  $X$  is defined to be the least set  $Y$  of birooted graphs such that  $X \subset Y$  and that  $Y$  is closed under disjoint product and left and right projections. This closure is denoted by  $\langle X \rangle_{\star, L, R}$ .

For every birooted graph  $B \in \langle X \rangle_{\star, L, R}$ , a combination of elements of  $X$  by disjoint products and left and right projection that equals  $B$  is called a *disjoint decomposition* of  $B$  over  $X$ .

**Examples.** The subset of  $HS_1(A)$  generated by disjoint products of elementary birooted graphs  $I_1$  and  $T_{1,a}$  with  $a \in A$  is just the free monoid  $A^*$ . Adding left and right projections, the disjoint closure of such a set is known in the literature as the free ample monoid  $FAM(A)$  whose elements are positive birooted trees (see [13]). Adding backward edges  $T_{1,\bar{a}}$  for every  $a \in A$ , the disjoint closure of the resulting set is the free inverse monoid  $FIM(A)$  whose elements are arbitrary birooted trees.

**Theorem 40 (Decidability and complexity).** Let  $X \subseteq_{fin} HS(A)$  be a finite subset of  $HS(A)$ . Then, the emptiness problem for MSO-definable subsets of the disjoint closure  $\langle X \rangle_{\star, R, L}$  is (non-elementary) decidable.

Moreover, for any MSO-definable language  $L \subseteq \langle X \rangle_{\star, R, L}$ , the membership problem  $B \in L$  for any  $B \in HS(A)$  is linear in the size of any disjoint decomposition of  $B$  over  $X$ .

*Proof (sketch of).* Every disjoint product in  $\langle X \rangle_{\star, R, L}$  is just a disjoint sum with a bounded glueing of roots. It follows that MSO decomposition techniques (see [28] or [31]) combined with partial algebra techniques [7] are available, as done in [5] for languages of labeled birooted trees, to achieve an algebraic characterization of MSO definable languages in terms of (partial algebra) morphisms into finite structures. Such an approach also proves the complexity claim for the membership problem.  $\square$

**Remark.** Of course, the membership problem is non elementary in the size of the MSO formula that defines  $L$ . This already follows from the case of MSO definable languages of finite words. Also, the problem of finding disjoint decompositions over  $X$  for birooted graphs may be delicate and is left for further studies.

**Remark.** Another possible proof for the above theorem may consist in showing that disjoint products and projections preserves graphs tree-width. More precisely, for every birooted graph  $B = \langle in : I_p \rightarrow G, out : I_q \rightarrow G \rangle$ , let  $twd(B)$  the maximum of  $p$ ,  $q$  and the tree-width of the graph  $G$ . Then we can show that for every birooted graph  $B_1$  and  $B_2$  we have  $twd(B_1^L) = twd(B_1)$ ,  $twd(B_2^R) = twd(B_2)$ , and if the disjoint product  $B_1 \star B_2$  is defined then we have  $twd(B_1 \star B_2) = \max(twd(B_1), twd(B_2))$ . It follows that, since  $X$  is finite, all graphs of  $\langle X \rangle_{\star, R, L}$  have bounded tree-width henceforth MSO would be decidable (see [9]). Then, the complexity claim for the membership problem would



follow, for instance, from the algebraic presentation of recognizable languages of graphs via cospans studied in [6].

As observed above,  $A^*$ ,  $FAM(A)$  and  $FIM(A)$  are examples of subsemigroup of  $HS(A)$  that are finitely generated by disjoint product, inverses and/or projections. By applying Theorem 40, this proves (again) that their MSO definable subsets have decidable emptiness problem.

Of course, any subsemigroup of  $HS(A)$  that would equal the disjoint closure of some of its finite subset would also have decidable (MSO) emptiness problem. However, finding such examples seems to be delicate. Thanks to [23], E-unitary inverse semigroups with virtually free group image may induce subsemigroup of  $HS(A)$  with decidable MSO language theory. Other examples need to be discovered.

Last, as an illustration of the power of the inverse semigroup framework that is proposed here, we show how birooted acyclic graphs can easily be defined as the quotient of the inverse semigroup of birooted graphs by the semigroup ideal of cyclic ones. More precisely:

**Lemma 41 (Semigroup ideal).** *Let  $\varphi$  be a graph property that is preserved under graph morphisms. Let  $I_\varphi$  be the set  $I_\varphi \subseteq HS(A)$  that contains 0 and all birooted graphs whose underlying graph satisfies  $\varphi$ . Then,  $I_\varphi$  is an semigroup ideal of  $HS(A)$ , that is,*

$$HS(A) \cdot I_\varphi \subseteq HS(A) \text{ and } I_\varphi \cdot HS(A) \subseteq HS(A)$$

and the Rees' quotient  $HS(A)/I_\varphi$ , that is, the set  $HS(A) - I_\varphi + \{0\}$  equipped with the product defined as in  $H(A)$  when the result does not belong to  $I_\varphi$  and defined to be 0 otherwise, is still an inverse semigroup.

*Proof.* Let  $\varphi$  be a graph property such that, for every graph  $G_1$  and  $G_2$ , every graph morphism  $\varphi : G_1 \rightarrow G_2$ , if  $G_1 \models \varphi$  then  $G_2 \models \varphi$ .

Let then  $B_1 = \langle in_1 : I_p \rightarrow G_1, I_q \rightarrow G_1 \rangle$  and  $B_2 = \langle in_2 : I_q \rightarrow G_2, I_r \rightarrow G_2 \rangle$  be two birooted graphs and let  $\langle h_1 : G_1 \rightarrow G, h_2 : G_2 \rightarrow G \rangle$  be a pushout of  $\langle out_1, in_2 \rangle$ . By definition,  $B_1 \cdot B_2 = \langle h_1 \circ in_1, h_2 \circ out_2 \rangle$ .

In the case we have  $G_1 \models \varphi$  (resp.  $G_2 \models \varphi$ ) then, by applying the remark above, thanks to the existence of morphism  $h_1$  (resp. morphism  $h_2$ ) we have  $G \models \varphi$ . In other words, in both case  $B_1 \in I_\varphi$  or  $B_2 \in I_\varphi$ , we have  $B_1 \cdot B_2 \in I_\varphi$ . Since  $0 \in I_\varphi$  this proves that  $I_\varphi$  is an ideal of  $HS(A)$ .

The fact that the quotient of an inverse semigroup by an ideal is still an inverse semigroup is routine.  $\square$

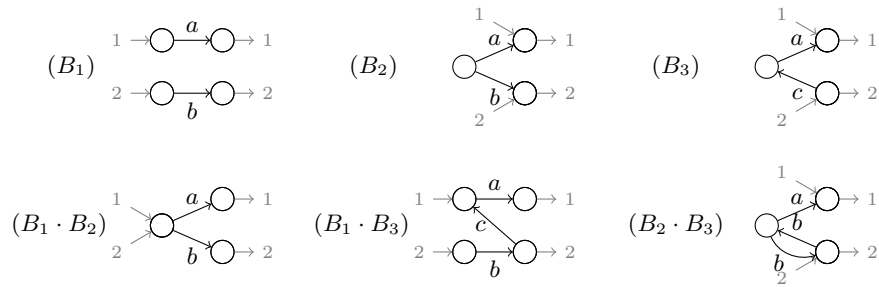
In other words, much in the same way 0 already appears with products in  $HS(A)$  that have no compatible types, when the property  $\varphi$  describes, in some concrete modeling context, a set of faulty models that is preserves under morphism, then the product in  $HS(A)/I_\varphi$  equals 0 also when the resulting birooted graph is faulty.

An application of this result is the modeling of causality by means of edge direction. More precisely, assume from now on that in graphs, vertices are system

(partial) states, and directed edges between vertices are causal (local) transition. Indeed, under such a modeling assumption, every birooted graph whose underlying graph has a (directed) cycle is faulty with respect to strict causality. In other words, we would like to restrict to birooted (directed) acyclic graphs. This can easily be done as follows.

Clearly, the existence of directed cycles is a property preserved by morphism. Then, the algebra of birooted acyclic graphs can simply be modeled as the inverse semigroup  $HS(A)/I_C$  where  $I_C \subseteq HS(A)$  is the resulting semigroup ideal containing 0 and all (directed) cyclic birooted graphs.

Such a situation is depicted in Figure 26 where examples show how products



**Fig. 26.** Causal constraints propagation via products.

of birooted graphs may propagate causality constraints eventually leading to non-causal graphs: the product  $(B_2 \cdot B_2)$ .

In other words, with the proposed approach, one can define a modeling software in such a way that non-causal models raised by combination of causal constraints are easily detected and forbidden, while, at the same time, the underlying algebraic framework still lays in the theory of inverse semigroups.

## 6 Conclusion

We have shown how a rather simple and intuitive composition operation on graphs, inheriting from long standing ideas (see [25]), induces a rich algebraic structure, an inverse semigroup, from which one can define a natural order and other mathematically robust operators such as left and right projections, that capture graph theoretical concepts.

Of course, defining graph products by means of cospans products has already a long history in Theoretical Computer Science (see e.g. [14, 6, 4]). The originality of our approach consists in restricting to the category of unambiguous graphs and connecting morphisms that allow the resulting semigroup to be an inverse semigroup.

Still, this inverse semigroup is far from being understood in the depth. Little is known about its subsemigroups. Thanks to [30], one can easily show that,

all  $A$  generated  $E$ -unitary inverse semigroups (see [24]) are subsemigroups of the monoid defined by birooted graphs of type (1,1). This suggests that the semigroup  $HS(A)$  may satisfy some universality property that is still to be discovered. Also, we have no direct characterization of the subsemigroups of  $HS(A)$  that could be defined by bounding the number of roots on generators.

By restricting the product to disjoint product, techniques arising from partial algebras [7] are applicable allowing us to inherit from the existing MSO-language theory of graphs of bounded tree-width [8,9]. Yet, closure property of MSO-definable languages remains to be detailed. Also, defining suitable subsemigroups of (possible Rees' quotient of)  $HS(A)$  that would also have decidable MSO languages is still to be investigated.

With a view towards application, beyond all experiments mentioned in the introduction, the modeling power of birooted graphs also needs to be investigated further in both practical modeling problems and more general modeling theories. For such a purpose, an implementation of the monoid  $HS(A)$  with both graphical and programmatic views of its elements is scheduled.

## Acknowledgements

Although the main idea presented here essentially remains the same, its presentation, completely changed, has benefited from the numerous and helpful comments of anonymous referees of former versions of this works.

It must also be mentioned that the idea of developing a notion of higher dimensional strings has been suggested to us by Mark V. Lawson during the summer 2012.

## References

1. S. Abramsky. A structural approach to reversible computation. *Theor. Comp. Sci.*, 347(3):441–464, 2005.
2. J.-R. Abrial. *Modeling in Event-B - System and Software Engineering*. Cambridge University Press, Cambridge, 2010.
3. F. Berthaut, D. Janin, and B. Martin. Advanced synchronization of audio or symbolic musical patterns: an algebraic approach. *International Journal of Semantic Computing*, 6(4):409–427, 12 2012.
4. C. Blume, H.J.S. Bruggink, M. Friedrich, and B. König. Treewidth, pathwidth and cospan decompositions with applications to graph-accepting tree automata. *Journal of Visual Languages and Computing*, 24(3):192 – 206, 2013.
5. A. Blumensath and D. Janin. A syntactic congruence for languages of birooted trees. *Semigroup Forum*, 2014.
6. H.J.S. Bruggink and B. König. On the recognizability of arrow and graph languages. In *Graph Transformations*, volume 5214 of *LNCS*, pages 336–350. Springer, 2008.
7. P. Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras*. Akademie-Verlag, 1986.
8. B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability. *Theor. Comp. Sci.*, 80(2):153–202, 1991.

9. B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012.
10. V. Danos and L. Regnier. Reversible, irreversible and optimal lambda-machines. *Theor. Comp. Sci.*, 227(1-2):79–97, 1999.
11. V. Diekert, M. Lohrey, and A. Miller. Partially commutative inverse monoids. *Semigroup Forum*, 77(2):196–226, 2008.
12. V. Diekert, N. Ondrusch, and M. Lohrey. Algorithmic problems on inverse monoids over virtually free groups. *Int. Jour. of Algebra and Comp.*, 18(01):181–208, 2008.
13. J. Fountain, G. Gomes, and V. Gould. The free ample monoid. *Int. Jour. of Algebra and Comp.*, 19:527–554, 2009.
14. F. Gadducci and R. Heckel. An inductive view of graph transformation. In *Recent Trends in Algebraic Development Techniques, 12th International Workshop, WADT'97, Selected Papers*, pages 223–237, 1997.
15. P. Hudak and D. Janin. Tiled polymorphic temporal media. In *Work. on Functional Art, Music, Modeling and Design (FARM)*, pages 49–60. ACM Press, 2014.
16. D. Janin. Overlapping tile automata. In A.A. Bulatov and A.M. Shur, editors, *8th Int. Computer Science Symp. in Russia (CSR)*, volume 7913 of *LNCIS*, pages 431–443. Springer, 06 2013.
17. D. Janin. On languages of labeled birooted trees: Algebras, automata and logic. *Information and Computation*, 2014.
18. D. Janin, F. Berthaut, and M. Desainte-Catherine. Multi-scale design of interactive music systems : the libTuiles experiment. In *Sound and Music Comp. (SMC)*, 2013.
19. D. Janin, F. Berthaut, M. DeSainte-Catherine, Y. Orlarey, and S. Salvati. The T-calculus : towards a structured programming of (musical) time and space. In *Work. on Functional Art, Music, Modeling and Design (FARM)*, pages 23–34. ACM Press, 2013.
20. J. Kellendonk. The local structure of tilings and their integer group of coinvariants. *Comm. Math. Phys.*, 187:115–157, 1997.
21. J. Kellendonk and M. V. Lawson. Tiling semigroups. *Journal of Algebra*, 224(1):140 – 150, 2000.
22. J. Kellendonk and M. V. Lawson. Universal groups for point-sets and tilings. *Journal of Algebra*, 276:462–492, 2004.
23. D. Kuske and M. Lohrey. Logical aspects of cayley-graphs: the group case. *Ann. Pure Appl. Logic*, 131(1-3):263–286, 2005.
24. M. V. Lawson. *Inverse Semigroups : The theory of partial symmetries*. World Scientific, 1998.
25. H. R. Lewis. A new decidable problem, with applications (extended abstract). In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 62–73. IEEE Press, 1977.
26. S. W. Margolis and J. C. Meakin. Inverse monoids, trees and context-free languages. *Trans. Amer. Math. Soc.*, 335:259–276, 1993.
27. J. Meakin. Groups and semigroups: connections and contrasts. In *Groups St Andrews 2005, Volume 2*, London Mathematical Society, Lecture Note Series 340. Cambridge University Press, 2007.
28. S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.
29. P. V. Silva. On free inverse monoid languages. *ITA*, 30(4):349–378, 1996.
30. J.B. Stephen. Presentations of inverse monoids. *Journal of Pure and Applied Algebra*, 63:81–112, 1990.

31. W. Thomas. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. In *Structures in Logic and Computer Science*, volume 1261 of *LNCS*, pages 118–143. Springer, 1997.
32. W. Thomas. Logic for computer science: The engineering challenge. In *Informatics - 10 Years Back, 10 Years Ahead.*, volume 2000 of *LNCS*, pages 257–267. Springer, 2001.