Representing 2-Dimensional Critical Pairs

Samuel Mimram*

May 31, 2013

Abstract

Polygraphs generalize to n-categories the usual notion of equational theory, thus allowing one to describe a category by the means of generators and relations. When the relations are oriented, such a presentation can be considered as a rewriting system and one might wonder whether the rewriting system is confluent and terminating in order to provide a notion of canonical representative of morphisms modulo equations (the normal forms of the morphisms). In term rewriting systems, confluence is often proved by computing the critical pairs, which are in finite number, and showing that they are joinable. We extend here this methodology to polygraphs presenting 2-categories. This task is not straightforward because a finite polygraph might admit an infinite number of critical pairs. This leads us to introduce the multicategory of contexts of the free compact 2-category generated by a 2-category, in which we can embed the original 2-category generated by the polygraph and compute a finite number of morphisms which generate all the critical pairs. We also introduce polygraphic nets, which are a concrete representation of contexts. These theoretical tools allow us to finally describe an algorithm for computing generating families of critical pairs in 2-dimensional polygraphs.

Term rewriting systems have proven very useful to reason about terms modulo equations. In some cases, the equations can be oriented and completed in a way giving rise to a convergent (that is both confluent and terminating) rewriting system, thus providing a notion of canonical representative of equivalence classes of terms. Usually, the terms are freely generated by a *signature* $(\Sigma_n)_{n\in\mathbb{N}}$, which consists of a family of sets Σ_n of generators of arity n, and one considers *equational theories* on such a signature, which consist of *equations* formalized by pairs of terms freely generated by the signature. For example, the equational theory of monoids contains two generators m and e, whose arities are respectively 2 and 0, and three equations

$$m(m(x,y),z) = m(x,m(y,z)) \qquad m(e,x) = x \quad \text{and} \quad m(x,e) = x \tag{1}$$

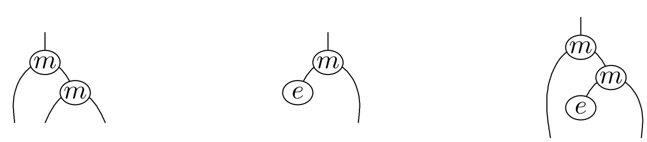
*CEA LIST, Laboratory for the Modelling and Analysis of Interacting Systems, Point Courrier 94, 91191 Gif-sur-Yvette, France. E-mail: `samuel.mimram@cea.fr`. This work was started while I was in the PPS team (CNRS – Univ. Paris 7) and has been supported by the CHOCO ("Curry Howard pour la Concurrence", ANR-07-BLAN-0324) French ANR.

If we consider these equations as oriented from left to right, and call them rewriting rules, they form a *rewriting system* on terms generated by the signature. One says that a term t rewrites to a term u, what we write $t \Rrightarrow u$, if u can be obtained from t by replacing in t an occurrence of a left member of a rewriting rule by the corresponding right member. Such a rewriting system is *terminating* when there is no infinite rewriting sequence $t \Rrightarrow t_1 \Rrightarrow t_2 \Rrightarrow \ldots$ for some term t. It is *confluent* when given a term t and two rewriting sequences $t \Rrightarrow \ldots \Rrightarrow u_1$ and $t \Rrightarrow \ldots \Rrightarrow u_2$ the terms u_1 and u_2 are joinable, which means that there exists a term v and rewriting sequences $u_1 \Rrightarrow \ldots \Rrightarrow v$ and $u_2 \Rrightarrow \ldots \Rrightarrow v$: in such a rewriting system, the order in which rewriting rules are applied does not really matter on the long run.

The rewriting systems which are both terminating and confluent, are called *convergent*. These are particularly interesting because any maximal rewriting sequence starting from a given term t will lead to a unique term $\hat{t}$, called the normal form of t, thus providing a canonical representative of terms modulo equations. The confluence of a term rewriting system can be deduced from joinability of *critical pairs*, which are minimal possible obstructions to confluence. For example, the system (1) can be shown to be terminating by giving an interpretation of the terms in a well-founded poset, such that the rewriting rules are strictly decreasing. Moreover, it can be verified that the pair of terms generated by the five critical pairs

$$m(m(m(x,y),z),t) \quad m(m(e,x),y) \quad m(m(x,e),y) \quad m(m(x,y),e) \quad m(e,e)$$

are joinable. The framework of term rewriting systems is very nice because these critical pairs are always in finite number when there is a finite number of rewriting rules, and they can be computed using a unification algorithm: given two terms t and u, such an algorithm computes minimal terms such that when we replace the variables by those, t and u become equal. For example, $x \mapsto e$, $y \mapsto m(y_1, y_2)$ is a unifier of $m(x, m(y_1, y_2))$ and $m(e, y)$ in the signature of monoids. The critical pairs generated by two left members t and u of rewriting rules can then be obtained by computing the unifiers of t with a subterm of u and vice versa. Graphically, the two terms above can be represented as on the left of (2), and the critical pair, represented on the right of (2), is computed by starting from the term on the left and "extending" it starting from the fact that we want the lower instance of m to be an instance of the m in the term in the middle.



(2)

We refer the reader to [BN99] for a detailed presentation of term rewriting systems along with the classic techniques to prove their convergence.

A nice categorical setting for understanding term rewriting systems was provided by Lawvere in his PhD thesis [Law63]. *Lawvere theories* are one of the main notions of this work: these are cartesian categories, whose objects are integers, and whose cartesian product is given on objects by addition. Every signature induces such a Lawvere theory, with morphisms $f : m \to n$ being n-uples $t_1, \ldots, t_n$ of terms with free variables within $x_1, \ldots, x_m$. Composition $g \circ f : m \to 1$ of a morphism $f = (t_1, \ldots, t_n) : m \to n$ with a morphism $g = (t) : n \to 1$ is given by replacing every variable x_i in t by the term t_i, and this can be extended by product to define the composite of any two morphisms, and identities are $(x_1, \ldots, x_n) : n \to n$. More generally, every equational theory (or term rewriting system) induces a Lawvere theory by quotienting the morphisms of the theory generated by the signature by the equations.

As a particular case, if we consider an equational theory E whose generators are of arity 1, the Lawvere theory it generates is characterized by the monoid M_E of endomorphisms on the object 1 (with composition as multiplication and identities as neutral elements), and one says that the equational theory *presents* a monoid M when M_E is isomorphic to M. For example the monoid $\mathbb{N}/2\mathbb{N}$ is presented by the equational theory with only one generator a of arity 1 and the equation $a(a(x_1)) = x_1$. These presentations of monoids (or of groups) are particularly useful and studied since they can provide finite description of monoids which may be infinite, thus allowing computations on these monoids and a manipulation of them with a computer. More generally, an equational theory E presents a Lawvere theory $\mathcal{C}$ when the category $\mathcal{C}$ is isomorphic to E via a pair of mutually inverse product-preserving functors.

A generalization of presentations to 2-categories (and actually even to ω-categories) can be given by the notion of *polygraph*. These were introduced in their 2-dimensional version by Street [Str76] under the name of computads and later on extended to higher dimensions by Power [Pow90b] and Burroni [Bur93]. They generalize term rewriting systems in the sense that a Lawvere theory being a cartesian category, it can be seen as a particular monoidal category, and therefore as a 2-category with only one 0-cell. In other words, polygraphs can be seen as term rewriting systems improved on the following points:

– the variables of terms are simply typed,

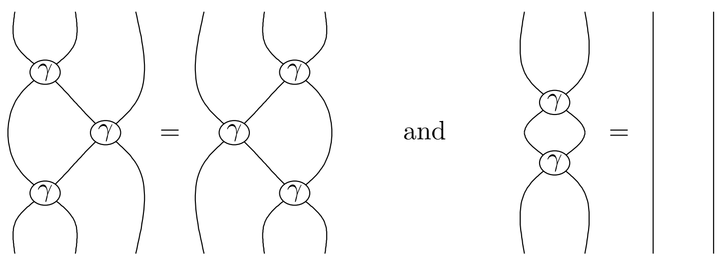
– variables in terms cannot necessarily be duplicated, erased or swapped,

– and the terms can have multiple outputs as well as multiple inputs.

A polygraph essentially consists of typed generators in dimensions 0, 1, 2 and 3, the three first generating a 2-category and the 3-generators expressing equations: a 2-category $\mathcal{C}$ is presented by the polygraph P when the 2-category it freely generates, quotiented by the equations, is isomorphic to $\mathcal{C}$. Many examples of presentations of monoidal categories where studied by Lafont [Laf03], Guiraud [Gui06c, Gui06b] and the author [Mim08, Mim09]. A fundamental example is the 3-polygraph S, presenting the monoidal category **Bij** (the category of finite ordinals and bijections). This polygraph has one generator for objects 1, one generator for morphisms $\gamma : 2 \to 2$ (where 2 is a notation for $1 \otimes 1$) and two equations

$$(\gamma\otimes 1)\circ(1\otimes\gamma)\circ(\gamma\otimes 1) = (1\otimes\gamma)\circ(\gamma\otimes 1)\circ(1\otimes\gamma) \qquad \text{and} \qquad \gamma\circ\gamma = 1\otimes 1 \quad (3)$$

where the morphism 1 is a short notation for id_1. That this polygraph is a presentation of the category **Bij** means that this category is isomorphic to the free monoidal category containing an object 1 and a generator γ, quotiented by the smallest congruence generated by the equations (3). This result can be seen as a generalization of the presentation of the symmetric groups by products of transpositions. The equations can be better understood with the graphical notation provided by *string diagrams*, which is a diagrammatic notation for morphisms in monoidal categories, introduced formally in [JS91]. The morphism γ should be thought as a device with two inputs and two outputs of type 1, and the two equations (3) can thus be represented graphically by



(4)

In this notation, wires represent identities (on the object 1), horizontal juxtaposition of diagrams corresponds to tensoring, and vertical linking of diagrams corresponds to composition of morphisms. Moreover, these diagrams should be considered modulo planar continuous deformations, so that the axioms of monoidal categories are verified. These diagrams are conceptually important because they allows us to see morphisms in monoidal categories (or more generally in 2-categories) either as algebraic objects or as geometric objects. Now, if we orient the two equations (4) from left to right, we get a rewriting system which can be shown to be convergent [Laf03]. It has the three critical pairs given in Figure 1. Moreover, for every morphism $\phi : 1 \otimes m \to 1 \otimes n$, the morphism on the left of Figure 2 can be rewritten in two different ways, thus giving rise to an infinite number of critical pairs for the rewriting system. This phenomenon was first observed by Lafont [Laf03] and later on studied by Guiraud and Malbos [GM09]. Interestingly, we can nevertheless consider that there is a finite number of (families of) critical pairs if we allow ourselves to consider the "diagram" on the center of Figure 2 as a critical pair (or more precisely as a generator for a family of critical pairs). Of course, this diagram does not make sense at first. However, we can give a precise meaning to it if we embed our terms in a larger category, which is compact: in such a category every object has a dual, which corresponds graphically to having the ability to bend wires (see the figure on the right).

There is also another kind of situation that we should handle: critical pairs can have "holes" in them. Namely, consider a polygraph presenting a

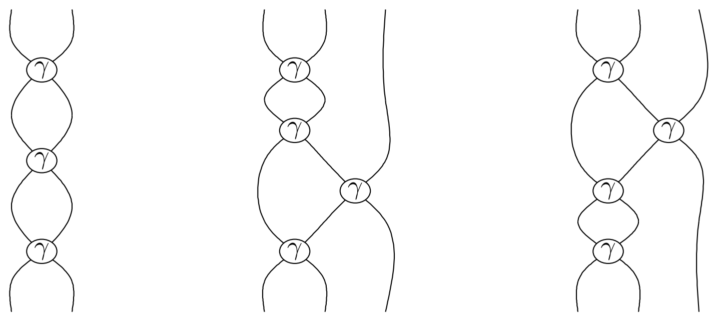


Figure 1: Three critical pairs of the presentation of **Bij**.

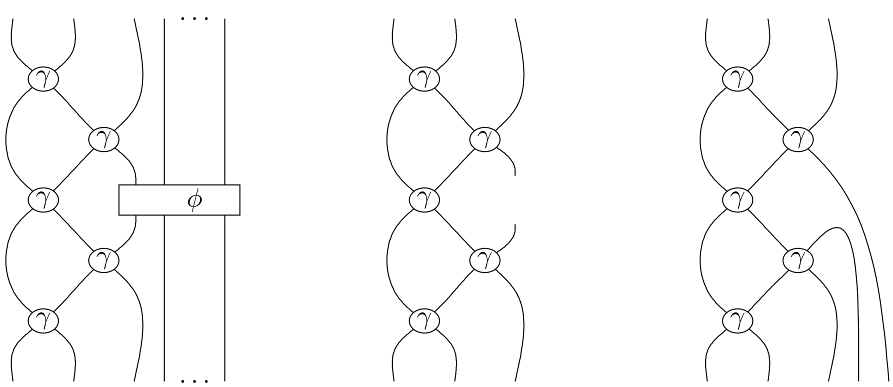


Figure 2: A family of critical pairs of the presentation of **Bij**.

monoidal category with one generator 1 for objects, three generators for morphisms $\delta : 1 \to 3$, $\mu : 3 \to 1$ and $\sigma : 1 \to 1$, and two equations whose left members are pictured on the left of Figure 3. The "morphism" pictured on the right of Figure 3 should be considered as a critical pair generated by the two rules. Again, we need a new theoretical tool in order to make sense of such morphisms containing holes, which is why we will model them as contexts.

These observations were the starting point of this paper which is devoted to formalizing the intuitions explained above, in order to propose an algorithm for computing critical pairs in polygraphs. We believe that this is a major area of higher-dimensional algebra where computer scientists should step in: typical presentations of categories can give rise to a very large number of critical pairs and having automated tools to compute them seems to be necessary in order to push further the study of those systems. The present paper constitutes a first

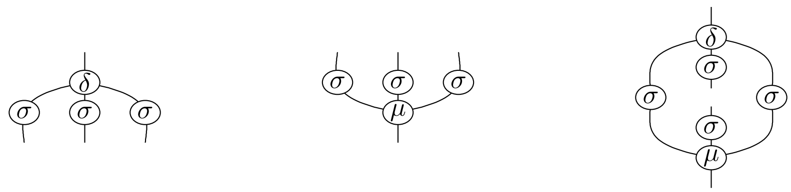


Figure 3: A critical pair containing a hole.

step in this direction, by defining the structures necessary to manipulate algorithmically the morphisms in categories generated by polygraphs, thus allowing us to propose an algorithm to compute the critical pairs in polygraphic rewriting systems. Conversely, algebra provides strong indications about technical choices that should be made in order to generalize rewriting theory in higher dimensions. The framework of polygraphs being very subtle, we deliberately refrained ourselves from being too abstract, because we think that the explicit manipulation of the structures involved is important in order to grasp and understand them. A more general, formal and categorical treatment of the matter should be given in a companion paper, with a uniform handling in higher dimensions of the notions introduced here. We believe that the major contributions of this paper are the representation of morphisms in free 2-categories by nets, the proof that the embedding of a 2-category into the free compact 2-category over it is full and faithful, and the definition of the multicategory of contexts in a 2-category. The unification algorithm for 3-polygraphs, which is the real motivation for introducing this theoretical setting, is sketched in the end of the paper.

1 Category theory recalled

We recall here basic definitions in category theory. A more detailed introduction to category theory can be found in MacLane's reference book [Mac71]. In the following, we will mostly be interested in 2-categories, which is why we only recall the definition of 2-categories for the lack of space. However, the definition of polygraphs is better stated in the general case, and we will we make use of the more general notion of n-category in next section, whose definition can be found in e.g. [Lei04].

2-categories. A *2-category* $\mathcal{C}$ is given by the following data.

- A class $\mathcal{C}_0$ of *0-cells*.
- A category $\mathcal{C}(A, B)$ for every pair of 0-cells A and B. Its objects $f : A \to B$ are called *1-cells*, its morphisms $\alpha : f \Rightarrow g$ are called *2-cells*, composition is

written $\circ$ and called *vertical composition*, and identities are called *vertical identities*.

- A function $\otimes : \mathcal{C}(A,B) \times \mathcal{C}(B,C) \to \mathcal{C}(A,C)$ called *horizontal composition*.
- A 1-cell $\mathrm{id}_A : A \to A$ for every object A called *vertical identity*.

These should be such that the following properties are satisfied.

- Horizontal composition is associative: for every 0-cells A, B, C and D, for every 1-cells $f, f' : A \to B$, $g, g' : B \to C$ and $h, h' : C \to D$, for every 2-cells $\alpha : f \Rightarrow f'$, $\beta : g \Rightarrow g'$ and $\gamma : h \Rightarrow h'$,

  $$(f \otimes g) \otimes h = f \otimes (g \otimes h) \quad (\alpha \otimes \beta) \otimes \gamma = \alpha \otimes (\beta \otimes \gamma) \quad (f' \otimes g') \otimes h' = f' \otimes (g' \otimes h')$$

- Horizontal identities are neutral elements for horizontal composition: for every 0-cells A and B, for every 1-cells $f, f' : A \to B$, for every 2-cell $\alpha : f \Rightarrow f'$,

  $$\mathrm{id}_A \otimes f = f = f \otimes \mathrm{id}_B \quad \mathrm{id}_{\mathrm{id}_A} \otimes \alpha = \alpha = \alpha \otimes \mathrm{id}_{\mathrm{id}_B} \quad \mathrm{id}_A \otimes f' = f' = f' \otimes \mathrm{id}_B$$

We sometimes simply write A for id_A and f for id_f. Two cells are *parallel* if they have the same source and target. This construction can be generalized in any dimension n and we write $\mathbf{Cat}_n$ for the category of n-categories. A strict *monoidal category* is a 2-category with only one 0-cell. All the monoidal categories involved in this paper are implicitly supposed to be strict.

Exchange law. In a 2-category $\mathcal{C}$, for any four 2-cells

$$\begin{array}{rcl} \alpha : f \Rightarrow f' : A \to B & & \beta : g \Rightarrow g' : B \to C \\ & \text{and} & \\ \alpha' : f' \Rightarrow f'' : A \to B & & \beta' : g' \Rightarrow g'' : B \to C \end{array}$$

we have

$$(\beta \otimes \beta') \circ (\alpha \otimes \alpha') \quad = \quad (\beta \circ \alpha) \otimes (\beta' \circ \alpha') \tag{5}$$

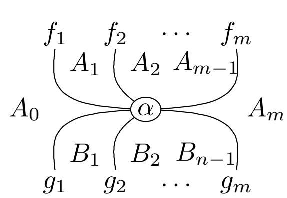
and moreover, for every objects A and B, identities are monoidal natural transformations

$$\mathrm{id}_{A \otimes B} \quad = \quad \mathrm{id}_A \otimes \mathrm{id}_B \tag{6}$$

String diagrams. As explained in the introduction, the morphisms in 2-categories can be represented using *string diagrams*. A 2-cell

$$\alpha \quad : \quad f_1 \otimes \ldots \otimes f_m \quad \Rightarrow \quad g_1 \otimes \ldots \otimes g_n$$

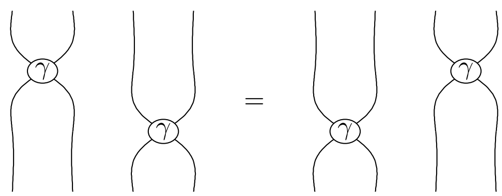
with $f_i : A_{i-1} \to A_i$ and $g_i : B_{i-1} \to B_i$ (and of course $A_0 = B_0$ and $A_m = B_n$) will be represented by a diagram



and bigger diagrams can be constructed from these diagrams by composing them: horizontal composition correspond to juxtaposing diagrams horizontally and vertical composition corresponds to juxtaposing diagrams vertically and linking the wires (see the examples in the introduction). Joyal and Street have shown in details that the category of those diagrams, modulo planar isotopies, is precisely the free 2-category generated by a 2-polygraph. For example, the equality

$$(1 \otimes 1 \otimes \gamma) \circ (\gamma \otimes 1 \otimes 1) \quad = \quad (\gamma \otimes 1 \otimes 1) \circ (1 \otimes 1 \otimes \gamma)$$

in the category $\mathcal{C}$ of the above example, which holds because of the exchange law (5) which is satisfied in any monoidal category, can be shown by continuously deforming the diagram on the left-hand side below into the diagram on the right-hand side:



All the equalities satisfied in any monoidal category generated by a signature have a similar geometrical interpretation. And conversely, any deformation of diagrams corresponds to an equality of morphisms in monoidal categories.

2 Free monoidal 2-categories

2.1 Polygraphs

Polygraphs [Str76, Pow90b, Bur93] were introduced as a way to give finite descriptions of categories, generalizing in particular the notion of presentation of a monoid. We only briefly recall here their construction. The formulation given here is inspired of [GM09].

Graphs. An *n-graph* G is a diagram

$$G_0 \overset{s_0}{\underset{t_0}{\leftleftarrows}} G_1 \overset{s_1}{\underset{t_1}{\leftleftarrows}} G_2 \overset{s_2}{\underset{t_2}{\leftleftarrows}} \ldots \overset{s_{n-1}}{\underset{t_{n-1}}{\leftleftarrows}} G_n$$

in **Set** such that $s_{i-1} \circ s_i = s_{i-1} \circ t_i$ and $t_{i-1} \circ s_i = t_{i-1} \circ t_i$ for every index i such that $0 < i < n$. An element $x \in G_i$ is called an i-generator and $s_{i-1}(x)$ and $t_{i-1}(x)$ are called respectively the source and the target of x. The n-graphs form a category $\mathbf{Grph}_n$ (of presheaves), a morphism between two n-graphs G and G' consisting of a sequence $f_i : G_i \to G'_i$ of functions (with $0 \leq i < n$) such that $s'_{i-1} \circ f_i = f_{i-1} \circ s_{i-1}$ and $t'_{i-1} \circ f_i = f_{i-1} \circ t_{i-1}$.

For every dimension n, there is a forgetful functor $\mathbf{Cat}_n \to \mathbf{Grph}_n$ which to every n-category $\mathcal{C}$ associates the n-graph G such that each G_i is the set of i-cells of $\mathcal{C}$, and for every i-cell $x \in G_i$, $s_{i-1}(x)$ and $t_{i-1}(x)$ are the source and the target of x in $\mathcal{C}$. This functor admits a left adjoint $\mathbf{Grph}_n \to \mathbf{Cat}_n$.

Standard disk and sphere. The *standard n-sphere* is the free n-category on the n-graph S such that for every index i, $S_i = \{x_i^-, x_i^+\}$, $s_{i-1}(x_i^-) = s_{i-1}(x_i^+) = x_{i-1}^-$ and $t_{i-1}(x_i^-) = t_{i-1}(x_i^+) = x_{i-1}^+$. Graphically, S_0, S_1 and S_2 are respectively

$$x_0^- \qquad x_0^+ \qquad\qquad \begin{tikzcd} x_0^- \arrow[r, bend left, "x_1^-"] \arrow[r, bend right, "x_1^+"'] & x_0^+ \end{tikzcd} \qquad\qquad \begin{tikzcd} x_0^- \arrow[r, bend left, "x_1^-"] \arrow[r, bend right, "x_1^+"'] \arrow[r, phantom, "x_2^- \Downarrow\Downarrow x_2^+"] & x_0^+ \end{tikzcd}$$

The *standard n-disk* is the free n-category on the n-graph D whose underlying $(n-1)$-graph is S_{n-1} and such that $D_n = \{x_n\}$ with $s_{n-1}(x_n) = x_{n-1}^-$ and $t_{n-1}(x_n) = x_{n-1}^+$. Graphically, D_0, D_1 and D_2 are respectively

$$x_0 \qquad\qquad \begin{tikzcd} x_0^- \arrow[r, "x_1"] & x_0^+ \end{tikzcd} \qquad\qquad \begin{tikzcd} x_0^- \arrow[r, bend left, "x_1^-"] \arrow[r, bend right, "x_1^+"'] \arrow[r, phantom, "\Downarrow x_2"] & x_0^+ \end{tikzcd}$$

We write $\mathcal{S}_n$ for standard n-sphere and $\mathcal{T}_n$ for the standard n-disk. We also denote by $I_n : \mathcal{S}_n \to \mathcal{T}_{n+1}$ the inclusion functor and by $J_n : \mathcal{S}_n \to \mathcal{T}_n$ the functor such that, for $i < n$ and $\varepsilon \in \{-,+\}$, $J_n(x_i^\varepsilon) = x_i^\varepsilon$ and $J_n(x_n^\varepsilon) = x_n$.

Cellular extension and collapsing. A k-sphere in an n-category $\mathcal{C}$, with $k \leq n$, is a functor $\alpha : \mathcal{S}_k \to \mathcal{C}$. Such a k-sphere α is characterized by the two parallel k-cells $\alpha(x_k^-)$ and $\alpha(x_k^+)$ in $\mathcal{C}$. Similarly, a k-disk in an n-category, with $k \leq n$, is a functor $\alpha : \mathcal{T}_k \to \mathcal{C}$. Such a k-disk α is characterized by the k-cell $\alpha(x_k)$ in $\mathcal{C}$.

Suppose that Γ is a set of k-spheres in $\mathcal{C}$. By coproduct, this set canonically induces an arrow $\Gamma \cdot \mathcal{S}_k \to \mathcal{C}$, that we still write Γ, where $\Gamma \cdot \mathcal{S}_k$ denotes the coproduct $\coprod_{\alpha \in \Gamma} \mathcal{S}_k$ of copies of $\mathcal{S}_k$ indexed by Γ. The *cellular extension* of $\mathcal{C}$ by Γ and the *collapsing of $\mathcal{C}$ by* Γ are the categories $\mathcal{C}[\Gamma]$ and $\mathcal{C}/\Gamma$ respectively defined as the pushouts

$$\begin{tikzcd} \Gamma \cdot \mathcal{S}_k \arrow[r, "\Gamma"] \arrow[d, "\Gamma \cdot I_k"'] & \mathcal{C} \arrow[d] \\ \Gamma \cdot \mathcal{T}_{k+1} \arrow[r] & \mathcal{C}[\Gamma] \end{tikzcd} \qquad \text{and} \qquad \begin{tikzcd} \Gamma \cdot \mathcal{S}_k \arrow[r, "\Gamma"] \arrow[d, "\Gamma \cdot J_k"'] & \mathcal{C} \arrow[d] \\ \Gamma \cdot \mathcal{T}_k \arrow[r] & \mathcal{C}/\Gamma \end{tikzcd} \tag{7}$$

in $\mathbf{Cat}_{n+1}$, where the n-categories involved in the diagrams are seen as $(n+1)$-categories with only identity $(n+1)$-cells. By extension, if Γ is a set of n-spheres in the n-category $\mathcal{C}$, we write $\mathcal{C}[\Gamma]$ and $\mathcal{C}/\Gamma$ for the cellular extension and collapsing of $\mathcal{C}$ seen as an $(n+1)$-category.

Suppose given a functor $F : \mathcal{C} \to \mathcal{D}$ between two n-categories, a set Γ of k-spheres in $\mathcal{C}$ and a functor $\phi : \Gamma \cdot \mathcal{T}_{k+1} \to \mathcal{D}$ (i.e. a set of $(k+1)$-disks in $\mathcal{D}$) such that the diagram

$$\begin{tikzcd} \Gamma \cdot \mathcal{S}_k \arrow[r, "\Gamma"] \arrow[d, "\Gamma \cdot I_k"'] & \mathcal{C} \arrow[d, "F"] \\ \Gamma \cdot \mathcal{T}_{k+1} \arrow[r, "\phi"'] & \mathcal{D} \end{tikzcd}$$

commutes. Then the universal property of the pushout amounts to state that there exists a unique functor $F[\phi] : \mathcal{C}[\Gamma] \to \mathcal{D}$ such that the diagrams

$$\begin{tikzcd} \mathcal{C} \arrow[d] \arrow[dr, "F"] & \\ \mathcal{C}[\Gamma] \arrow[r, "F[\phi]"'] & \mathcal{D} \end{tikzcd} \qquad \text{and} \qquad \begin{tikzcd} \Gamma \cdot \mathcal{T}_{k+1} \arrow[d] \arrow[dr, "\phi"] & \\ \mathcal{C}[\Gamma] \arrow[r, "F[\phi]"'] & \mathcal{D} \end{tikzcd}$$

commute, where the vertical arrows are obtained by the pushout construction.

Polygraphs. Polygraphs formalize the notion of presentation of an n-category $\mathcal{C}$, i.e. the description of $\mathcal{C}$ as a free category quotiented by relations. In order to generate free n-categories, one could start from an n-graph G and use the left adjoint $\mathbf{Grph}_n \to \mathbf{Cat}_n$ to the forgetful functor described above. In such a description, the source and the target of an i-generator $x \in G_i$ would both be $(i-1)$-generators (i.e. elements of G_{i-1}) and not a composite of those. Such a description would necessarily be very redundant and thus not satisfactory. This consideration essentially motivates the introduction of the notion of polygraph.

A 0-*polygraph* P consists of a set P_0 and we write $\mathbf{Pol}_0 = \mathbf{Set}$ for the category of 0-polygraphs. We also write $-^* : \mathbf{Pol}_0 \to \mathbf{Cat}_0$ for the identity function: given a 0-polygraph P, P^* is the set P seen as the free 0-category generated by P.

In higher dimensions, given an integer $n > 0$, the category of n-polygraphs and the free n-category functor are defined by induction as follows.

– An *n-polygraph* $P = (\tilde{P}, P_n)$ consists of an $(n-1)$-polygraph $\tilde{P}$ and a set P_n of n-spheres in P^*, called *n-generators*.

– A morphism $\phi : P \to Q$ between two n-polygraphs consists of a morphism $\tilde{\phi} : \tilde{P} \to \tilde{Q}$ of $(n-1)$-polygraphs and a function $\phi_n : P_n \to Q_n$ such that $\phi_n(\alpha) = \tilde{\phi} \circ \alpha$ for every $\alpha \in P_n$.

– The category of n-polygraphs is denoted by $\mathbf{Pol}_n$.

– The *free n-category* functor $-^* : \mathbf{Pol}_n \to \mathbf{Cat}_n$ is the functor which to every n-polygraph P associates the n-category $P^* = \tilde{P}^*[P_n]$ and to every morphism of n-polygraphs $\phi : P \to Q$ associates the functor $\phi^* : P^* \to Q^*$ defined as $\phi^* = \tilde{\phi}^*[\psi]$, where $\psi : P_n \cdot \mathcal{S}_n \to Q^*$ is the morphism defined as

$$\begin{tikzcd}
P_n \cdot S_n \arrow[r, "\phi \cdot S_n"] & Q_n \cdot \mathcal{S}_n \arrow[r, "Q_n"] \arrow[d, "Q_n \cdot I_n"'] & \tilde{Q}^* \arrow[d] \\
& Q_n \cdot \mathcal{T}_{n+1} \arrow[r] & \tilde{Q}^*[Q_n] = Q^*
\end{tikzcd}$$

the square being a pushout.

It can be shown that the free n-category functor $-^* : \mathbf{Pol}_n \to \mathbf{Cat}_n$ admits a right adjoint $\mathbf{Cat}_n \to \mathbf{Pol}_n$, thus justifying its name.

Given a polygraph $P = (\tilde{P}, P_n)$, we often write P_n^* for the set of n-cells of P^*. A polygraph P is *finite* when all its sets P_i of i-generator are. Given an n-polygraph P and an integer $m \leq n$, we write P/m for the underlying m-polygraph obtained by truncating the polygraph P: $P/n = P$ and $P/(m-1)$ is the underlying $(m-1)$-polygraph of P/m. This operation induces a forgetful functor $\mathbf{Pol}_n \to \mathbf{Pol}_m$ which admits a left adjoint: the *canonical inclusion* $\mathbf{Pol}_m \to \mathbf{Pol}_n$.

The n-category $\overline{P}$ *presented* by an $(n+1)$-polygraph P is defined as the n-category $\overline{P} = \tilde{P}/P_{n+1}$. More generally, an n-category $\mathcal{C}$ is *presented* by P when $\mathcal{C}$ is isomorphic to $\overline{P}$. In this sense, the underlying n-polygraph of a $(n+1)$-polygraph can be thought as a *signature* generating terms which are to be considered modulo the *relations* described by $(n+1)$-generators. In the following, we will be mostly interested in such presentations in dimension $n = 2$.

A different equivalent presentation is given in [Bur93]. For example, a *3-polygraph* consists of a diagram

$$\begin{tikzcd}
P_0 \arrow[d, "i_0"'] & P_1 \arrow[dl, shift right, "s_0"'] \arrow[dl, shift left, "t_0"] \arrow[d, "i_1"] & P_2 \arrow[dl, shift right, "s_1"'] \arrow[dl, shift left, "t_1"] \arrow[d, "i_2"] & P_3 \arrow[dl, shift right, "s_2"'] \arrow[dl, shift left, "t_2"] \\
P_0^* & P_1^* \arrow[l, shift right, "s_0^*"'] \arrow[l, shift left, "t_0^*"] & P_2^* \arrow[l, shift right, "s_1^*"'] \arrow[l, shift left, "t_1^*"] &
\end{tikzcd} \tag{8}$$

in **Set** such that

$$s_i^* \circ s_{i+1} = s_i^* \circ t_{i+1} \quad \text{and} \quad t_i^* \circ s_{i+1} = t_i^* \circ t_{i+1}$$

for $i = 0$ and $i = 1$, together with a structure of 2-category on the 2-graph

$$\begin{tikzcd}
E_0^* & E_1^* \arrow[l, shift right, "s_0^*"'] \arrow[l, shift left, "t_0^*"] & E_2^* \arrow[l, shift right, "s_1^*"'] \arrow[l, shift left, "t_1^*"]
\end{tikzcd}$$

Here, P_i^* is the set of i-cells of the i-category generated by the underlying i-polygraph and the morphisms s_{i-1} and t_{i-1} respectively associate to an i-generator α the $(i-1)$-cells $s_{i-1}(\alpha) = \alpha(x_i^-)$ and $t_{i-1}(\alpha) = \alpha(x_i^+)$ called the *source* and *target* of the generator.

Example 1. The polygraph M corresponding to the theory of monoids has the following generators (we write $f : A \to B$ to indicate that f is a generator whose source is A and target is B, etc.):

$$
\begin{array}{lcl}
E_0 & = & \{*\} \\
E_1 & = & \{1 : * \to *\} \\
E_2 & = & \{\mu : 1 \otimes 1 \Rightarrow 1,\ \eta : * \Rightarrow 1\} \\
E_3 & = & \{a : \mu \circ (\mu \otimes 1) \Rrightarrow \mu \circ (1 \otimes \mu),\ l : \mu \circ (\eta \otimes 1) \Rrightarrow 1,\ r : (1 \otimes \eta) \to 1\}
\end{array}
$$

Graphically, the 3-generators can be pictured as

This 3-polygraph presents the simplicial category Δ (it is a monoidal category and can therefore be seen as a 2-category with only one 0-cell). This category corresponds to the theory of monoids in the sense that the category of monoidal functors and monoidal natural transformations from Δ to a strict monoidal category $\mathcal{C}$ is equivalent to the category of monoids in $\mathcal{C}$.

Example 2. The polygraph S corresponding to the theory of symmetries has the following generators

$$
\begin{array}{lcl}
E_0 & = & \{*\} \\
E_1 & = & \{1 : * \to *\} \\
E_2 & = & \{\gamma : 1 \otimes 1 \Rightarrow 1 \otimes 1\} \\
E_3 & = & \{y : (\gamma \otimes 1) \circ (1 \otimes \gamma) \circ (\gamma \otimes 1) \Rrightarrow (1 \otimes \gamma) \circ (\gamma \otimes 1) \circ (1 \otimes \gamma), \\
 & & \ \ s : \gamma \circ \gamma \Rrightarrow 1 \otimes 1\}
\end{array}
$$

As mentioned in the introduction, this polygraph presents the category **Bij**.

Weight and size. Suppose that P is an n-polygraph. We write $\mathbb{N}$ for the additive monoid $\mathbb{N}$ seen as an n-category with exactly one i-cell 0 for $i < n$, whose set of n-cells is $\mathbb{N}$, whose compositions in all dimensions are given by addition and whose identities are the i-cells 0. The *weight* $w_\alpha : P^* = \tilde{P}^*[P_n] \to \mathbb{N}$ of an n-generator α is the functor defined by the universal property of the pushout

$$
\begin{tikzcd}
P_n \cdot \mathcal{S}_k \arrow[r, "P_n"] \arrow[d, "P_n \cdot I_k"'] & \tilde{P}^* \arrow[d] \arrow[ddr, bend left, "0"] & \\
P_n \cdot \mathcal{T}_{k+1} \arrow[r] \arrow[drr, bend right, "\chi_\alpha"'] & \tilde{P}^*[P_n] \arrow[dr, dotted, "w_\alpha"] & \\
 & & \mathbb{N}
\end{tikzcd}
$$

where 0 denotes the constant functor equal to 0 and χ_α is the functor such that the image of an n-cell $(\beta, x) \in (P_n \cdot \mathcal{T}_{k+1})$ is 1 if $\beta = \alpha$ and 0 otherwise. The *size* $\|x\|$ of an n-cell x in P^* is defined as $\|x\| = \sum_{\alpha \in P_n} w_\alpha(P)$ (the size function can also be defined by replacing χ_α by the constant functor equal to 1 on non-identity $(k+1)$-cells in the pushout diagram above). Given an n-cell φ of P^*, $w_\alpha(\varphi)$ counts the number of occurrences of α in φ and $\|\varphi\|$ is the total number of generators composing φ. We sometimes say that φ *contains* the generator α whenever $w_\alpha(\varphi) > 0$.

2.2 Algebraic construction of a free 2-category

Since our purpose is essentially to manipulate morphisms of the 2-category freely generated by a 2-polygraph, we need a concrete description of this category.

Free categories. Suppose that we are given a 1-polygraph P, i.e. a graph

$$P_0 = P_0^* \underset{t_0^*}{\overset{s_0^*}{\leftleftarrows}} P_1$$

The category generated by this polygraph has the elements A of P_0 as objects and its sets of morphisms are the smallest sets such that

- for every 1-generator $f \in P_1$, such that $s_0(f) = A$ and $t_0(f) = B$, there is a morphism $f : A \to B$,
- for every morphisms $f : A \to B$ and $g : B \to C$ there is a morphism $f \otimes g : A \to C$,
- for every 0-generator $A \in P_0$, there is a morphism $\mathrm{id}_A : A \to A$,

quotiented by the smallest congruence (with respect to composition) imposing that the formal composition is associative and admits the formal identities as neutral elements. Notice that instead of considering formal composites and identities modulo a congruence, we could also have simply constructed morphisms as finite sequences of composable arrows.

Free 2-categories. The 2-category freely generated by a 2-polygraph can be described by a free algebraic construction in a similar fashion. The construction above describes the underlying 1-category and its sets of two-cells are the smallest sets containing the 2-generators and closed under formal horizontal and vertical composition and identities, quotiented by the smallest congruence (with respect to both compositions) such that

- horizontal composition is associative and admits horizontal identities as neutral elements,
- vertical composition is associative and admits vertical identities as neutral elements,

– the exchange laws (5) and (6) between vertical and horizontal composition are satisfied.

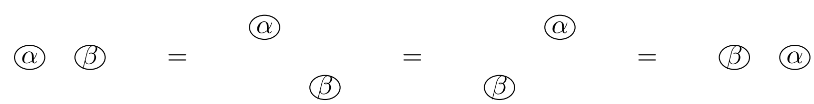
This algebraic construction is however quite difficult to work with, if we want to manipulate morphisms in such 2-categories and effectively decide their equality. For example, suppose that A is a 0-cell and $\alpha : A \Rightarrow A$ and $\beta : A \Rightarrow A$ are two 2-cells in a given 2-category $\mathcal{C}$. The equality

$$\alpha \otimes \beta \quad = \quad \beta \otimes \alpha$$

can be deduced from the following sequence of equalities:

$$\begin{aligned} \alpha \otimes \beta \quad &= \quad (\mathrm{id}_A \circ \alpha) \otimes (\beta \circ \mathrm{id}_A) \quad = \quad (\mathrm{id}_A \otimes \beta) \circ (\alpha \otimes \mathrm{id}_A) \\ &= \quad (\beta \otimes \mathrm{id}_A) \circ (\mathrm{id}_A \otimes \alpha) \quad = \quad (\beta \circ \mathrm{id}_A) \otimes (\mathrm{id}_A \circ \alpha) \quad = \quad \beta \otimes \alpha \end{aligned}$$

which can be pictured graphically by



It requires inserting and removing identities, and using the exchange law in both directions. So, it seems to be very hard to find a generic way to handle formal composites of generators modulo the congruence described above. We will therefore define an alternative construction of these morphisms which doesn't require such a quotienting. The rest of this section is devoted to constructing such a representation.

2.3 Polygraphic nets

The construction of the 2-category generated by a polygraph given in previous section is algebraic but requires to consider morphism modulo a congruence which is difficult to work with. On the other hand, string diagrams are simpler to manipulate but are geometric and thus cannot be directly used for a manipulation of morphisms with a computer. This lead us to introduce a new construction of the 2-category generated by a 2-polygraph using what we call *polygraphic nets* (or *nets* for short), based on polygraphs, which combines the best of both worlds: it is algebraic and does not require working modulo a complex congruence (only isomorphism). We named it this way because it is very close in the spirit to the nets often used to represent logical proofs such as proof-nets [Gir87], interaction nets [Laf90], etc. It is also reminiscent of pasting schemes [Pow90a].

Polygraphic nets are based on the idea that a term (a morphism) generated by a particular signature S is itself an object of the same nature as a signature. For example, consider the 1-polygraph S with

$$S_0 = \{A,\ B\} \quad \text{and} \quad S_1 = \{f : A \to B,\ g : B \to A\} \tag{9}$$

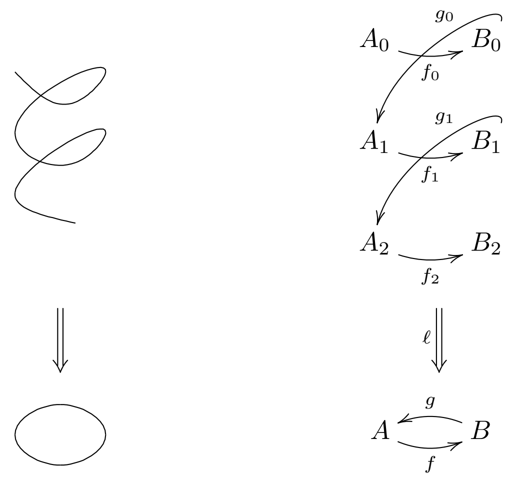


Figure 4: Morphisms are "unfoldings" of the signature.

which can be represented graphically by

$$\begin{tikzcd} A \arrow[r, bend right, "f"'] & B \arrow[l, bend right, "g"'] \end{tikzcd}$$

We will see the term t defined as $f \circ g \circ f \circ g \circ f : A \to A$, which is generated by the previous signature S, as a polygraph P such that

$$P_0 \quad = \quad \{A_0,\ B_0,\ A_1,\ B_1,\ A_2,\ B_2\}$$

and

$$P_1 = \{f_0 : A_0 \to B_0,\ g_0 : B_0 \to A_1,\ f_1 : A_1 \to B_1,\ g_1 : B_1 \to A_2,\ f_2 : A_2 \to B_2\}$$

and such that the A_i, B_i, f_i and g_i are *labeled* by A, B, f and g respectively (formally, these labels are given by a morphism of polygraphs $\ell : P \to S$). This polygraph can be viewed as a particular representation of the term t where each instance of A, B, f and g involved in the definition of this term has been given a distinct "name". It can also be seen as an "unfolding" of the signature S: geometrically, the relation between signatures and the terms they generate is very similar to the relation between spaces and their coverings [Hat02], as illustrated in Figure 4 (the picture on the left represents the circle together with part of its universal covering, which is an infinite spiral, and the figure on the right represents the signature with the term). We will not make explicit use of this parallel with geometry in this paper, but it is nice to keep this picture in mind in order to build intuitions.

We now recast the construction of the free 2-category on a 2-polygraph S given in Section 2.2 as a 2-category $\mathbf{Net}_2^S$ whose k-cells are themselves k-polygraphs. We recall that the categories $\mathbf{Pol}_n$ of n-polygraphs are cocomplete, in particular pushouts always exist.

2.3.1 Polygraphic 0-nets

First, suppose that we are given a 0-polygraph S (i.e. a set S_0). An *atomic* 0-polygraph is a polygraph which contains only one 0-cell. The category $\mathbf{Net}_0^S$ of *polygraphic* 0*-nets* on this polygraph is the subcategory of $\mathbf{Pol}_0 \downarrow S$, whose objects are atomic polygraphs over S and whose morphisms are the isomorphisms. This groupoid should really be thought as an "unicategory", i.e. a "weak set" (just like a bicategory is a weak category).

More explicitly, objects of this category are pairs (x, A), where x is an element of any set with one element and A is an element of S_0 and there is one morphism between two objects (x, A) and (x', A') if and only if $A = A'$. An object (x, A) should be thought as an *instance* of a, where x is the *name* of the instance. The objects of this category form a proper class and not a set (because x can be "anything"). However, in practice, we only need to consider finitely many instances of A at once since we only consider finite polygraphs as rewriting systems, so we can suppose without loss of generality that x is an element of a universe U_0, which is a set at least countable, typically $\mathbb{N}$, and we sometimes write A_i for the pair (i, A) with $i \in \mathbb{N}$.

The category of 0-nets on S is equivalent to the set S_0, seen as a category with only identities.

2.3.2 Polygraphic 1-nets

This construction can be generalized to 1-polygraphs as follows. A 1-polygraph is *atomic* when it has only one 1-cell f and two 0-cells, which are the source and the target 0-cells of the 1-cell, which are distinct. Graphically, an atomic polygraph looks like

$$x_1 \xrightarrow{y} x_2$$

but not like

$$x_1 \xrightarrow{y_1} x_2 \xrightarrow{y_2} x_3 \qquad \text{nor} \qquad \begin{tikzcd} x_1 \arrow[r, bend left, "y_1"] \arrow[r, bend right, "y_2"'] & x_2 \end{tikzcd} \qquad \text{nor} \qquad \begin{tikzcd} x_1 \arrow[loop above, "y_1"] \end{tikzcd}$$

Suppose fixed a 1-polygraph S. The bicategory $\mathbf{Net}_1^S$ of *polygraphic* 1*-nets* on S has the 0-nets M on $S/0$ as objects. The inclusion $\mathbf{Pol}_0 \hookrightarrow \mathbf{Pol}_1$ induces an inclusion functor $\mathbf{Pol}_0 \downarrow (S/0) \hookrightarrow \mathbf{Pol}_1 \downarrow S$, enabling us to see 0-nets on $S/0$ as elements of $\mathbf{Pol}_1 \downarrow S$. The morphisms $N : M_1 \to M_2$ of $\mathbf{Net}_1^S$ are the elements of the smallest set of cospans

$$M_1 \xrightarrow{s} N \xleftarrow{t} M_2 \tag{10}$$

in $\mathbf{Pol}_1 \downarrow S$ such that

- every cospan (10) such that M_1 and M_2 are 0-nets, N is an atomic 1-polygraph with f as unique 1-cell, $s(M_1) = s_0(f)$ and $t(M_2) = t_0(f)$ is a morphism $M_1 \to M_2$,
- for every two morphisms $N_1 : M_1 \to M_2$ and $N_2 : M_2 \to M_3$, the composite morphism $N_2 \circ N_1 : M_1 \to M_3$, defined as the pushout

$$
\begin{tikzcd}
 & & N_2 \circ N_1 & & \\
 & N_1 \arrow[ur, dotted] & & N_2 \arrow[ul, dotted] & \\
M_1 \arrow[ur] & & M_2 \arrow[ul] \arrow[ur] & & M_3 \arrow[ul]
\end{tikzcd}
$$

  is a morphism,
- for every 0-net M, the cospan

$$M \xrightarrow{\mathrm{id}_M} M \xleftarrow{\mathrm{id}_M} M$$

  is the identity morphism on N.

Since composition is defined by a pushout construction, it is not a priori strictly associative, which is why we construct a bicategory (with isomorphisms of polygraphs as 2-cells) which is not necessarily a category.

Example 3. Consider the polygraph S defined in (9) and the polygraphs N_1 and N_2 defined by

$$(N_1)_0 = \{A_0, B_0\} \quad \text{and} \quad (N_1)_1 = \{f_0 : A_0 \to B_0\}$$

and

$$(N_2)_0 = \{B_0, A_0\} \quad \text{and} \quad (N_2)_1 = \{g_0 : B_0 \to A_0\}$$

These polygraphs are elements of $\mathbf{Pol}_1 \downarrow S$ with the obvious labeling morphisms of polygraphs sending A_0, B_0, f_0 and g_0 on A, B, f and g respectively. The composite of $N_1 : A_0 \to B_0$ and $N_2 : B_0 \to A_0$ is (up to isomorphism) the polygraph $N_2 \circ N_1 : A_0 \to A_0$ defined as

$$(N_2 \circ N_1)_0 = \{A_0, B_0, A_1\} \quad \text{and} \quad (N_2 \circ N_1)_1 = \{f_0 : A_0 \to B_0, g_0 : B_0 \to A_1\}$$

Graphically, this can be represented as

$$A_0 \xrightarrow{f_0} B_0 \quad \otimes \quad B_0 \xrightarrow{g_0} A_0 \quad = \quad A_0 \xrightarrow{f_0} B_0 \xrightarrow{g_0} A_1$$

Notice that we have to name differently the two instances of A in the composite, be the choice of their names is of course arbitrary.

Given a 1-polygraph, we define the *horizontal ordering* relation $<_0$ between 1-cells as the smallest transitive relation such that $f <_0 g$ whenever $t_0(f) = s_0(g)$. Polygraphic 1-nets can be characterized with this relation as follows.

Property 4. The 1-cells $N : M_1 \to M_2$ of the bicategory $\mathbf{Net}_1^S$ are precisely the cospans $M_1 \xleftarrow{s} M \xrightarrow{t} M_2$ in $\mathbf{Pol}_1 \!\downarrow\! S$ which are

1. *linear*: a 0-generator $x \in N_0$ is the source (resp. the target) of exactly one 1-generator $y \in N_1$, excepting for $t(M_2)$ (resp. $s(M_1)$) which is the source (resp. the target) of none,
2. *acyclic*: the relation $<_0$ is irreflexive.

Such a polygraph M is the same as a linear graph

$$x_0 \xrightarrow{y_1} x_1 \xrightarrow{y_2} x_2 \cdots x_{n-1} \xrightarrow{y_n} x_n \tag{11}$$

where $x_i \in E_0$ and $y_i \in E_1$ such that for every index i, $\ell(s_0(y_{i+1})) = \ell(x_i)$ and $\ell(t_0(y_{i+1})) = \ell(x_{i+1})$, where $\ell : M \to S$ denotes the labeling functor. We thus recover the usual construction of the free category on a graph as the category of paths on this graph:

Property 5. The category on a signature S, obtained from the bicategory $\mathbf{Net}_1^S$ by quotienting objects and morphisms by isomorphism of polygraphs, is isomorphic to the free category generated by the polygraph S.

Remark 6. Suppose that S is a polygraph with one 0-generator A and two 1-generators $f, g : A \to A$ and consider the polygraphs P and Q in $\mathbf{Pol}_1 \!\downarrow\! S$ such that

$$P_0 = \{A_0\} \qquad P_1 = \{f_0 : A_0 \to A_0\} \qquad Q_0 = \{A_0\} \qquad Q_1 = \{g_0 : A_0 \to A_0\}$$

Graphically, P and Q can be respectively pictured as

$$\begin{tikzcd} A_0 \arrow[loop above, "f_0"] \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} A_0 \arrow[loop above, "g_0"] \end{tikzcd}$$

Notice that by Property 4, these polygraphs are not morphisms in $\mathbf{Net}_1^S$ (with obvious source and target). If it was the case, then this bicategory would contain two morphisms such that $Q \circ P \cong P \circ Q$, which can be both pictured as

$$\begin{tikzcd} A_0 \arrow[loop left, "f_0"] \arrow[loop right, "g_0"] \end{tikzcd}$$

thus failing to be isomorphic (when quotiented by isomorphism of polygraphs) to the free category generated by S. This explains why we need to take care of which *instance* of a generator of S is used in a polygraph (we see A_0 as an instance of A).

2.3.3 Polygraphic 2-nets

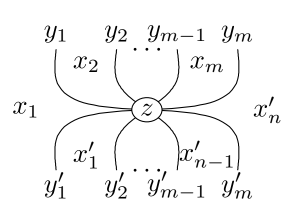
A 2-polygraph P is *atomic* when

- it contains only one 2-generator z,
- the source and target $s_1(z)$ and $t_1(z)$ are 1-nets,
- $P_1 = (s_1(z))_1 \cup (t_1(z))_1$ and $(s_1(z))_1 \cap (t_1(z))_1 = \emptyset$,
- $P_0 = (s_1(z))_0 \cup (t_1(z))_0$ and $(s_1(z))_0 \cap (s_1(z))_0 = \{s_0(s_1(z)), t_0(t_1(z))\}$.

Graphically, an atomic 2-polygraph looks like

$$\begin{tikzcd}
 & x_2 \arrow[r, "y_2"] & x_3 \ldots x_{m-1} \arrow[r, "y_{m-1}"] & x_m \arrow[dr, "y_m"] & \\
x_1 \arrow[ur, "y_1"] \arrow[dr, "y'_1"'] & & z \Downarrow & & x'_n \\
 & x'_1 \arrow[r, "y'_2"'] & x'_2 \ldots x'_{n-2} \arrow[r, "y'_{n-1}"'] & x'_{n-1} \arrow[ur, "y'_n"'] &
\end{tikzcd}$$

or



in diagrammatic and in string-diagrammatic notations, where the y_i and y'_i are all distinct and the x_i and x'_i are all distinct.

Suppose that we are given a 2-polygraph S. The weak 2-category $\mathbf{Net}_2^S$ of 2-nets on S is defined by a generalization of the previous construction. Its underlying category is the category $\mathbf{Net}_1^{S/1}$ of 1-nets on the polygraph $S/1$. Again, such 1-nets can be seen as objects in the 2-category $\mathbf{Pol}_2{\downarrow}S$. The 2-cells $P : N_1 \Rightarrow N_2 : M_1 \to M_2$ of $\mathbf{Net}_2^S$ will be cospans

$$N_1 \xrightarrow{s} P \xleftarrow{t} N_2 \tag{12}$$

in the category $\mathbf{Pol}_2{\downarrow}S$ such that N_1 and N_2 are both 1-nets from M_1 to M_2, thus inducing diagrams of the form

$$\begin{tikzcd}
 & N_1 \arrow[d] & \\
M_1 \arrow[ur] \arrow[dr] & P & M_2 \arrow[ul] \arrow[dl] \\
 & N_2 \arrow[u] &
\end{tikzcd}$$

Vertical composition of two morphisms

$$P : N_1 \Rightarrow N_2 : M_1 \to M_2 \quad \text{and} \quad Q : N_2 \Rightarrow N_3 : M_1 \to M_2$$

is given by the pushout of consecutive cospans of the form (12) as shown in the left of Figure 5 and composition $P \otimes Q$ of two morphisms

$$P : N_1 \Rightarrow N_2 : M_1 \to M_2 \quad \text{and} \quad Q : N_3 \Rightarrow N_4 : M_2 \to M_3$$

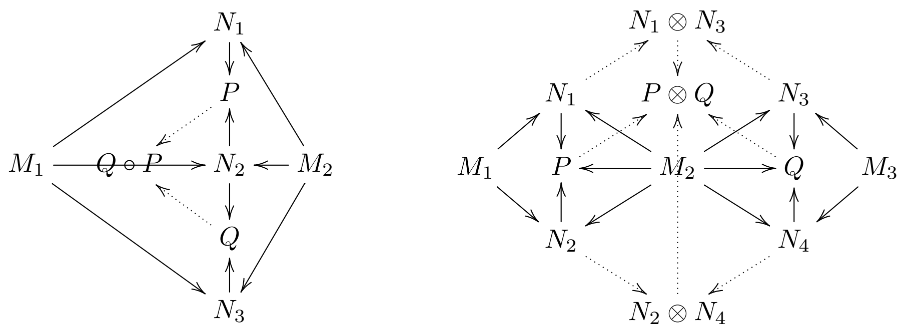
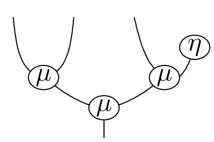


Figure 5: Composition by pushouts.

is given by the sequence of pushouts shown in the right of Figure 5 (horizontal arrows are obtained by composition and vertical dotted arrows are obtained by the universal property of the pushouts). We define the set of 2-cells of the 2-category $\mathbf{Net}_2^S$ as the smallest set of 2-cells containing atomic 2-polygraphs over S and moreover closed under both vertical and horizontal composition and identities. Since $\mathbf{Net}_1^{S/1}$ is a bicategory and composition of spans is not strictly associative, we have defined a weak 2-category. However, in the following we will consider 2-nets up to isomorphism (which corresponds to injective renaming of cells) and these form a (strict) 2-category.

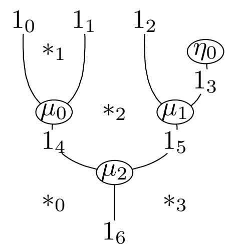
Example 7. The morphism $\mu \circ (\mu \otimes (\mu \circ (1 \otimes \eta)))$ in the theory of monoids (see Example 1) whose string-diagrammatic notation is



can be represented by the polygraph M whose generators are

$$\begin{array}{rcl} M_0 & = & \{*_0,\ *_1,\ *_2,\ *_3\} \\ M_1 & = & \{1_0 : *_0 \Rightarrow *_1,\ 1_1 : *_1 \Rightarrow *_2,\ 1_2 : *_2 \Rightarrow *_3, \\ & & \ 1_3 : *_3 \Rightarrow *_3,\ 1_4 : *_0 \Rightarrow *_2,\ 1_5 : *_2 \Rightarrow *_3,\ 1_6 : *_0 \Rightarrow *_3\} \\ M_2 & = & \{\eta_0 : *_3 \Rrightarrow 1_3,\ \mu_0 : 1_0 \otimes 1_1 \Rrightarrow 1_4, \\ & & \ \mu_1 : 1_2 \otimes 1_3 \Rrightarrow 1_5,\ \mu_2 : 1_4 \otimes 1_5 \Rrightarrow 1_6\} \end{array}$$

Graphically, this corresponds to giving a different label to each instance of generator of the signature occurring in the morphism:



(13)

There is no obvious canonical choice for those labels in the general case, which explains why we have to consider nets modulo isomorphism (i.e. injective renaming of those labels).

Since the definition of the category $\mathbf{Net}_2^S$ is given as generated by suitable pushouts of nets, it can easily be shown that

Theorem 8. The 2-category $\mathbf{Net}_2^S$ on a signature S is equivalent to the free 2-category generated by the polygraph S.

Proof. Given a 2-generator $\alpha \in S_2$, there exists, up to isomorphism of nets, exactly one atomic net A_α whose source and target are respectively $\alpha(x_1^-)$ and $\alpha(x_1^+)$ and whose 2-generator is labeled by α. From this, we can construct maps making the diagram

$$\begin{tikzcd} S_2 \cdot \mathcal{S}_1 \arrow[r, "S_2"] \arrow[d, "S_2 \cdot I_1"'] & \mathbf{Net}_1^{S/1} \arrow[d, "v"] \\ S_2 \cdot \mathcal{T}_2 \arrow[r, "h"'] & \mathbf{Net}_2^S \end{tikzcd}$$

commute in $\mathbf{Cat}_2$: the vertical v arrow sends a 2-generator $\alpha \in S_2$ to A_α (it is the inclusion of the source and target 1-sphere of A_α into A_α) and the horizontal arrow h sends a 2-generator $\alpha \in S_2$ to A_α seen as a 2-disk in $\mathbf{Net}_2^S$. Now, suppose that $\mathcal{D}$ is a 2-category, $F : \mathbf{Net}_1^{S/1} \to \mathcal{D}$ is a 2-functor, and $\phi : S_2 \cdot \mathcal{T}_2 \to \mathcal{D}$ is a family of 2-disks in $\mathcal{D}$ such that the diagram

$$\begin{tikzcd} S_2 \cdot \mathcal{S}_1 \arrow[r, "S_2"] \arrow[d, "S_2 \cdot I_1"'] & \mathbf{Net}_1^{S/1} \arrow[d, "F"] \\ S_2 \cdot \mathcal{T}_2 \arrow[r, "\phi"'] & \mathcal{D} \end{tikzcd}$$

commutes in $\mathbf{Cat}_2$. Because of the inductive definition of $\mathbf{Net}_2^S$, there exists an unique 2-functor $f : \mathbf{Net}_2^S \to \mathcal{D}$ such that for every 2-generator α, we have $f(A_\alpha) = \phi_\alpha(x_2)$. This functor satisfies $f \circ v = F$ and $f \circ h = \phi$, and moreover it is the only one satisfying these equalities because of the property mentioned at the beginning of the proof. $\square$

Property 9. The *vertical ordering* relation $<_1$ is the smallest transitive relation on 2-generators P_2 of a 2-polygraph P such that $z <_1 z'$ whenever $(t_1(z))_1 \cap (s_1(z'))_1 \neq \emptyset$. The 2-polygraphs

$$P : N \Rightarrow N' : M \to M'$$

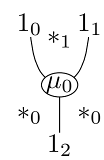
in $\mathbf{Net}_2^S$ are

- *linear*: a 1-generator $y \in P_1$ is in the source (resp. in the target) of exactly one 2-generator $z \in P_2$, i.e. $y \in (s_1(z))_1$ (resp. $y \in (t_1(z))_1$), excepting for the elements of $(t(P))_0$ (resp. $(s(P))_0$) which are in the source (resp. target) of none,
- *acyclic*: the relation $<_1$ is irreflexive.

Remark 10. In Example 7, the cell $1_3 : *_3 \to *_3$ has the same source and target, showing that 2-nets are not acyclic in the sense of Property 4 (i.e. the horizontal ordering $<_0$ is not irreflexive).

Remark 11. Property 9 does not give a characterization of 2-polygraphs which are 1-nets. For example, the polygraph

$$\begin{array}{lcl} E_0 & = & \{*_0,\ *_1\} \\ E_1 & = & \{1_0 : *_0 \to *_1,\ 1_1 : *_1 \to *_0,\ 1_2 : *_0 \to *_0\} \\ E_2 & = & \{\mu_0 : 1_0 \otimes 1_1 \to_2\} \end{array}$$



is not a 2-net over the signature of monoids M, intuitively because two distinct portion of the plane have been given the same name $*_0$. It seems difficult to give a direct characterization of 2-nets amongst 2-polygraphs.

3 Confluence for 3-polygraphs

3.1 The multicategory of contexts

We introduce here the notion of *context* in a 2-category. These contexts should be thought as morphisms in which typed variables (or holes) occur. Since we consider contexts which can have multiple holes those are naturally structured as a multicategory (also sometimes called "colored operad"). A detailed introduction to multicategories can be found in [Lei04], we only recall the definition here.

Definition 12 (Multicategory)**.** A *multicategory* $\mathcal{M}$ is given by

- a class $\mathcal{M}_0$ of *objects*,
- a class $\mathcal{M}_1(A_1, \ldots, A_n; A)$ of *operations* for every objects $A_1, \ldots, A_n$ and A, we write $f : A_1, \ldots, A_n \to A$ to indicate that $f \in \mathcal{M}_1(A_1, \ldots, A_n; A)$,

- a *composition* function which to every operations $f_i : A_i^1, \dots, A_i^{k_i} \to A_i$, for $1 \leq i \leq n$, and $f : A_1, \dots, A_n \to A$, associates a composite operation

$$f \circ (f_1, \dots, f_n) \quad : \quad A_1^1, \dots, A_1^{k_1}, \dots, A_n^1, \dots, A_n^{k_n} \to A$$

  that we often simply write $f(f_1, \dots, f_n)$,

- an operation $\mathrm{id}_A : A \to A$, called *identity*, for every object A,

such that

- the composition is associative:

$$\begin{aligned} & f \circ \left(f_1 \circ (f_1^1, \dots, f_1^{k_1}), \dots, f_n \circ (f_n^1, \dots, f_n^{k_n})\right) \\ = \quad & (f \circ (f_1, \dots, f_n)) \circ (f_1, \dots, f_1^{k_1}, \dots, f_n^1, \dots, f_n^{k_n}) \end{aligned}$$

  for every operations f, f_i and f_i^j for which compositions make sense,

- the composition admits identities as neutral elements: for every operation $f : A_1, \dots, A_n \to A$, we have $f \circ (\mathrm{id}_A, \dots, \mathrm{id}_A) = f$.

A *symmetric multicategory* is a multicategory $\mathcal{M}$ together with a bijection between $\mathcal{M}(A_1, \dots, A_n; A)$ and $\mathcal{M}(A_{\sigma(1)}, \dots, A_{\sigma(n)}; A)$, for every permutation $\sigma : n \to n$, satisfying coherence axioms.

The multicategory of contexts of a 2-category is defined as follows:

Definition 13 (Multicategory of contexts of a 2-category)**.** Suppose that $\mathcal{C}$ is a 2-category. The symmetric multicategory of contexts of $\mathcal{C}$, written $\mathcal{K}_\mathcal{C}$ is the multicategory defined as follows. Its objects pairs of parallel 1-cells $f, g : A \to B$ of $\mathcal{C}$, written $f \Rightarrow g : A \to B$ or simply $f \Rightarrow g$ when there is no ambiguity on their source and target. Its morphisms

$$K \quad : \quad f_1 \Rightarrow g_1, \dots, f_n \Rightarrow g_n \quad \Rrightarrow \quad f \Rightarrow g$$

are the morphisms $K : f \to g$ in $\mathcal{C}[f_1 \Rightarrow g_1, \dots, f_n \Rightarrow g_n]$ such that for every index i, $w_{f_i \Rightarrow g_i}(K) = 1$ (i.e. variables occur exactly once) and composition is induced by substitution in the obvious way. Notice that an object of $\mathcal{K}_\mathcal{C}$ can be considered as a 1-sphere in $\mathcal{C}$, so in the notation above we identify $f_i \Rightarrow g_i$ with the corresponding sphere.

When we restrict to unary morphisms, we get a category of contexts which acts on the 2-category $\mathcal{C}$. If $K : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g$ is a unary context and $\alpha : f_1 \Rightarrow g_1$ is a 2-cell of $\mathcal{C}$, we write $K(\alpha) : f \Rightarrow g$ for the corresponding morphism: the context K is a morphism $K : f \to g$ in $\mathcal{C}[f_1 \Rightarrow g_1]$ and $K(\alpha)$ denotes $K[\phi]$ where $\phi : \mathcal{T}_2 \to \mathcal{C}$ is the functor which sends the 2-cell x_2 of standard 2-disk to α. Similarly, if $\mathcal{C}$ is the underlying 2-category of a 3-category $\mathcal{D}$, $K : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g$, and $r : \alpha \Rrightarrow \beta : f_1 \Rightarrow g_1$ is a 3-cell in $\mathcal{D}$, we write $K(r) : K\alpha \Rrightarrow K\beta : f \to g$ for the obvious 3-cell of $\mathcal{D}$.

Remark 14. Given a 2-polygraph P, a generator α occurs in a 2-cell β of P^* (i.e. $w_\alpha(\beta) > 0$) if and only if there exists a unary context K of P^* such that $\beta = K(\alpha)$.

Remark 15. Given a 2-polygraph P, a 2-cell $\alpha : f_1 \Rightarrow g_1$ and a 2-cell $\beta : f \Rightarrow g$, there exists a unary context $K : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g$ such that $\beta = K(\alpha)$ if and only if there exists a morphism $i : \alpha \to \beta$ of polygraphs over P, where α and β are seen as 2-nets in $\mathbf{Net}_2^P$.

3.2 Critical pairs

A 3-polygraph S freely generates a 3-category S^*. Two coinitial 3-cells

$$r_1 : \alpha \Rrightarrow \beta_1 : f \Rightarrow g : A \to B \qquad \text{and} \qquad r_2 : \alpha \Rrightarrow \beta_2 : f \Rightarrow g : A \to B$$

of this 3-category are *joinable* when there exists a 2-cell $\beta : f \Rightarrow g$ and two 3-cells $s_1 : \beta_1 \Rrightarrow \beta$ and $s_2 : \beta_2 \Rrightarrow \beta$ such that $s_1 \circ_2 r_1 = s_2 \circ_2 r_2$ (where $\circ_2$ denotes the composition in dimension 2). Given a 3-generator $r : \alpha \Rrightarrow \beta$ and 2-cells α' and β', we write $\alpha \Rrightarrow^{K,r} \beta$ (or sometimes simply $\alpha \Rrightarrow^{r} \beta$), when there exists a unary context K such that $K(\alpha) = \alpha'$ and $K(\beta) = \beta'$. A polygraph is *locally confluent* when for every cells such that $\alpha \Rrightarrow^{K_1,r_1} \beta_1$ and $\alpha \Rrightarrow^{K_2,r_2} \beta_2$, the two 3-cells $K_1(r_1)$ and $K_2(r_2)$ are joinable. It is *terminating* when there is no infinite sequence $\alpha_1 \Rrightarrow^{K_1,r_1} \alpha_2 \Rrightarrow^{K_2,r_2} \ldots$.

The Newman's lemma is still valid in this framework [GM09]:

Lemma 16. A terminating polygraph is confluent if and only if it is locally confluent.

We have seen in Section 2.3.3, that the 2-cells of the 3-category generated by a 3-polygraph S can be seen as polygraphic 2-nets. In some simple cases, termination of polygraphs can be deduced from the following lemma:

Lemma 17. A 3-polygraph S such that for every 3-generator $r : \alpha \Rrightarrow \beta$ we have $\|\alpha\| > \|\beta\|$ is terminating.

This simple criterion for showing the termination of a polygraph is often too weak. More elaborate termination orders for 3-polygraphs have been studied by Guiraud [Gui06a]. In this paper, we are mostly interested in studying local confluence of polygraphs.

The usual notion of critical pair can be extended to the setting of 3-polygraphs as follows.

Definition 18 (Unifier)**.** A *unifier* of a two 2-cells

$$\alpha_1 : f_1 \Rightarrow g_1 \qquad \text{and} \qquad \alpha_2 : f_2 \Rightarrow g_2$$

in a 2-category $\mathcal{C}$ is a pair of cofinal unary contexts

$$K_1 : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g \qquad \text{and} \qquad K_2 : f_2 \Rightarrow g_2 \Rrightarrow f \Rightarrow g$$

such that $K_1(\alpha_1) = K_2(\alpha_2)$. A unifier is a *most general unifier* when it is

– *non-trivial*: there is no binary context

$$K : f_1 \Rightarrow g_1, f_2 \Rightarrow g_2 \quad \Rrightarrow \quad f \Rightarrow h$$

such that

$$K_1 = K \circ (\mathrm{id}_{f_1 \Rightarrow g_1}, \alpha_2) \quad \text{and} \quad K_2 = K \circ (\alpha_1, \mathrm{id}_{f_1 \Rightarrow g_1})$$

– *minimal*: for every unifier (K_1', K_2') of α_1 and α_2 such that $K_1 = K_1'' \circ K_1'$ and $K_2 = K_2'' \circ K_2'$ for some contexts K_1'' and K_2'', the unary contexts K_1'' and K_2'' are invertible.

Definition 19 (Critical pair)**.** A *critical pair* (K_1, r_1, K_2, r_2) in a 3-polygraph S consists of two 3-generators

$$r_1 : \alpha_1 \Rrightarrow \beta_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad r_2 : \alpha_2 \Rrightarrow \beta_2 : f_2 \Rightarrow g_2$$

and a unifier

$$K_1 : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g \quad \text{and} \quad K_2 : f_2 \Rightarrow g_2 \Rrightarrow f \Rightarrow g$$

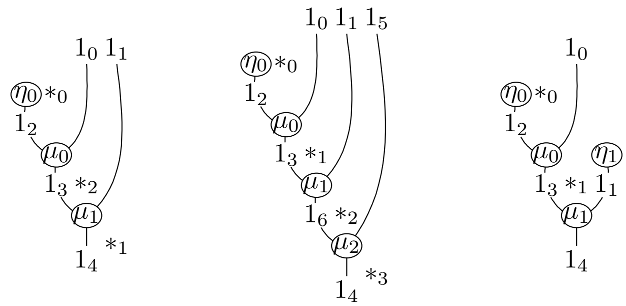
of α_1 and α_2. We sometimes say that the 2-cell $\alpha = K_1(\alpha) = K_2(\alpha_2)$ is a critical pair, by abuse of language.

We have seen in Section 2.3.3 that the morphisms of the free 2-category generated by a polygraph S can be seen as 2-nets, i.e. (cospans of) 2-polygraphs over S, allowing us to consider morphisms of polygraphs between nets over S. By Remark 15, a 2-nets β of the form $\beta = K(\alpha)$ can be seen as a 2-net β together with a morphism $i : \alpha \Rrightarrow \beta$ of 2-polygraphs over S. This allows us to characterize concretely some critical pairs in nets. Namely, suppose that we are given two 2-nets $\alpha_1 : f_1 \Rightarrow g_1$ and $\alpha_2 : f_2 \Rightarrow g_2$ in $\mathbf{Net}_2^S$. An *unifier* of these two nets can be equivalently defined as a 2-net $\alpha : f \Rightarrow g$ in $\mathbf{Net}_2^S$ together with two morphisms $i_1 : \alpha_1 \to \alpha$ and $i_2 : \alpha_2 \to \alpha$ of 2-polygraphs over S. This unifier is non-trivial in the sense of Definition 19 if and only if the 2-net α' defined as the pullback

$$\begin{tikzcd}
 & \alpha' \arrow[dl, dotted] \arrow[dr, dotted] & \\
\alpha_1 \arrow[dr, "i_1"'] & & \alpha_2 \arrow[dl, "i_2"] \\
 & \alpha &
\end{tikzcd}$$

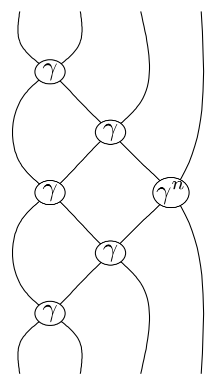
contains at least one 2-generator. Moreover, if we write $+$ for the coproduct of polygraphs, when the morphism $i_1 + i_2 : \alpha_1 + \alpha_2 \to \alpha$ is epi, the unifier is also minimal – but a critical pair does not necessarily satisfy this condition, for example it is not the case for the critical pairs of Example (21).

Example 20. Consider the theory of monoids given in Example 1. The net on the left of



is a unifier of the rules a and l. However the two nets on the right are not because they are respectively trivial and not minimal.

Example 21. Consider the 3-polygraph S of symmetries (Example 2). We write $\gamma^n : 1 \otimes 1 \to 1 \otimes 1$ for the morphism defined by induction on the integer n by $\gamma^0 = 1 \otimes 1$ and $\gamma^{n+1} = \gamma \circ \gamma^n$. Then for every integer n, the morphism



is a critical pair, since the source of the 3-generator y appears on the upper-left part and on the lower-left part of the morphism, and both share one 2-generator γ.

The usual property of critical pairs extends to our framework [GM09]:

Property 22. A 3-polygraph S is locally confluent if and only if for each of its critical pair (K_1, r_1, K_2, r_2), the 3-cells $K_1(r_1)$ and $K_2(r_2)$ are joinable.

4 Free compact 2-categories

Example 21 shows that a finite 3-polygraph can give rise to an infinite number of critical pairs. As explained in the introduction, this is not the case anymore if we allow ourselves to consider diagrams such as the one depicted on the center of Figure 2. We formalize these kind of intuitive diagrams by formally adding adjoints to the 2-categories generated by 2-polygraphs. Graphically, this corresponds to adding the possibility of "bending" wires.

4.1 Compact 2-categories

The notion of adjunction can be formalized between 1-cells in a 2-category as follows [KS72], generalizing the situation in **Cat**.

Definition 23 (Adjoint)**.** Given a 2-category $\mathcal{C}$, a 1-cell $f : A \to B$ is *left adjoint* to a 1-cell $g : B \to A$, what we write

$$\begin{tikzcd} A \arrow[r, bend left, "f"] \arrow[r, phantom, "\perp"] & B \arrow[l, bend left, "g"] \end{tikzcd}$$

when there exists two 2-cells $\eta : A \to f \otimes g$ and $\varepsilon : g \otimes f \to B$ such that

$$(f \otimes \varepsilon) \circ (\eta \otimes f) = f \qquad \text{and} \qquad (\varepsilon \otimes g) \circ (g \otimes \eta) = g$$

The 1-cell g is then said to be *right adjoint* to f.

The notion of 2-category with adjoints was studied in the case of symmetric monoidal categories [KL80] (where they are called *compact closed* categories), monoidal categories [JS93] (where they are called *autonomous* categories), as well as other variants such as *spherical* categories [BW99]; see [Sel08] for a concise presentation of those.

Definition 24 (Compact 2-category)**.** A 2-category is *compact* when every 1-cell admits both a left and a right adjoint.

A *strictly compact* 2-category is a compact 2-category in which every 1-cell $f : A \to B$ has an assigned left adjoint $f^{-1} : B \to A$ and an assigned right adjoint $f^{+1} : B \to A$. We write η^+_f and ε^+_f (resp. η^-_f and ε^-_f) for the unit and the counit of the adjunction $f \dashv f^{+1}$ (resp. $f^{-1} \dashv f$). The following coherence axioms should moreover be satisfied:

- for every pair of composable 1-cells f and g,

  $$(f \otimes g)^{-1} = g^{-1} \otimes f^{-1} \qquad \text{and} \qquad (f \otimes g)^{+1} = g^{+1} \otimes f^{+1}$$

  and

  $$\eta^+_{f \otimes g} = (f \otimes \eta^+_g \otimes f^+) \circ \eta^+_f \qquad \text{and} \qquad \varepsilon^+_{f \otimes g} = \varepsilon^+_f \circ (f^{+1} \otimes \varepsilon^+_g \otimes f)$$

  and

  $$\eta^-_{f \otimes g} = (f^{-1} \otimes \eta^-_g \otimes f) \circ \eta^-_f \qquad \text{and} \qquad \varepsilon^-_{f \otimes g} = \varepsilon^-_f \circ (f \otimes \varepsilon^-_g \otimes f^{-1})$$

– for every 0-cell A,

$$\mathrm{id}_A^{-1} = \mathrm{id}_A = \mathrm{id}_A^{+1}$$

and

$$\eta^+_{\mathrm{id}_A} = \mathrm{id}_A = \varepsilon^+_{\mathrm{id}_A} \quad \text{and} \quad \eta^-_{\mathrm{id}_A} = \mathrm{id}_A = \varepsilon^-_{\mathrm{id}_A}$$

– for every 1-cell f,

$$(f^{+1})^{-1} = f = (f^{-1})^{+1}$$

and

$$\eta^+_{f^{-1}} = \eta^-_f \quad \text{and} \quad \varepsilon^+_{f^{-1}} = \varepsilon^-_f$$

and

$$\eta^-_{f^{+1}} = \eta^+_f \quad \text{and} \quad \varepsilon^-_{f^{+1}} = \varepsilon^+_f$$

For any 1-cell $f : A \to B$ in a strictly compact 2-category and integer n, the morphism f^n denotes the morphism defined by $f^0 = f$, $f^{n+1} = (f^n)^{+1}$ and $f^{n-1} = (f^n)^{-1}$. We also simply write $\eta_f : B \Rightarrow f^{-1} \otimes f$ and $\varepsilon_f : f \otimes f^{-1} \Rightarrow A$ for the unit and the counit of the adjunction between f^{-1} and f.

In the following, we suppose for simplicity that all the compact categories we consider are equipped with a structure of strictly compact category. This is not restrictive since every compact 2-category can be shown to be equivalent to a strict one using an argument similar to the coherence theorem for compact closed categories [KL80].

4.2 Embedding 2-categories into compact 2-categories

There is an obvious forgetful functor from the category of compact 2-categories to the category of 2-categories, and this forgetful functor admits a left adjoint. We write $\mathcal{A}(\mathcal{C})$ for the free compact 2-category on a 2-category $\mathcal{C}$ (the $\mathcal{A}$ here stands for "adjoints"). The construction of this free 2-category is detailed in [PL07] and consists essentially in adapting the work of Kelly and Laplaza on compact closed categories [KL80] to monoidal categories which are not supposed to be symmetric. We recall briefly this construction here.

Every compact 2-category has an underlying category with formal adjoints in the following sense:

Definition 25 (Category with formal adjoints)**.** A category with formal adjoints $(\mathcal{C}, (-)^{-1}, (-)^{+1})$ is a category together with two functors

$$(-)^{-1} : \mathcal{C} \to \mathcal{C}^{\mathrm{op}} \quad \text{and} \quad (-)^{+1} : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$$

such that $((-)^{-1})^{+1} = \mathrm{id}_{\mathcal{C}}$ and $((-)^{+1})^{-1} = \mathrm{id}_{\mathcal{C}^{\mathrm{op}}}$.

Given a 2-category $\mathcal{C}$ with underlying category $\mathcal{C}/1$, the underlying category of $\mathcal{A}(\mathcal{C})$ is the free category with formal adjoints on $\mathcal{C}/1$. More concretely, this category is the free category on the graph whose objects are the objects of $\mathcal{C}/1$ as objects and whose arrows $f^n : A \to B$ are pairs constituted of an integer

$n \in \mathbb{Z}$, called *winding number*, and a morphism $f : A \to B$ in $\mathcal{C}$ if n is even (resp. a morphism $f : B \to C$ in $\mathcal{C}$ if n is odd), quotiented by the following equalities:

– for every pair of composable morphisms f^n and g^n,

$$f^n \otimes g^n \quad = \quad \begin{cases} (f \otimes g)^n & \text{if } n \text{ is even} \\ (g \otimes f)^n & \text{if } n \text{ is odd} \end{cases}$$

– for every object A,

$$(\mathrm{id}_A)^n \quad = \quad \mathrm{id}_A$$

The 2-cells of $\mathcal{A}(\mathcal{C})$ are formal vertical and horizontal composites of:

– $\alpha^0 : f^0 \Rightarrow g^0$, where $\alpha : f \Rightarrow g$ is a 2-cell of $\mathcal{C}$,

– $\eta_{f^n} : B \Rightarrow f^{n-1} \otimes f^n$, for every 1-cell $f^n : A \to B$,

– $\varepsilon_{f^n} : f^n \otimes f^{n-1} \Rightarrow A$, for every 1-cell $f^n : A \to B$,

quotiented by

– the axioms of 2-categories (see Section 2.2),

– for every pair of vertically composable 2-cells α^0 and β^0,

$$\beta^0 \circ \alpha^0 \quad = \quad (\beta \circ \alpha)^0$$

– for every 1-cell f^0,

$$\mathrm{id}_{f^0} \quad = \quad (\mathrm{id}_f)^0$$

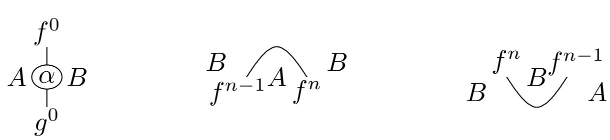
– for every pair of horizontally composable 2-cells α^0 and β^0,

$$\alpha^0 \otimes \beta^0 \quad = \quad (\alpha \otimes \beta)^0$$

– for every 1-cell f^n,

$$(f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^n) = f^{n-1} \quad \text{and} \quad (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) = f^n$$

Graphically, if we write respectively



for $\alpha^0 : f^0 \Rightarrow g^0 : A \to B$, η_{f^n} and ε_{f^n} (where $f^n : A \to B$), the four last equalities can be pictured as in Figure 6.

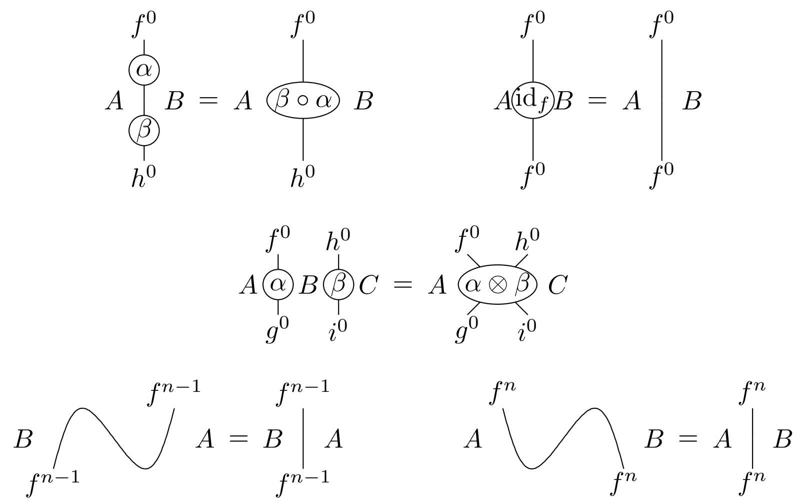


Figure 6: Axioms for free compact categories.

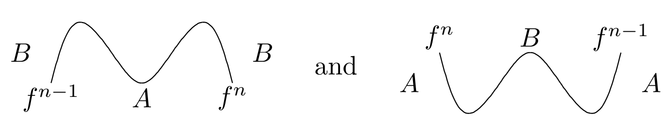
Remark 26. In particular, if $\mathcal{C}$ is the 2-category P^* generated by a 2-polygraph P, the compact 2-category $\mathcal{A}(\mathcal{C})$ is presented by the 3-polygraph Q such that

- $Q_0 = P_0$
- $Q_1 = \{ f^n \mid f \in P_1, n \in \mathbb{Z} \mid \}$
- $Q_2 = \{ \alpha^0 \mid \alpha \in P_2 \mid \uplus \}\{ \eta_{f^n}, \varepsilon_{f^n} \mid f^n \in Q_1 \mid \}$
- $Q_3 = \{ l_{f^n}, r_{f^n} \mid f^n \in Q_1 \mid \}$

with

$$
\begin{array}{rclcl}
l_{f^n} & : & (f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^n) & \Rrightarrow & f^{n-1} \\
r_{f^n} & : & (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) & \Rrightarrow & f^n
\end{array}
$$

and other cells have the obvious source and target. By Lemma 17, the polygraph Q is terminating and by Lemma 16 it is confluent since all its critical pairs, which are of the form



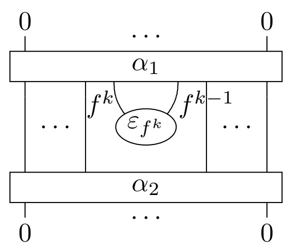
for some 1-cell f^n, are joinable.

Lemma 27. With the notations of the preceding remark, if $f_1, \ldots, f_m$ and $g_1, \ldots, g_n$ are parallel lists of composable morphisms of P^*, then the 2-cells

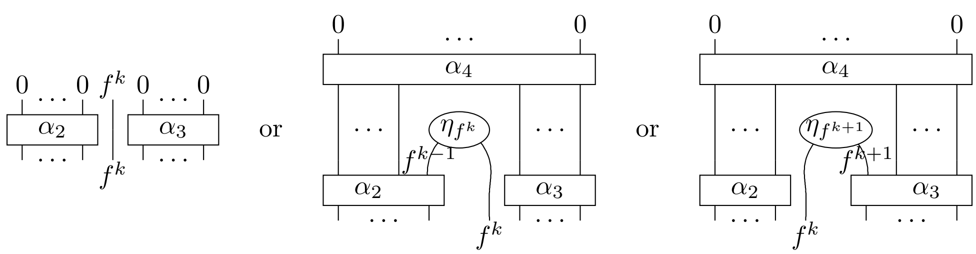
$$
\alpha \quad : \quad f_1^0 \otimes \ldots \otimes f_m^0 \quad \Rightarrow \quad g_1^0 \otimes \ldots \otimes g_n^0 \tag{14}
$$

in the underlying 2-category of Q which are normal forms (with respect to the rewriting rules of Q) do not contain any 2-generator η_{f^k} or ε_{f^k}.

Proof. It is easy to show that a 2-cell α in Q^* can be written as a composite of morphisms of the form $\mathrm{id}_f \otimes \beta \otimes \mathrm{id}_g$ where β is either a 2-cell of $\mathcal{C}$ or a morphism of the form η_{h^k} or ε_{h^k} (see for example [Laf03]). Suppose that α contains a 2-generator of the form ε_{f^k} with $k > 0$. It can therefore be written as a composite of the form



The 2-cells α_1 and α_2 are noted with boxes for clarity and 0 stands for a 1-cell whose winding number is 0. Since the only generators whose target contain a 1-cell of the form f^k are η_k and η_{k+1}, the 2-cell α_1 is necessarily of one of the following forms:



The first case is impossible since we have supposed that all the winding numbers of the 1-generators occurring in the source of α are 0 and $k > 0$. The second case is not possible either since the morphism α would not be a normal form (the rule $l_{f^{k-1}}$ could be applied). Therefore the 2-cell α contains a 2-generator $\eta_{f^{k+1}}$. By using a similar argument, α also contains the 2-generator $\varepsilon_{f^{k+2}}$. So, by induction, the 2-cell α would contain all the 2-generators $\varepsilon_{f^{k+2i}}$ with $i \in \mathbb{N}$ and would therefore be a composite of an infinite number of generators. This is absurd since the 2-cells in Q^* are inductively generated. We deduce that α does not contain a 2-generator of the form ε_{f^k} with $k > 0$. Similarly, it does not contain a 2-generator η_{f^k} with $k > 0$. And the cases where $k \leq 0$ are also similar (we construct an infinite sequence of generators that α would contain, with strictly decreasing winding numbers). $\square$

From this, we can deduce that the 2-cells (14) in bijection with the 2-cells

$$\alpha \quad : \quad f_1 \otimes \ldots \otimes f_m \quad \Rightarrow \quad g_1 \otimes \ldots \otimes g_n$$

of $\mathcal{C}$, which shows that the embedding of P^* into Q^* is full and faithful. Moreover, the argument can easily be generalized to any category $\mathcal{C}$, not necessarily generated by a 2-polygraph (but we will only make use of the case proved in previous lemma):

Property 28. The components $\eta_{\mathcal{C}} : \mathcal{C} \to \mathcal{A}(\mathcal{C})$ of the unit of the adjunction between 2-categories and compact 2-categories are full and faithful.

This property formally explains why we can manipulate the cells of a 2-category $\mathcal{C}$ into the "larger space" $\mathcal{A}(\mathcal{C})$.

4.3 Rotative 2-categories

The following property shows that the distinction between the source and the target of a 1-cell in a compact 2-category is artificial.

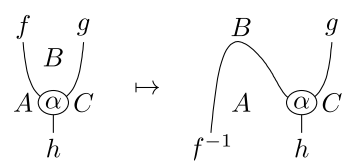
Property 29. If $\mathcal{C}$ is a compact 2-category, the sets

$$\mathrm{Hom}(f \otimes g, h) \quad \cong \quad \mathrm{Hom}(g, f^{-1} \otimes h)$$

are naturally isomorphic by the function

$$\alpha \quad \mapsto \quad (f^{-1} \otimes \alpha) \circ (\eta_f \otimes g)$$

Graphically,



And similarly, the sets

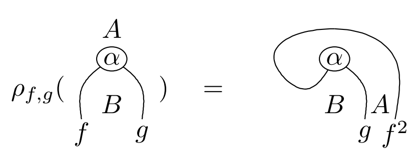
$$\mathrm{Hom}(f \otimes g, h) \quad \cong \quad \mathrm{Hom}(f, h \otimes g^1)$$

are naturally isomorphic by the function

$$\alpha \quad \mapsto \quad (\alpha \otimes g^1) \circ (f \otimes \eta_{g^1})$$

In particular, for any pair of 1-cells $f, g : A \to B$, the set $\mathrm{Hom}(f, g)$ is isomorphic to $\mathrm{Hom}(B, f^{-1} \otimes g)$. This shows that the notion of "input" and "output" of 2-cells is fairly artificial in compact 2-categories. We investigate here an alternative axiomatization of compact 2-categories, where 2-cells have one "border" instead of having both a source and a target.

Given two 1-cells $f : A \to B$ and $g : B \to A$ in a compact 2-category $\mathcal{C}$, we write $\rho_{f,g}$ for the canonical isomorphism, given by Property 29 and called *rotation*, between $\mathrm{Hom}(A, f \otimes g)$ and $\mathrm{Hom}(B, g \otimes f^2)$. Graphically,



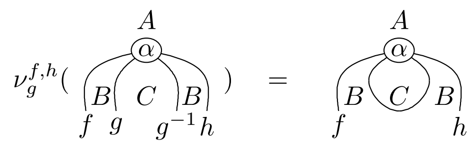
(we sometimes simply write ρ_f when g is clear from the context). We also write

$$\nu_g^{f,h} : \mathrm{Hom}(A, f \otimes g \otimes g^{-1} \otimes h) \to \mathrm{Hom}(A, f \otimes h)$$

for the function, called *hiding*, which to every 2-cell $\alpha : A \Rightarrow f \otimes g \otimes g^{-1} \otimes h$ associates

$$\nu_g^{f,h}\alpha \quad = \quad (f \otimes \varepsilon_g \otimes h) \circ \alpha$$

(we sometimes simply write ν_g when f and h are clear from the context). Graphically,



Together with these functions, every compact 2-category $\mathcal{C}$ induces a structure of what we call a *rotative 2-category* consisting of

1. a category with formal adjoints: the underlying category with formal adjoints of $\mathcal{C}$,

2. for every object A and endomorphism $f : A \to A$ of the category a set $R(f)$ of 2-cells, defined as $R(f) = \mathrm{Hom}(A, f)$ – we sometimes write $\alpha : f$ to indicate that $\alpha \in R(f)$ and call f the *border* of the 2-cell α,

3. for every morphism $f : A \to B$ a distinguished 2-cell $\eta_f : f^{-1} \otimes f$, called the *identity* on f,

4. an invertible function $\rho_{f,g} : R(f \otimes g) \to R(g \otimes f^2)$ called *rotation* for every pair of composable arrows f and g,

5. a function $\otimes_A : R(f) \times R(g) \to R(f \otimes g)$ called *parallel composition* for every 1-cells $f, g : A \to A$,

6. a function $\nu_g^{f,h} : R(f \otimes g \otimes g^{-1} \otimes h)$ called *hiding* for every 1-cells $f : A \to B$, $g : B \to C$ and $h : B \to A$

Moreover, these data are enough to recover the original compact 2-category:

Property 30. Given a rotative 2-category $\mathcal{R}$ induced by a compact 2-category $\mathcal{C}$, we define a compact 2-category $\mathcal{D}$ as follows:

– its underlying category with formal adjoints is the one of $\mathcal{R}$,

– the 2-cells $\alpha : f \Rightarrow g$ are the elements of $R(f^{-1} \otimes g)$,

– vertical composition is given on two 2-cells $\alpha : f \Rightarrow g$ and $\beta : g \Rightarrow h$ by

$$\beta \circ \alpha \quad = \quad \nu_g^{f^{-1},h}(\alpha \otimes \beta)$$

– vertical identities are identity 2-cells of $\mathcal{R}$,

– horizontal composition of two 2-cells

$$\alpha : f \Rightarrow g : A \to B \quad \text{and} \quad \beta : h \Rightarrow i : B \to C$$

is given by

$$\alpha \otimes \beta \quad = \quad \rho^{-1}_{h^{-1}, f^{-1} \otimes g \otimes i}(\alpha \otimes_B (\rho_{h^{-1},i}\beta))$$

– for any 1-cell $f : A \to B$ the units and counits $\eta_f : B \Rightarrow f^{-1} \otimes f$ and $\varepsilon_f : f \otimes f^{-1} \Rightarrow A$ of the adjunctions are given by identities:

$$\eta_f = \mathrm{id}_f \quad \text{and} \quad \varepsilon_f = \mathrm{id}_{f^{-1}}$$

The compact 2-category $\mathcal{D}$ defined as above is isomorphic to the compact 2-category $\mathcal{C}$.

The notion of rotative 2-category can be axiomatized directly in a way such that the category of rotative categories is equivalent to the category of compact 2-categories. The notion of compact 2-category is conceptually nice since it reformulates the concept of compact 2-category using operations which are familiar to concurrency theory and game semantics, decomposing composition in more atomic operations. It is also closely related to the concept of cyclic operad [GK95]. For the lack of space, we did not include the full axiomatization, the only thing we need to know here is that this concept is “equivalent” to compact 2-categories, in the sense explained above. Its use is moreover not fundamental in this work (we could have simply used compact 2-categories) but it simplifies the algorithm for computing critical pairs given in Section 5.1 since we do not have to handle both the source and target of 2-cells, but only their border.

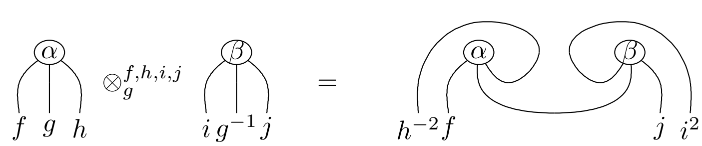
Remark 31. Suppose that $\mathcal{C}$ is a rotative 2-category. To every pair of two cells

$$\alpha : f \otimes g \otimes h \quad \text{and} \quad \beta : i \otimes g^{-1} \otimes j$$

we can associate a 2-cell $\alpha \otimes_g^{f,h,i,j} \beta : h^{-2} \otimes f \otimes j \otimes i^2$ defined by

$$\alpha \otimes_g^{f,h,i,j} \beta \quad = \quad \nu_g((\rho^{-1}_{h^{-2}, f \otimes g}\alpha) \otimes (\rho_{i, g^{-1} \otimes j}\beta))$$

Graphically,



The operations of rotation, composition and hiding can be recovered from this operation as follows:

– *rotation*: for every 1-cells $f : A \to B$ and $g : B \to A$ and 2-cell $\alpha : f \otimes g$,

$$\rho_{f,g}(\alpha) \quad = \quad \mathrm{id}_{\mathrm{id}_B} \otimes_B^{B,B,f,g} \alpha$$

– *composition*: for every pair of 2-cells $\alpha : f : A \to A$ and $\beta : g : A \to A$,

$$\alpha \otimes_A \beta \quad = \quad \alpha \otimes_A^{f,A,A,g} \beta$$

– *hiding*: for every 2-cell $\alpha : f \otimes g \otimes g^{-1} \otimes h$ with $f : A \to B$, $g : B \to C$ and $h : B \to A$,

$$\nu_g(\alpha) \quad = \quad \alpha \otimes_{g \otimes g^{-1}}^{f,h,A,A} \mathrm{id}_{g^{-1}}$$

and moreover, we could have equivalently formalized the notion of rotative category using only this generalized composition operation instead of rotation, composition and hiding. It is more concise, which is why we use it in the following, but leads to a less nice axiomatics.

Remark 32. Given a rotative 2-category $\mathcal{R}$ generated by a compact 2-category $\mathcal{C}$, the multicategory of contexts of $\mathcal{R}$ can be defined similarly to Definition 13, with the 1-cells of $\mathcal{R}$ as objects. Alternatively, this construction can be recovered by restricting the multicategory of contexts of $\mathcal{C}$ to objects of the form $A \Rightarrow f : A \to A$ for some 1-cell $f : A \to A$.

4.4 Compact polygraphs

A *compact 2-polygraph* P can be defined as in (8), where P_1^* (and s_0^* and t_0^*) is generated by a free category with formal adjoints construction on the underlying graph. However, since compact 2-categories are equivalent to rotative 2-categories, it will prove simpler to define a compact 2-polygraph as follows.

A *compact 1-polygraph* is a diagram

$$\begin{tikzcd} P_0 \arrow[d, "i_0"'] & P_1 \arrow[dl, shift right, "s_0"'] \arrow[dl, shift left, "t_0"] \\ P_0^\star & \end{tikzcd}$$

in **Set**, where $P_0^\star = P_0$ and i_0 is the identity (a compact 1-polygraph is therefore the same as a 1-polygraph). Every such (poly)graph freely generates a category with formal adjoints. We write $P_1^\star$ for the set of its morphisms, $i_1 : P_1 \to P_1^\star$ for the canonical injection and $s_0^\star$ and $t_0^\star$ for the morphisms such that $s_0^\star \circ i_1 = s_0$ and $t_0^\star \circ i_1 = t_0$.

Remark 33. The category (with formal duals) thus generated is isomorphic to the category generated by the polygraph

$$\begin{tikzcd} P_0 \arrow[d, "i_0"'] & (P_1 \times \mathbb{Z}) \arrow[dl, shift right, "s_0'"'] \arrow[dl, shift left, "t_0'"] \\ P_0^* & \end{tikzcd}$$

with

$$(s_0'(f,n), t_0'(f,n)) = \begin{cases} (s_0(f), t_0(f)) & \text{if } n \text{ is even,} \\ (t_0(f), s_0(f)) & \text{if } n \text{ is odd.} \end{cases}$$

A *compact 2-polygraph* is a diagram

$$\begin{tikzcd} P_0 \arrow[d, "i_0"'] & P_1 \arrow[dl, shift right, "s_0"'] \arrow[dl, shift left, "t_0"] \arrow[d, "i_1"] & P_2 \arrow[dl, "b_1"] \\ P_0^\star & P_1^\star \arrow[l, shift right, "s_0^\star"'] \arrow[l, shift left, "t_0^\star"] & \end{tikzcd} \tag{15}$$

in **Set**, consisting of a compact 1-polygraph together with the category with formal duals it generates, along with a set P_2 and a function $b_1 : P_2 \to P_1^\star$ such that $s_0^\star \circ b_1 = t_0^\star \circ b_1$. Every such polygraph generates a rotative 2-category whose set of 2-cells is written $P_2^\star$. We write $i_2 : P_2 \to P_2^\star$ for the canonical injection and $b_1^\star$ for the morphism such that $b_1^\star \circ i_2 = b_1$. A *compact 3-polygraph* is a diagram

$$\begin{tikzcd} P_0 \arrow[d, "i_0"'] & P_1 \arrow[dl, shift right, "s_0"'] \arrow[dl, shift left, "t_0"] \arrow[d, "i_1"] & P_2 \arrow[dl, "b_1"] \arrow[d, "i_2"] & P_3 \arrow[dl, shift right, "s_2"'] \arrow[dl, shift left, "t_2"] \\ P_0^\star & P_1^\star \arrow[l, shift right, "s_0^\star"'] \arrow[l, shift left, "t_0^\star"] & P_2^\star \arrow[l, "b_1^\star"'] & \end{tikzcd}$$

in **Set**, consisting of a compact 2-polygraph together with the rotative 2-category it generates, along with a set P_3 and two functions $s_2, t_2 : P_3 \to P_2^\star$ such that $s_2 \circ b_1^\star = t_2 \circ b_1^\star$.

We write $\mathbf{cPol}_n$ for the category of compact n-polygraphs (for $n = 0, 1, 2, 3$).

4.5 The multicategory of contexts of a compact 2-category

4.5.1 Concrete representation of compact nets

The construction of polygraphic nets can be adapted to the setting of rotative 2-categories (and compact 2-categories). As explained in Remark 32, a multicategory of contexts of a compact 2-category can be defined and it can be constructed concretely using nets. We give here a variant of this construction which is suitable for algorithmically manipulating morphisms in compact 2-categories and will be used in Section 5.1 to give an algorithm for computing critical pairs. In particular, composition being done by a pushout construction, there is no need to keep track of winding numbers of inner 1-generators (those which do not belong to any border). Moreover, instead of allowing renaming of generators we define composition as a partial operation (a similar situation occurs in λ-calculus if we don't allow α-conversion: the β-reduction of the term $(\lambda x.M)N$ is defined only if x does not occur as a free variable in N, and usual composition can be recovered by quotienting terms modulo α-conversion later on).

We suppose fixed throughout the section signature which a compact 2-polygraph S of the form (15) that we call the *signature*. We also suppose that we are given three denumerable sets whose elements are called respectively 0-, 1- and 2-generators.

Borders. A *winding path* p is a list of odd length of the form

$$p \quad = \quad x_0, (y_1, w_1), x_1, (y_2, w_2), x_2 \ldots, x_{n-1}, (y_n, w_n), x_n \tag{16}$$

where x_i are 0-generators, y_i are 1-generators, and $w_i \in \mathbb{Z}$ are *winding numbers*, together with a function τ_p which to every x_i associates an element $\tau_p(x_i)$ of S_0 and to every y_i associates an element $\tau_p(x_i)$ of S_1, their *types*, such that for every index $i > 0$

$$\tau_p(x_{i-1}) = s_0 \circ \tau_p(y_i) \quad \text{and} \quad \tau_p(x_i) = t_0 \circ \tau_p(y_i)$$

By extension, for any winding path p, we write $\tau(p)$ for the 1-cell of S defined by $\tau(x_0) = \mathrm{id}_{\tau_p(x_0)}$ and

$$\tau(x_0, (y_1, w_1), x_1, \ldots, (y_n, w_n), x_n) \quad = \quad \tau_p(y_1)^{w_1} \otimes \ldots \otimes \tau_p(y_n)^{w_n}$$

The 0-generators x_{i-1} and x_i are called respectively the *source* and the *target* of the 1-generator y_i; the 0-generators x_0 and x_n are also called respectively the source and the target of the path. We write respectively $p_0 = \{x_i\}$ and $p_1 = \{y_i\}$ for the set of 0- and 1-generators occurring in a winding path p. A *path* p is a list of odd length of the form

$$p \quad = \quad x_0, y_1, x_1, y_2, x_2 \ldots, x_{n-1}, y_n, x_n$$

where x_i are 0-generators and y_i are 1-generators, together with a type function τ_p defined similarly to winding paths. Given two (winding) paths p_1 and p_2 such that the target of p_1 is equal to the source of p_2, we write $p_1 \cdot p_2$ for their concatenation. Given a winding path p, we write $W(p)$ for the path obtained from p by forgetting the winding numbers in p. Similarly, given a 1-cell $f = f_1^{w_1} \otimes \ldots \otimes f_n^{w_n}$ of S, we write $W(f)$ for the 1-cell $f_1 \otimes \ldots \otimes f_n$. Given a winding path p of the form (16), we write p^w, with $w \in \mathbb{Z}$, for the winding path

$$p^w \quad = \quad x_0, (y_1, w_1 + w), x_1, (y_2, w_2 + w), x_2 \ldots, x_{n-1}, (y_n, w_n + w), x_n$$

A (*winding*) *border* b is a (winding) path whose source and target are equal. A (winding) border of the form (16) is *linear* when all the 0-cells and all the 1-cells occurring in it are distinct (excepting the first and last 0-cells which are required to be equal): for every index i such that $0 < i < n$, $x_i \neq x_{i+1}$ and $y_i \neq y_{i+1}$.

Compact nets. A *compact net* N is a finite set N of 2-generators together with a function τ_N which to every element z of N associates a *type* $\tau_N(z) \in S_2$ and a function b_N which to every element z of N associates a (non-winding) *border* $b_N(z)$ such that $\tau(b_N(z)) = W(b_1(\tau_N(z)))$. A 1-cell y in the border of a 2-cell z is an *input* of z if the winding number associated to the corresponding 1-cell in $b_1(\tau_N(z))$ is odd and an *output* otherwise.

The multicategory of nets. We define the multicategory $\mathcal{K}_S$ as the smallest multicategory, whose objects are winding borders and whose operations are nets, such that

- for every 2-generator $\alpha : f$ in the signature, every linear winding border b such that $\tau(b) = f$ and every net $K_\alpha = \{z\}$ with $b_{K_\alpha}(z) = W(b)$ and $\tau_{K_\alpha}(z) = \alpha$, the border b is an object of $\mathcal{K}_S$ and the net K_α is an operation of $\mathcal{K}_S(;b)$,
- for every object b, the empty net, written id_b, is an operation of $\mathcal{K}_S(b;b)$,
- for every objects $b_1 = p_1 \cdot p \cdot p_2$ and $b_2 = p_3 \cdot p \cdot p_4$, such that the sets

$$(p_1)_0 \cap (p_3)_0 \qquad (p_1)_0 \cap (p_4)_0 \qquad (p_2)_0 \cap (p_3)_0 \qquad (p_2)_0 \cap (p_4)_0$$

  are all included in $(p)_0$ and the sets

$$(p_1)_1 \cap (p_3)_1 \qquad (p_1)_1 \cap (p_4)_1 \qquad (p_2)_1 \cap (p_3)_1 \qquad (p_2)_1 \cap (p_4)_1$$

  are all included in $(p)_1$ (i.e. all unbound 0- and 1-generators in b_1, in the sense defined below, are distinct from those in b_2), the empty net, written $\otimes_p^{p_1,p_2,p_3,p_4}$, is an operation of $\mathcal{K}_S(b_1, b_2; p_2^{-2} \cdot p_1 \cdot p_4 \cdot p_3^2)$.

Given an operation $K : b_1, \ldots, b_n \Rrightarrow b$, a k-generator (with $k = 0, 1$) is *bound* when it is in $(b_i)_k \cap (b)_k$ for some index i and *unbound* otherwise. Suppose that we are moreover given n operations

$$K_i \quad : \quad b_1^1, \ldots, b_n^{k_i} \quad \Rrightarrow \quad b_i$$

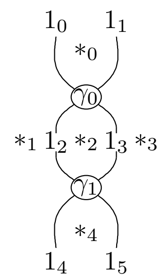
such that the unbound generators of the borders b_i (with respect to K) and the unbound variables of the b_j^i (with respect to K_i) are all pairwise distinct, and moreover the 2-generators of K and of the K_i are all pairwise distinct. Their composition $K \circ (K_1, \ldots, K_n)$ is defined on 2-cells as

$$K \circ (K_1, \ldots, K_n) \quad = \quad K \cup K_1 \cup \ldots \cup K_n$$

and by a coproduct for $\tau_{K \circ (K_1, \ldots, K_n)}$ and $b_{K \circ (K_1, \ldots, K_n)}$.

A *renaming* of generators r is an function mapping 0-, 1- and 2-generators to 0-, 1- and 2-generators respectively. Every renaming induces an obvious morphism on borders and nets that we still write r. Given two i-generators x and x', we sometimes write $x \mapsto x'$ for the renaming r which is the identity excepting on x where $r(x) = x'$. Two operations borders b and b' are *α-equivalent* when there exists an injective renaming r such that $b' = r(b)$ and two operations $K : b_1, \ldots, b_n \Rrightarrow b$ and $K' : b'_1, \ldots, b'_n \Rrightarrow b'$ are *α-equivalent* when there exists an injective renaming r such that $K' = r(K)$, $b' = r(b)$ and $b'_i = r(b_i)$ for every index i. These relations are equivalence relations and we consider objects and operations of $\mathcal{K}_S$ modulo these equivalence relations. It is simple to check that composition is a well-defined total operation. In particular, it is total because the nets we consider involve a finite number of generators and the sets of generators are supposed to be denumerable, so we can always generate "fresh" generators.

Example 34. Consider the signature of symmetries defined in the Example 2 (seen as a compact 2-polygraph). We consider the net N representing the morphism $\gamma \circ \gamma$ whose graphical representation is



which is defined by

$$N_2 = \{\gamma_0, \gamma_1\} \qquad N_1 = \{1_0, 1_1, \ldots, 1_5\} \qquad N_0 = \{*_0, *_1, \ldots, *_4\}$$

with $\tau_N(\gamma_i) = \gamma$, $\tau_N(1_i) = 1$, $\tau_N(*_i) = *$,

$$b_N(\gamma_0) = *_3, 1_1, *_0, 1_0, *_1, 1_2, *_2, 1_3, *_3 \qquad b_N(\gamma_1) = *_3, 1_3, *_2, 1_2, *_1, 1_4, *_4, 1_5, *_3$$

The following are operations in $\mathcal{K}_S$:

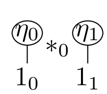
$$N \;:\; \quad \Rrightarrow \quad *_3, (1_1, -1), *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3 \tag{17}$$

$$N \;:\; \quad \Rrightarrow \quad *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3, (1_1, 1), *_0 \tag{18}$$

$$N \;:\; *_2 \quad \Rrightarrow \quad *_3, (1_1, -1), *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3 \tag{19}$$

Suppose that we are given a net $N : b_1, \ldots, b_n \Rrightarrow b$. A 1-generator of this net is an *input* if it occurs in one of the b_i (resp. in b) with an even (resp. odd) winding number and an *output* if it occurs in one of the b_i (resp. in b) with an odd (resp. even) winding number. It can be shown that a 1-generator of N occurs at most twice in the b_i or b, once as an input and once as an output.

Remark 35. A 1-generator can occur twice in the border of a net (apart from being both at the beginning and at the end of the border), i.e. the multicategory $\mathcal{K}_S$ may contain operations with non-linear winding borders. For example, consider the theory of monoids given in Example 1. The morphism $\eta \otimes \eta$ can be represented by the net N pictured as



with type $N : \Rrightarrow *_0, (1_0, 0), *_0, (1_1, 0), *_0$.

Given an operation $N : b_1, \ldots, b_n \Rrightarrow b$, we sometimes write N_2 for the set of 2-cells of N, and N_1 and N_0 for the sets of 1- and 0-cells respectively occurring either in the border $b_N(z)$ of a 2-generator z of N or in b or in one of the b_i. A net M is *distinct* from a net N if $M_i \cap N_i = \emptyset$, with $i = 0$, 1 or 2. A net M is *included* in a net N, what we write $M \subseteq N$, when $M_i \subseteq N_i$, for every cell $x \in M_i$, $\tau_M(x) = \tau_N(x)$, with $i = 0$, 1 or 2, and for every 2-cell $z \in M_2$, $b_M(z) = b_N(z)$.

Connected nets. Two 2-cells of a net N are immediately connected whenever they share 1-generators on their border, i.e. $(b_N(z_1))_1 \cap (b_N(z_2))_1 \neq \emptyset$. They are *connected* when they are transitively immediately connected. The equivalence classes of 2-generators of N with respect to this relation are called *connected components*. A net $N : b_1, \ldots, b_n \Rrightarrow b$ is *connected* when all its 2-generators are in the same connected component and moreover every 0-generator of N occurs in exactly one of the b_i or b (this implies in particular that the b_i and b are linear winding paths). For instance, in Example 34, the net (17) is not connected but the net (19) is.

4.5.2 Operations on nets

Tensoring. Suppose that $M : b_1, \ldots, b_n \Rrightarrow b$ and $N : c_1, \ldots, c_n \Rrightarrow c$ are two disjoint nets, with the winding paths b and c respectively of the form

$$b \quad = \quad x_0, (y_1, w_1), x_1, (y_2, w_2), x_2 \ldots, x_{n-1}, (y_n, w_n), x_n$$

and

$$c \quad = \quad x'_0, (y'_1, w'_1), x'_1, (y'_2, w'_2), x'_2 \ldots, x'_{n'-1}, (y'_{n'}, w'_{n'}), x'_{n'}$$

Given two indices i and j such that $w'_j = w_i - 1$, we write

$$M \underset{y_i = y'_j}{\otimes} N \quad : \quad b_1, \ldots, b_n, c_1, \ldots, c_n \quad \Rrightarrow \quad d$$

where

$$d \quad = \quad b''^{-2} \cdot b' \cdot c'' \cdot c'^2$$

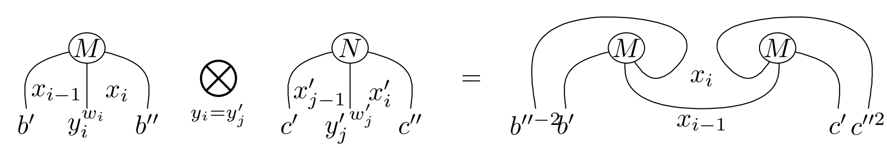
with

$$b = b' \cdot (x_{i-1}, (y_i, w_i), x_i) \cdot b'' \qquad \text{and} \qquad c = c' \cdot (x'_{j-1}, (y'_j, w'_j), x'_j) \cdot c''$$

for the net obtained as the image of the (necessarily disjoint) union of the nets M and N under the renaming r defined by

$$x'_{j-1} \mapsto x_{i-1} \qquad\qquad y'_j \mapsto y_i \qquad\qquad x'_j \mapsto x_i$$

Graphically, this corresponds to the extended form of composition of morphisms which was introduced in Remark 31:



Remark 36. This operation can easily be extended in a similar way to the case where y_i occurs in one of the b_i instead of b (and we still use the same notation for this operation).

Rotating. Suppose that $N : b_1, \dots b_n \Rrightarrow b$ is a net with b of the form

$$b = x_0, (y_1, w_1), x_1, (y_2, w_2), x_2, \dots, x_{n-1}, (y_n, w_n), x_n$$

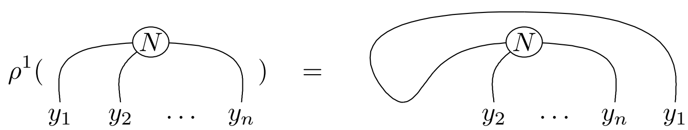
We write $\rho^1(N) : b_1, \dots b_n \Rrightarrow b'$ for the net N (in which only the border has been changed) with

$$b' = x_1, (y_2, w_2), x_2, \dots, x_{n-1}, (y_n, w_n), x_n, (y_1, w_1 + 2), x_1$$

and $\rho^{-1}(N) : b_1, \dots b_n \Rrightarrow b'$ for the net N (in which only the border has been changed) with

$$b' = x_{n-1}, (y_n, w_n), x_n, (y_1, w_1), x_1, \dots, x_{n-2}, (y_{n-1}, w_{n-1}), x_{n-1}$$

Graphically,



More generally, we write $\rho^0(N) = N$ and for every $n \in \mathbb{Z}$, $\rho^{n+1}(N) = \rho^1(\rho^n(N))$ and $\rho^{n-1}(N) = \rho^{-1}(\rho^n(N))$. Again a similar rotation operation can be defined on internal borders and we write $\rho_i^n(N)$ for the net N where the border b_i has been rotated n times (with $n \in \mathbb{Z}$).

Hiding. Suppose that $N : b_1, \dots b_n \Rrightarrow b$ is a net with b of the form

$$b = b' \cdot (x_{i-1}, (y_i, w_j), x_i) \cdot b'' \cdot (x_{j-1}, (y_j, w_j), x_j) \cdot b'''$$

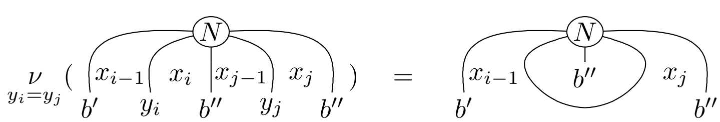
with $w_j = w_i - 1$. We write

$$\underset{y_i = y_j}{\nu}(N) \quad : \quad b_1, \dots b_n, b''^{-1} \quad \Rrightarrow \quad b' \cdot b'''$$

for the image of the net under the renaming defined by

$$x_{j-1} \mapsto x_{i-1} \qquad\qquad y_j \mapsto y_i \qquad\qquad x_j \mapsto y_i$$

Graphically,



This operation can easily be extended in a similar way to the case where both y_i and y_j occur in some internal border b_k of the net instead of the external border b (and we still use the same notation for this operation).

Remark 37. The notation for the previous operations is a bit imprecise because a same 1-generator might occur multiple times (actually at most twice) in the borders of a net. So, in order to be really precise we would have to distinguish between occurrences of a 1-generator in the border. This can be done but it would complicate very much the notations which is why we chose not to handle this precisely here.

4.5.3 Critical pairs

A critical pair between two nets is defined by adapting Definition 19 to this framework (we see the two nets as left members of two rewriting rules).

Definition 38 (Critical pair)**.** A *critical pair* between two nets

$$M : b_1, \ldots, b_n \Rrightarrow b \quad \text{and} \quad N : c_1, \ldots, c_n \Rrightarrow c$$

is a net

$$P : d_1, \ldots, d_n \Rrightarrow d$$

satisfying

- *inclusion*:
$$M \subseteq P \quad \text{and} \quad N \subseteq P$$
- *non-triviality*:
$$M_2 \cap N_2 \neq \emptyset$$
- *minimality*: for $i = 0$, 1 or 2,
$$M_i \cup N_i = P_i$$

Notice that if P is a critical pair of two nets M and N, then every rotation of P is also a critical pair of those. Our algorithm will compute all the critical pairs of the two nets up to α-equivalence and rotation (there are a finite number of equivalence classes up to theses equivalences).

5 Computing critical pairs

We now sketch the unification algorithm, which computes critical pairs for 3-polygraphs. In order to keep the paper with a reasonable size, we have decided to postpone the precise description of the algorithm along with a proof of correctness in future works. However, we felt that it was important to give a rough idea of how the algorithm works because it motivates the introduction of the previous theoretical setting.

It is first important to understand what this algorithm computes exactly. Suppose given a 3-polygraph P containing two 3-generators $r_1 : \alpha \Rrightarrow \alpha' : f \Rightarrow g$ and $r_2 : \beta \Rrightarrow \beta' : h \Rightarrow i$ from which we want to compute the generated critical pairs. It's underlying 2-polygraph S generates a 2-category $\mathcal{C} = S^*$, containing α and β as 2-cells. The canonical full and faithful embedding $\mathcal{C} \to \mathcal{A}(\mathcal{C})$ of $\mathcal{C}$ into the free compact 2-category $\mathcal{A}(\mathcal{C})$ it generates (see Section 4) enables us to see α and β as 2-cells $\alpha^0 : f^0 \Rightarrow g^0$ and $\beta^0 : h^0 \Rightarrow g^0$ of $\mathcal{A}(\mathcal{C})$. In turn, these morphisms can be seen as nullary contexts $K_\alpha : \Rrightarrow f^{-1} \otimes g^0$ and $K_\beta : \Rrightarrow h^{-1} \otimes i^0$ in the multicategory $\mathcal{K}_{\mathcal{A}(\mathcal{C})}$ of compact contexts in $\mathcal{C}$. Our algorithm will therefore compute the "critical pairs" between α and β in the multicategory $\mathcal{K}_{\mathcal{A}(\mathcal{C})}$: it computes a finite number of contexts C_i, which generate all the critical pairs induced by α and β, and are minimal such. More explicitly, given a critical pair $\gamma : k \Rightarrow l$ of α and β, the corresponding compact context $K_\gamma : \Rrightarrow k^{-1} \otimes l^0$ is of the form

$$K \circ C_i \circ (K_1, \ldots, K_n) \quad : \quad \Rrightarrow k^{-1} \otimes l^0 \tag{20}$$

for some computed critical pair C_i, and conversely, every context of the form (20) corresponds to a 2-cell $\gamma : k \Rightarrow l$ of $\mathcal{C}$, which is a non-trivial unifier of α and β.

5.1 A unification algorithm

5.1.1 Auxiliary functions

In this section, we introduce some operations which will be used by the unification algorithm.

It is easy to remark that nets N occurring in $\mathcal{K}_S$ are such that if a 0- or 1-generator occurs in the border of two 2-generators z_1 and z_2 then their types with respect to both 2-generators coincide (but not their winding numbers in general) and write $\tau_N(x)$ and $\tau_N(y)$ for the type of such a 0- or 1-generator x or y. Moreover, given a 1-cell y occurring in such a net N, it occurs in the border of at most two 2-cells z_1 and z_2, once as an input and once as an output. We write $\mathrm{father}_N(y)$ (resp. $\mathrm{son}_N(y)$) for the 2-cell z such that y occurs in its border as an output (resp. as an input); by convention we write $\mathrm{father}_N(y) = \bot$ (resp. $\mathrm{son}_N(y) = \bot$) if there is no such 2-cell. If $N : b_1, \ldots, b_n \Rrightarrow b$ is a net and y is a 1-generator such that $\mathrm{father}_N(y) = \bot$ (resp. $\mathrm{son}_N(y) = \bot$) then it can be shown that y occurs in some b_i with an even (resp. odd) winding number of in b with an odd (resp. even) winding number.

In order to describe the algorithm, some other simple auxiliary functions will be needed.

– Given a 2-generator α of the signature, the function $\mathrm{fresh\_atomic}(\alpha)$ returns an atomic 2-net (in a sense similar to the definition of Section 2.3.3) whose only 2-generator is of type α and whose generators are fresh.

– The function $\mathrm{border\_index}(b, y)$ gives the index of a 2-generator y in a border b.

– The function $\mathrm{border\_ith}(b, i)$ gives the i-th 2-generator of a border b.

– The function $M \otimes_{y_1 = y_2} N$ returns the net which is the disjoint union of M and N together with the renaming described in Section 4.5.2.

– The function put_first(M, y) rotates the borders of M in order to put the 2-generator y in first position when it occurs in a border.

– The function winding(b, y) gives the winding number associated to a 2-generator y in a border b.

– The function merging(M, y_1, y_2) gives the renaming r described in Section 4.5.2, so that $r(M) = \nu_{y_1=y_2}(M)$.

5.1.2 The algorithm

Suppose that we are given two connected nets

$$M : b_1, \dots, b_n \Rrightarrow b \quad \text{and} \quad N : c_1, \dots, c_n \Rrightarrow c$$

for which we want to compute the unifiers. An *unification position* of is a pair $(z_1, z_2) \in N \times M$ of pairs of 2-cells of N and M, often written $z_1 \stackrel{?}{=} z_2$.

States of the algorithm. We now describe our unification algorithm. A *state*

$$S \quad = \quad S_0, S_1, S_2, U_0^M, U_1^M, U_2^M, U_0^N, U_1^N, U_2^N, (P : d_1, \dots, d_n \Rrightarrow d)$$

of the algorithm is a tuple of lists such that the elements of S_i and U_i are elements of $N_i \times P_i$, P is a net and the d_i and d are borders (notice that P doesn't always need to be a proper context). Informally, P is the critical pair which is being constructed, the S_i are the i-generators to unify (called *unification targets*, constituted of a pair of i-generators of N and P) and the U_i^M (resp. U_i^N) are the i-generators already unified in M (resp. in N) – they encode the injection of M (resp. N) into P by a pair of i-generators of M (resp. N) and P. We write $\emptyset$ of the empty list and $S_i = (x_1 \stackrel{?}{=} x_2) :: S_i'$ (resp. $U_i = (x_1 = x_2) :: U_i'$) to indicate that the list S_i (resp. U_i) is not empty with (x_1, x_2) as head and S_i' (resp. U_i') as tail. Given a renaming r, we write $S[r]$ for the for the state where every generator x has been replaced by $r(x)$, excepting in the first component of the elements of S_i and the U_i which are left unchanged (i.e. we rename the cells of P). An *initial state* of the algorithm is a state of the form

$$S \quad = \quad \emptyset, \emptyset, (z_1 \stackrel{?}{=} z_2) :: \emptyset, U_0^M, U_1^M, U_2^M, \emptyset, \emptyset, \emptyset, (M : b_1, \dots, b_n \Rrightarrow b)$$

where (z_1, z_2) is a unification position of M and N and U_i^M is a list whose elements are the couples $(x = x)$ for some i-generator x occurring in M.

The algorithm. Our algorithm consists of an iteration of rules which modify the components of S (starting from an initial state). We write $S_i := S_i'$ to indicate that the next iteration will be done with the state where S_i has been replaced by S_i' (the other elements of the state remaining unchanged), etc. The execution is non-deterministic, a failure of a branch is indicated by *Failure*, and the result is the set of results given by non-failed branches (non-deterministic executions are indicated by "either ... or" or "some" constructions). The algorithm proceeds by executing the first rule which applies and iterating until either a value is returned (by rule SUCCESS) or a *Failure* is triggered. A semi-formal description of the algorithm is as follows:

1. DUPLICATE-0:
   if $S_0 = (x_1 \stackrel{?}{=} x_2) :: S'_0$ and $(x_1 = x'_2) \in U_0^N$ then
     if $x'_2 = x_2$ then $S_0 := S'_0$ else *Failure*

2. DUPLICATE-1:
   if $S_1 = (y_1 \stackrel{?}{=} y_2) :: S'_1$ and $(y_1 = y'_2) \in U_1^N$ then
     if $y'_2 = y_2$ then $S_1 := S'_1$ else *Failure*

3. DUPLICATE-2:
   if $S_2 = (z_1 \stackrel{?}{=} z_2) :: S'_2$ and $(z_1 = z'_2) \in U_2^N$ then
     if $z'_2 = z_2$ then $S_2 := S'_2$ else *Failure*

4. TYPECHECK-2:
   if $S_2 = (z_1 \stackrel{?}{=} z_2) :: S'_2$ then
     if $\tau_P(z_1) = \tau_P(z_2)$ then $S_2 := S'_2$ else *Failure*

5. PROPAGATE-0:
   if $S_0 = (x_1 \stackrel{?}{=} x_2) :: S'_0$ then
     $S_0 := S'_0$

6. PROPAGATE-1:
   if $S_1 = (y_1 \stackrel{?}{=} y_2) :: S'_1$ then
     $S_1 := S'_1$
     if $\mathrm{father}_N(y_1) \neq \bot$ then
       let $z_1 = \mathrm{father}_N(y_1)$ in
         if $\mathrm{father}_P(y_2) \neq \bot$ then
           let $z_2 = \mathrm{father}_P(y_2)$ in
             $U_1 := (y_1 = y_2) :: U_1$
             $S_2 := (z_1 \stackrel{?}{=} z_2) :: S_2$
         else
           either
             let $(A : \Rrightarrow b') = \mathrm{fresh\_atomic}(\tau_N(z_1))$ in
             let $z_2 =$ *the unique 2-generator of* A in
             let $i = \mathrm{border\_index}(b_N(z_1), y_1)$ in
             let $y'_2 = \mathrm{border\_ith}(b_A(z_2), i)$ in
               $(P : b_1, \ldots, b_n \Rrightarrow b), r := P \otimes_{y_2 = y'_2} A$
               $S := S[r]$
           or
             $(P : b_1, \ldots, b_n \Rrightarrow b) := \mathrm{put\_first}(P, y_2)$
             let $e =$ the border b_i or b in which y_2 occurs in
             let $y'_2 =$ some element of e in

if $\text{winding}(e, y_2') \neq \text{winding}(e, y_2) - 1$ then *Failure*
let $r = \text{merging}(P, y_2, y_2')$ in
$S := S[r]$
if $\text{son}_N(y_1) \neq \perp$ then
similar to the previous case

7. Propagate-2:
if $S_2 = (z_1 \stackrel{?}{=} z_2) :: S_2'$ then
let $x_0, y_1, x_1, \ldots, x_{n-1}, y_n, x_n = b_N(z_1)$ in
let $x_0', y_1', x_1', \ldots, x_{n-1}', y_n', x_n' = b_P(z_2)$ in
$S_2 := S_2'$
$U_2^N := (z_1 = z_2) :: U_2^N$
$S_1 := (y_1 \stackrel{?}{=} y_1') :: \ldots :: (y_n \stackrel{?}{=} y_n') :: S_1$
$S_0 := (x_1 \stackrel{?}{=} x_1') :: \ldots :: (x_n \stackrel{?}{=} x_n') :: S_0$

8. Success

In the end of the algorithm, every resulting state S obtained as a result contains a net $P : d_1, \ldots, d_n \Rrightarrow d$ which is a unifier of M and N. In these states, the lists $U_i^M = (x_1^i = {x_1^i}') :: \ldots :: (x_n^i = {x_n^i}')$ induces a morphism of nets $i_M : M \Rrightarrow P$ which to every i-generator x_k^i associates ${x_k^i}'$ which is the injection of M into P (the injection $i_N : N \Rrightarrow P$ is defined similarly using the lists U_i^N).

The purpose of this paper was to introduce the structures necessary to manipulate morphisms in categories generated by polygraphs. We will detail the algorithm in future works and prove that

Claim 39. The algorithm terminates on every pair of two nets M and N and every unifier of the two nets is computed by the algorithm (up to isomorphism and rotation).

5.1.3 An example

The way our algorithm works is best understood by an example. We suppose fixed a signature consisting of

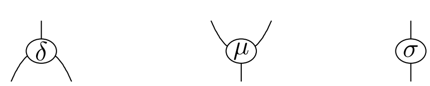
– one 0-generator $*$,

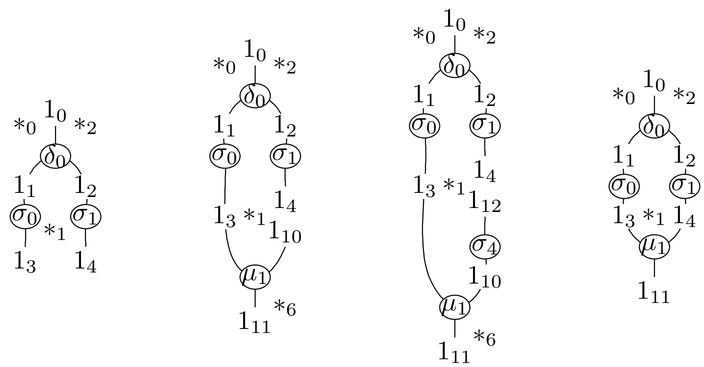
– one 1-generator 1,

– three 2-generators

$$\delta : 1 \to 1 \otimes 1 \qquad \mu : 1 \otimes 1 \to 1 \qquad \sigma : 1 \to 1$$

respectively depicted as

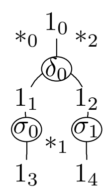




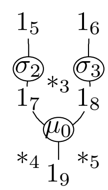
(1) (2) (3) (4)

Figure 7: Various states of the unifier during the execution.

Suppose moreover that we want to unify two morphisms corresponding respectively to the nets M and N whose graphical representation are respectively



and



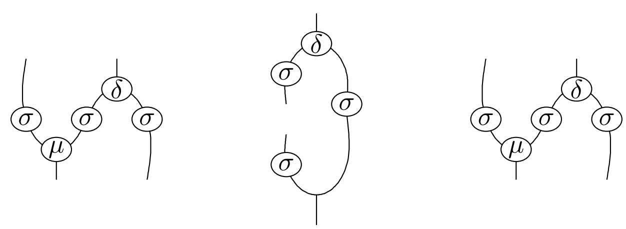
We now describe the main steps during the execution of the algorithm starting from the unification position $\sigma_0 \stackrel{?}{=} \sigma_2$. We write D-1 as a short notation for the rule Duplicate-1, etc.

We start from the "unifier" P, pictured as in (1) of Figure 7 which is equal to M. There is a trivial injection $i_M : M \Rrightarrow P$ (the identity) and our algorithm will "grow" it until there is also an injection $i_N : N \Rrightarrow P$. The rule T-2 first checks that the types of σ_0 and σ_2 coincide, which is the case (they are both equal to σ). Then, the rule P-2 sets $i_N(\sigma_2) = \sigma_0$ and propagates the unification by creating two new unification targets $1_5 \stackrel{?}{=} 1_1$ and $1_7 \stackrel{?}{=} 1_3$. The unification target $1_5 \stackrel{?}{=} 1_1$ leads by P-1 to setting $i_N(1_5) = 1_1$, and $i_N(*_4) = *_0$ and $i_N(*_3) = *_1$ by P-0. Since $\mathrm{son}_N(1_7) = \mu_0$, the unification target $1_7 \stackrel{?}{=} 1_3$ leads to adjoining a fresh atomic 2-net of type μ to P by P-1 and the state now contains the "unifier" P, pictured in (2) of Figure 7, together with the unification target $\mu_0 \stackrel{?}{=} \mu_1$. By P-2, this leads to setting $i_N(\mu_0) = \mu_1$ and creates two new unification targets $1_9 \stackrel{?}{=} 1_{11}$ and $1_8 \stackrel{?}{=} 1_{10}$. The first unification target will eventually lead to setting $i_N(1_9) = 1_{11}$ by P-1, and $i_N(*_5) = *_6$ by P-0. The unification target $1_8 \stackrel{?}{=} 1_{10}$ is more subtle since it will nondeterministically lead to two scenarios by rule P-1 because $\mathrm{father}_N(1_8) = \sigma_3$ and $\mathrm{father}_P(1_{10}) = \perp$:

1. A fresh atomic 2-net of type σ is adjoined to P which becomes (3) of Figure 7 and the unification target remains $i_N(1_8) = 1_{10}$. By the P-i rules, this eventually leads to setting $i_N(1_8) = 1_{10}$, $i_N(\sigma_3) = \sigma_4$ and $i_N(1_6) = 1_{12}$. And the P thus computed is a unifier of M and N.

2. The "unifier" P becomes $\nu_{1_4=1_{10}}(P)$, that is (4) of Figure 7 and the unification target becomes $1_8 \stackrel{?}{=} 1_4$. By the P-i rules, this eventually leads to setting $i_N(1_8) = 1_4$, $i_N(\sigma_3) = \sigma_1$ and $i_N(1_6) = 1_2$. And the P thus computed is a unifier of M and N.

The other unifiers of M and N are



and are computed by starting the algorithm on the unification positions $\sigma_3 \stackrel{?}{=} \sigma_0$, $\sigma_3 \stackrel{?}{=} \sigma_1$ and $\sigma_2 \stackrel{?}{=} \sigma_1$ respectively.

5.1.4 Remarks

In this section we give a few remarks about our algorithm.

The need for contexts. The example given in Section 5.1.3 is simple enough not to really need the structure of multicategory. However, the example given in Figure 3 illustrates why we need it in the general case.

The restriction to connected nets. For simplicity, we have restricted the algorithm to the case where both nets are connected. This is necessary in order for the propagation steps to explore the whole net N. We believe that it can be extended to the general case by using more general unification positions with multiple unification targets. Moreover, this extension does not seem really necessary for now since all the polygraphic rewriting systems the author is aware of can be expressed using only connected nets in the left member of the rules.

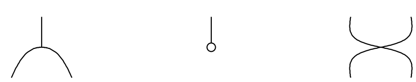
Confluence on metaterms. As explained before, in order to compute the unifier of two morphisms in a 2-category $\mathcal{C}$, we embed this category into a "bigger universe" – the multicategory of compact contexts $\mathcal{K}_{\mathcal{A}(\mathcal{C})}$ – and compute the critical pairs in $\mathcal{K}_{\mathcal{A}(\mathcal{C})}$. The formal justification for this is that the embedding of $\mathcal{C}$ into $\mathcal{A}(\mathcal{C})$ is full and faithful, so we can recover all the critical pairs in $\mathcal{C}$ from the critical pairs in $\mathcal{K}_{\mathcal{A}(\mathcal{C})}$, which are in finite number. For example, the morphism on the right of Figure 2 in the free compact 2-category can be used to generate all the morphisms on the form depicted on the left of the figure. In this sense, they can be thought as generating families of critical pairs.

In order study local confluence of rewriting systems, it would be tempting to study the joinability of critical pairs directly in the free compact 2-category. Unfortunately, the joinability of all critical pairs in the 2-category does not imply the joinability of critical pairs in the free compact 2-category. For example, consider the polygraph corresponding to the theory of symmetrie described in the introduction and in Example 2. It can be shown to be confluent [Laf03], however the critical pair shown on the right of Figure 2 is not joinable in the multicategory of compact contexts (however all the critical pairs it generates which are depicted on the left are joinable). This is very similar to the situation in the rewriting systems of calculi for explicit substutions [Kes07]: some of those systems are confluent on terms, but not confluent if we consider terms with meta-variables.

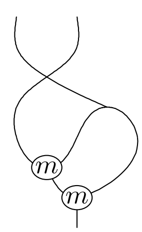
Recovering the usual unification algorithm. Burroni has shown [Bur93] that there is a forgetful functor U from equational term-theories to polygraphic theories, which to equational theory on a term-signature (Σ_n) associates a polygraph S containing: one 0-generator $*$, one 1-generator $1 : * \to *$, a 2-generator $\alpha_i^n : n \to 1$ for every element α_i^n of Σ_n and two 2-generators

$$\delta : 1 \to 2 \quad \text{and} \quad \varepsilon : 1 \to 0 \quad \text{and} \quad \gamma : 2 \to 2 \tag{21}$$

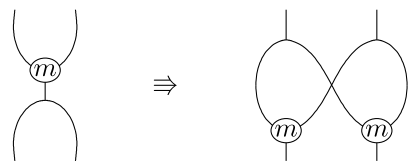
(which should be seen as explicit duplication, erasure and swapping of variables) with the relations corresponding to the relations of the equational term-theory, equations expressing that the generators (21) satisfy the laws of commutative comonoids, and equations expressing the compatibility of the generators (21) with operations coming from the (Σ_n). The generators (21) are usually respectively pictured as



Example 40. Suppose that (Σ_n) is the signature of monoids with multiplication $m \in \Sigma_2$. In a context with two variables x_0 and x_1, a morphism in the polygraphic theory corresponding to the term $m(m(x_1, x_0), x_0)$ is



and for example, the relation expressing compatibility of μ with m is



(which is the usual bialgebra law).

This construction allows us to embed term rewriting systems into polygraphic rewriting systems and thus to compare the usual unification algorithm for terms [BN99] with our algorithm on polygraphic nets. We conjecture that our algorithm can be seen in this case as a "small step" simulation of a variation of the usual unification algorithm.

5.2 Toy implementation

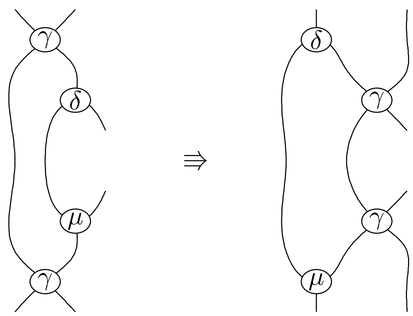
We have made a toy implementation of the algorithm described in Section 5.1.2 in less than 2000 lines of OCaml. It has been used to successfully recover the unifiers of many rewriting systems defined in [Laf03] and, even though we did not particularly focus on efficiency, the execution times are good (typically less than a second on a desktop computer and negligible compared to the compilation time of the produced LaTeX file) because the morphisms involved in typical polygraphic rewriting systems are usually small, even though they can generate a large number of critical pairs.

6 Further directions

We have introduced a representation of morphisms generated by 2-polygraphs which is suitable to manipulate them with a computer and have proposed an algorithm to compute the unifiers of two morphisms in such categories.

We believe that there are many open research tracks left out in this paper, the most obvious one being the proof of correctness and termination of the unification algorithm: our focus here was mainly to establish the main structures necessary to formulate it and we plan to address seriously this topic on subsequent works.

Compact rewriting systems. The use of compact 2-categories seems to be very promising, since it provides a bigger world in which unification is simple to handle (there a finite number of critical pairs in particular). Moreover, left and right members of rules in polygraphic rewriting systems are morphisms in 2-categories, but we can extend the framework to have "compact rewriting rules" whose left and right members are morphisms in compact 2-categories. There is no known finite convergent polygraphic rewriting system presenting the category **Rel** of finite sets and relations [Laf03] (which corresponds to the theory of qualitative bicommutative bialgebras [Mim09]). We conjecture that such a system does not exist. However, we believe that it would be possible to have a finite convergent compact polygraphic rewriting system containing rules such as



where γ is the generator for the symmetry, δ is the comultiplication and μ is the multiplication. We plan to use our unification algorithm in order to define and study such a rewriting system. It would also be interesting to adapt the techniques developed by Guiraud to show termination of polygraphic systems [Gui06a] .

Parametric polygraphs. In order to describe those free compact 2-categories, we had to modify the definition of the notion of polygraph by replacing the free category construction by a free category with formal adjoints construction, and the free 2-category construction by a free compact 2-category construction. This suggest that it might be interesting to investigate a more modular notion of polygraph, parametrized by a series of adjunctions, which could be used to generate free n-categories *with properties* (e.g. compact categories, groupoids, etc.).

Towards higher dimensions. Since the notion of polygraphic rewriting system can be generalized to any dimension, we would like to also have a generalization of rewriting theory to higher dimensions using polygraphic rewriting systems. This would require a more abstract and general formulation of the unification techniques that are used here, in order to be able to extend them easily to higher dimensions.

A practical use of this work. In some sense, our work can be considered as an algebraic study of the notion of a bunch of operators linked by planar wires. We believe that this point of view should be taken seriously and we plan to investigate a possible application of the polygraphic rewriting techniques to electronic circuits. This could namely provide a nice theoretical framework in which we could express and study optimization of integrated circuits.

Thanks. I would like to thank John Baez, Albert Burroni, Jonas Frey, Yves Guiraud, Martin Hyland, Yves Lafont, Paul-André Melliès and François Métayer for all the enlightening discussions we had on the subject of this paper.

References

[BN99] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1999.

[Bur93] A. Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, 1993.

[BW99] J.W. Barrett and B.W. Westbury. Spherical Categories. *Advances in Mathematics*, 143(2):357–375, 1999.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[GK95] E. Getzler and MM Kapranov. Cyclic operads and cyclic homology. *Geometry, topology & physics*, pages 167–201, 1995.

[GM09] Y. Guiraud and P. Malbos. Higher-dimensional categories with finite derivation type. *Theory and Applications of Categories*, 22(18):420–478, 2009.

[Gui06a] Y. Guiraud. Termination orders for 3-dimensional rewriting. *Journal of Pure and Applied Algebra*, 207(2):341–371, 2006.

[Gui06b] Y. Guiraud. The three dimensions of proofs. *Annals of pure and applied logic*, 141(1-2):266–295, 2006.

[Gui06c] Y. Guiraud. Two polygraphic presentations of Petri nets. *Theoretical Computer Science*, 360(1-3):124–146, 2006.

[Hat02] A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2002.

[JS91] A. Joyal and R. Street. The Geometry of Tensor Calculus, I. *Advances in Mathematics*, 88:55–113, 1991.

[JS93] A. Joyal and R. Street. Braided tensor categories. *Advances in Mathematics*, 102(1):20–78, 1993.

[Kes07] D. Kesner. The theory of calculi with explicit substitutions revisited. *Lecture Notes in Computer Science*, 4646:238, 2007.

[KL80] G.M. Kelly and M.L. Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980.

[KS72] G.M. Kelly and R. Street. Review on the Elements of 2-categories. *Proceedings Sydney Category Theory Seminar*, 73:75–103, 1972.

[Laf90] Y. Lafont. Interaction Nets. *Proceedings of the 17th symposium on Principles of programming languages*, pages 95–108, 1990.

[Laf03] Y. Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2-3):257–310, 2003.

[Law63] F. W. Lawvere. *Functorial Semantics of Algebraic Theories and Some Algebraic Problems in the context of Functorial Semantics of Algebraic Theories*. PhD thesis, Columbia University, 1963.

[Lei04] T. Leinster. *Higher Operads, Higher Categories*. Number 298 in London Mathmatics Society Lecture Note Series. Cambridge University Press, 2004.

[Mac71] S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer Verlag, 1971.

[Mim08] S. Mimram. *Sémantique des jeux asynchrones et réécriture 2-dimensionnelle*. PhD thesis, PPS, CNRS – Université Paris Diderot, 2008.

[Mim09] S. Mimram. The Structure of First-Order Causality. In *Proceedings of the 24th symposium on Logic in Computer Science*, pages 212–221, 2009.

[PL07] A. Preller and J. Lambek. Free compact 2-categories. *Mathematical Structures in Computer Science*, 17(02):309–340, 2007.

[Pow90a] J. Power. A 2-Categorical Pasting Theorem. *J. Algebra*, (129):439–445, 1990.

[Pow90b] J. Power. An n-categorical pasting theorem. *Category Theory, Proc. Int. Conf. Como*, pages 326–358, 1990.

[Sel08] P. Selinger. A survey of graphical languages for monoidal categories. Draft manuscript., 2008.

[Str76] R. Street. Limits indexed by category-valued 2-functors. *J. Pure Appl. Algebra*, 8(2):149–181, 1976.