Computing Critical Pairs in Polygraphs

Samuel Mimram*

September 30, 2009

Abstract

Polygraphs generalize to 2-categories the usual notion of equational theory, by describing them as quotients, modulo equations, of freely generated 2-categories on a given set of generators. In order to work with morphisms modulo the equations, it is often convenient to orient the equations into a confluent rewriting system. In the case of a terminating system, confluence can be checked by showing that critical pairs are joinable. However, the computation of the critical pairs is more complicated for polygraphs than for term rewriting systems: in particular, two left members of a rule don't necessarily have a finite number of unifiers. We advocate here that a more general notion of rewriting system should be considered instead, and introduce an operad of compact contexts in a 2-category, in which two rules have a finite number of unifiers. A concrete representation of contexts is proposed, as well as an unification algorithm for these.

*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 and PANDA ANRs.

Term rewriting systems have proven very useful to reason about terms modulo equations. In many cases, the equations can be oriented and completed in a way giving rise to a normalizing (that is confluent and terminating) rewriting system, thus providing a notion of canonical representative of equivalence classes of terms.

In most of the settings, terms are freely generated by a signature (Σ_n), which consists of a family of sets Σ_n (with $n \in \mathbb{N}$) of generators of arity n, and an equational theory is a signature together with a set of equations, which are pairs of terms. For example, the equational theory of monoids contains two generators m and e, whose arities are respectively 2 and 0, and three equations

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

These equations, when oriented from left to right, form a rewriting system which is normalizing. The termination of this system can be shown by giving an interpretation of the terms in a well-founded poset, such that the rewriting rules are strictly decreasing. Since the system is terminating, the confluence can be deduced from the local confluence, which can itself be shown by verifying that 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. A more detailed presentations term rewriting systems along with the techniques to prove their normalization can be found in [BN99].

As a particular case, when the generators of an equational theory are of arity 1, the category of terms modulo the congruence generated by the equations is a monoid, with addition given by composition and neutral element is the identity. A presentation of a monoid $(M,+,0)$ is such an equational theory whose generated monoid 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 $a(a(x)) = x$. These presentations of monoids are a particularly useful since they can provide finite description of monoids which may be infinite, thus allowing a manipulation of these monoids with a computer. More generally, equational theories give presentations of Lawvere theories [Law63], that is cartesian categories whose objects are the natural integers and such that product is given on objects by addition.

Polygraphs are algebraic structures which where introduced in their 2-dimensional version by Street [Str76] under the name computads and later on generalized to higher dimensions by Power [Pow90b] and Burroni [Bur93]. They can be seen as a generalization of term rewriting systems in the sense that they provide a formal framework in which one can give presentations of any (strict) n-category. We are interested here in adapting the techniques mentioned to show the normalization of 3-polygraphs (which present 2-categories), their local confluence in particular. These polygraphs can namely be seen as term rewriting systems improved on the following points :

- the variables of terms are simply typed (this can be thought as switching from a Lawvere theory of terms to a cartesian category of terms),

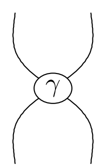
– variables in terms can not necessarily be duplicated, erased or swapped (the categories of terms are not necessarily cartesian but only monoidal),

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

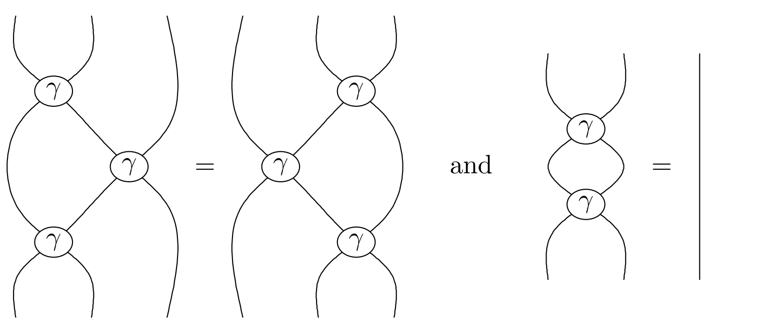
Many examples of presentations of monoidal categories where studied by Lafont [Laf03], Guiraud [Gui06c, Gui06b] and the author [Mim08, Mim09]. For example, Lafont introduced a 3-polygraph B presenting the monoidal category **Bij**, the skeleton of the category of finite sets and bijections. This polygraph 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

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

The morphism 1 in the above equations is a short notation for id_1, the identity on the object 1. That this polygraph is a presentation of the category **Bij** means that this category is isomorphic to the free monoidal category containing a generator γ, quotiented by the smallest congruence generated by the equations (1). These equations can be better understood with the graphical notation provided by string diagrams. The morphism γ should be thought as a device with two inputs and two outputs of type 1, thus depicted as



and the two equations (1) can be represented graphically by


(2)

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. This notation is explained in details in Section 2.3. If we orient both equations from left to right, we get a normalizing rewriting system. It has the three obvious critical pairs depicted in Figure 1.

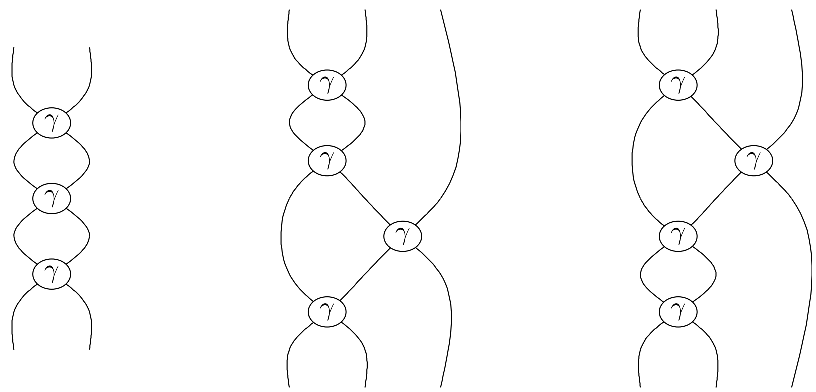


Figure 1: Three critical pairs for symmetries

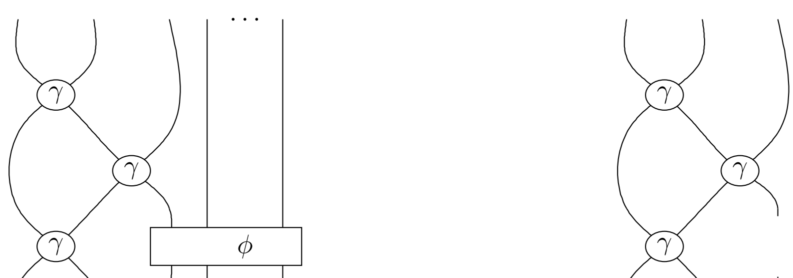


Figure 2: A generic form of a family of critical pairs

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 studied by Guiraud and Malbos [GM08]. Interestingly, we can nevertheless consider that there is a finite number of critical pairs if we allow ourselves to consider the "diagram" on the right of Figure 2 as a critical pair. This observation was the starting point of this paper which is devoted to formalizing these intuitions in order to propose an algorithm to compute 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 step in this direction, by defining the structures necessary to manipulate algorithmically the morphisms in categories generated by polygraphs and by proposing an algorithm to compute the critical pairs in polygraphic rewriting systems. The framework of polygraphs is more subtle than term rewriting systems and 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.

We begin by recalling classical definitions in category theory (Section 1), define polygraphs as well as the categories they generate (Section 2) and formulate the unification problem in the framework of polygraphic rewriting systems (Section 3). Then we show that 2-categories can be fully and faithfully embedded into the free compact 2-category they generate (Section 4) and finally, we describe an unification algorithm for polygraphic rewriting systems (Section 5).

1 Category theory recalled

We recall basic definitions in category theory. A more detailed introduction to category theory can be found in MacLane's reference book [Mac71].

Definition 1 (Category)**.** A *category* $\mathcal{C}$ is given by the data of

- a class $\mathcal{C}_0$ of *objects*,
- a class $\mathcal{C}_1(A, B)$ of *morphisms* for every pair of objects A and B (we write $f : A \to B$ to indicate that $f \in \mathcal{C}_1(A, B)$),
- a function $- \circ - : \mathcal{C}_1(B, C) \times \mathcal{C}_1(A, B) \to \mathcal{C}_1(A, C)$ for every objects A, B and C called *composition*,
- a morphism $\mathrm{id}_A : A \to A$ for every object A called *identity*,

such that

– the composition is associative: for every objects A, B, C and D, and morphisms $f : A \to B$, $g : B \to C$ and $h : C \to D$,

$$(h \circ g) \circ f \quad = \quad h \circ (g \circ f)$$

– the composition admits units as neutral elements: for every objects A and B, and every morphism $f : A \to B$,

$$\mathrm{id}_B \circ f \quad = \quad f \quad = \quad f \circ \mathrm{id}_A$$

Definition 2 (2-category)**.** 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 $\otimes$ and called *vertical composition*, and identities are called *vertical identities*.

– A functor $\circ : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \to \mathcal{C}(A, C)$ 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 $\gamma : h \Rightarrow h'$,

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

– 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 $\alpha : f \Rightarrow f'$,

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

We sometimes simply write A for id_A and f for id_f.

Property 3 (Exchange law)**.** In a 2-category $\mathcal{C}$, for any four 2-cells

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

we have

$$(\beta \otimes \beta') \circ (\alpha \otimes \alpha') \quad = \quad (\beta \circ \alpha) \otimes (\beta' \circ \alpha') \tag{3}$$

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

$$\mathrm{id}_{A \otimes B} = \mathrm{id}_A \otimes \mathrm{id}_B \tag{4}$$

Remark 4. As it can be observed in equation (3), we use the convention for monoidal categories write vertical composition in 2-categories.

Definition 5 (Monoidal category)**.** A *monoidal category* $(\mathcal{C}, \otimes, I)$ is a category together with a functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ called *tensor product*, an object I of $\mathcal{C}$ called *unit* and three invertible natural transformations of components

$$\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C) \qquad \lambda_A : I \otimes A \to A \qquad \rho_A : A \otimes I \to A$$

satisfying MacLane's coherence axioms [Mac71]. A monoidal category is *strict* when the three natural transformations are identities.

In the following, the considered monoidal categories are implicitly supposed to be strict unless explicitly stated otherwise. Such a monoidal category can be considered as a 2-category with only one 0-cell, where the horizontal composition corresponds to tensoring and vertical composition to composition of the monoidal category in the following sense:

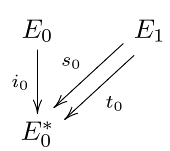
Property 6. The category **StrMonCat** of strict monoidal categories is equivalent to the full subcategory of 2-**Cat** (the category of 2-categories) whose objects are 2-categories with only one object.

2 Free monoidal 2-categories

2.1 Polygraphs

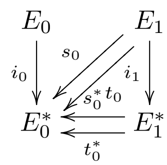
Polygraphs introduced by Burroni [Bur93] as a way to give finite descriptions of categories, generalizing in particular the notion of presentation of a monoid. We only briefly recall the construction of 3-polygraphs, a more formal and detailed account of the general construction of n-polygraphs can be found in Burroni's paper [Bur93].

Suppose that we are given a set E_0 of *0-generators*, such a set will be called a *0-polygraph*. We write $E_0^* = E_0$ and $i_0 : E_0 \to E_0^*$ the identity function (E_0^* should be thought here as the free 0-category, i.e. set, on E_0 and i_0 the corresponding injection). A 1-polygraph on these generators is a graph, that is a diagram



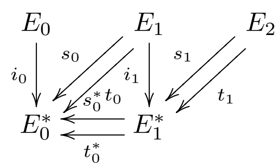
(5)

in **Set**, with E_0 as vertices, the elements of E_1 being called *1-generators*. We can construct a free category on the graph (5): its set E_1^* of morphisms are the paths in the graph (identities are the empty paths), and the source $s_0^*(f)$ (resp. target $t_0^*(f)$) of a morphism $f \in E_1^*$ is the source (resp. target) of the path. If we write $i_1 : E_1 \to E_1^*$ for the injection of the 1-generators into morphisms of this category, which to every 1-generator associates the corresponding path of length one, we thus get a diagram



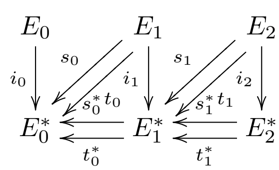
(6)

in **Set**, which is commutative in the sense that $s_0^* \circ i_1 = s_0$ and $t_0^* \circ i_1 = t_0$. A *2-polygraph* on this 1-polygraph consists of a diagram



(7)

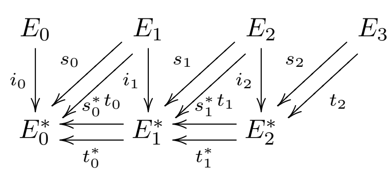
in **Set**, such that $s_0^* \circ s_1 = s_0^* \circ t_1$ and $t_0^* \circ s_1 = t_0^* \circ t_1$. The elements of E_2 are called *2-generators*. Again we can generate a free 2-category on this data, whose underlying category is the category generated in (6) and which has the 2-generators as morphisms. If we write E_2^* for its set of morphisms and $i_2 : E_2 \to E_2^*$ for the injection of the 2-generators into morphisms, we thus get a diagram



(8)

We can now formulate the definition of 3-polygraphs as follows.

Definition 7 (3-polygraph)**.** A *3-polygraph* consists of a diagram



(9)

(where E_i^*, s_i^* and t_i^* are freely generated as previously explained), 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

$$E_0^* \overset{s_0^*}{\underset{t_0^*}{\leftleftarrows}} E_1^* \overset{s_1^*}{\underset{t_1^*}{\leftleftarrows}} E_2^*$$

Again, a 3-polygraph freely generates a 3-category $\mathcal{C}$ whose underlying 2-category is the underlying 2-category of the polygraph and whose 3-cells are generated by the 3-generators of the polygraph. A quotient 2-category $\tilde{\mathcal{C}}$ can be constructed from this 2-category: it is defined as the underlying 2-category of $\mathcal{C}$ quotiented by the congruence identifying two 2-cells whenever there exists a 3-cell between them in $\mathcal{C}$. A 3-polygraph P *presents* a 2-category $\mathcal{D}$ when $\mathcal{D}$ is isomorphic to the 2-category $\tilde{\mathcal{C}}$ generated by the polygraph. In this sense, the underlying 2-polygraph of a 3-polygraph can be thought as a *signature* generating terms which are to be considered modulo the *relations* described by 3-generators.

A *morphism of polygraphs* F between two 3-polygraphs P and Q is a 4-uple (F_0, F_1, F_2, F_3) of functions $F_i : E_i^P \to E_i^Q$, such that the obvious diagrams commute (for example, for every i, $s_i^Q \circ F_{i+1} = F_i^* \circ s_i^P$, where $F_i^* : E_i^{P*} \to E_i^{Q*}$ is the monoid morphism induced by F_i). We write 3-**Pol** for the category of 3-polygraphs.

Example 8. 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.):

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

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

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

The construction of polygraphs can be carried on in any dimension n and we write n-**Pol** for the category of n-polygraphs (we don't detail this construction here since we will be mostly interested in polygraphs in dimension 3 or lower). 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. This operation induces a forgetful functor $U_m^n : n\text{-}\mathbf{Pol} \to m\text{-}\mathbf{Pol}$ which admits a left adjoint F_m^n such that $U^n_m \circ F^n_m = \mathrm{Id}_{m\text{-}\mathbf{Pol}}$:

$$m\text{-}\mathbf{Pol} \underset{U^n_m}{\overset{F^n_m}{\rightleftarrows}} n\text{-}\mathbf{Pol} \quad (\perp) \tag{10}$$

These functors compose in the sense that for every integers $m \leq n \leq o$, we have $F^o_m = F^o_n \circ F^n_m$, $F^m_m = \mathrm{Id}_{m\text{-}\mathbf{Pol}}$ and similarly for U^n_m.

The sets of k-cells (for $0 \leq k \leq n$) of an n-polygraph P will be denoted E^P_k. A polygraph P is *finite* when all the sets E^P_k are finite. In the following, we will only consider finite polygraphs, which are the polygraphs of interest for computer scientific applications, thus avoiding "size issues".

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.

Suppose that we are given a 1-polygraph of the form (5), i.e. a graph. The category generated by this polygraph has the elements A of E_0 as objects and its sets of morphisms are the smallest sets such that

- for every 1-generator $f \in E_1$, 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 $g \circ f : A \to C$,
- for every 0-generator $A \in E_0$, 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 (see Definition 1). 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.

The 2-category freely generated by a 2-polygraph of the form (7) 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, quotiented by the smallest congruence (with respect to both compositions) such that

1. horizontal composition is associative and admits horizontal identities as neutral elements,
2. vertical composition is associative and admits vertical identities as neutral elements,
3. the exchange laws (3) and (4) 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 $f : A \Rightarrow A$ and $g : A \Rightarrow A$ are two 2-cells in a given 2-category $\mathcal{C}$. The equality

$$f \otimes g \quad = \quad g \otimes f$$

can be deduced from the following sequence of equalities:

$$f \otimes g = (\mathrm{id}_A \circ f) \otimes (g \circ \mathrm{id}_A) = (\mathrm{id}_A \otimes g) \circ (f \otimes \mathrm{id}_A) = g \otimes f$$

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.

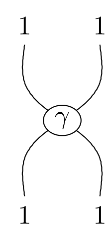
2.3 String diagrams

Joyal and Street [JS91] gave a geometrical construction of the 2-category generated by a 2-polygraph, formalizing the representation of the morphisms in these categories by *string diagrams*. We briefly recall their construction here. An exhaustive survey on string diagrams and variations thereupon can be found in [Sel08].

Given an object 1 in a strict monoidal category $\mathcal{C}$, a morphism

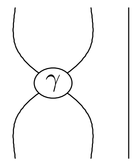
$$\gamma : 1 \otimes 1 \to 1 \otimes 1$$

can be drawn graphically as a device with two inputs and two outputs of type 1 as follows:

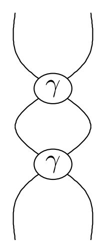


In the following, we sometimes omit the source and target of the morphisms when they are clear from the context. Similarly, the identity $\mathrm{id}_1 : 1 \to 1$ (which we sometimes simply write M) can be pictured as a wire

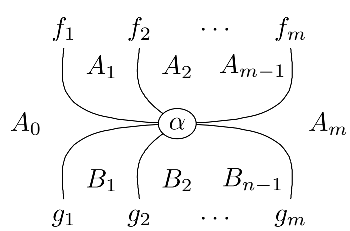
The tensor $f \otimes g$ of two morphisms $f : A \to B$ and $g : C \to D$ is obtained by putting the diagram corresponding to f on the left of the diagram corresponding to g. So, for instance, the morphism $\gamma \otimes \mathrm{id}_1$ can be drawn diagrammatically as



Finally, the composite $g \circ f : A \to C$ of two morphisms $f : A \to B$ and $g : B \to C$ can be drawn diagrammatically by putting the diagram corresponding to g below the diagram corresponding to f and "linking the wires". The diagram corresponding to the morphism $\gamma \circ \gamma$ is thus



These diagrams were introduced to represent morphisms in monoidal categories, but they can be used more generally to represent morphisms in 2-categories. In these diagrams, points correspond to 2-cells, lines to 1-cells and portions of the plane to 0-cells. So more generally, given a 2-polygraph (7), a generator $\alpha : f_1 \otimes \ldots \otimes f_m \Rightarrow g_1 \otimes \ldots \otimes g_n$ with $f_i : A_{i-1} \to A_i$ and $g_i : B_{i-1} \to B_i$ (with $A_0 = B_0$ and $A_m = B_n$) can be represented by a diagram

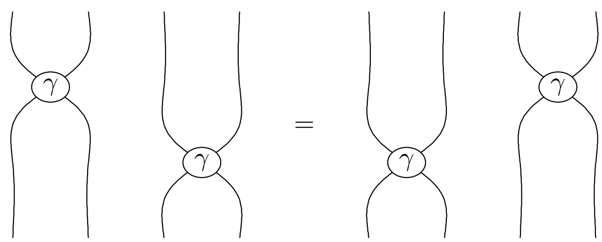


and bigger diagrams can be constructed from these diagrams by composing and tensoring them, as explained above – for example, the diagrams (2) are the graphical representation of the equations (1) given in the introduction. Joyal and Street have shown in details that the category of those diagrams, modulo planar isotopies, is precisely the free 2-category generated by a 2-polygraph (as a matter of fact they have shown this for monoidal categories generated by a

2-polygraph with only one 0-generator but their result can be straightforwardly extended to the case of 2-categories). 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 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.

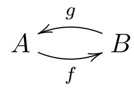
2.4 Categorical nets

The construction of the 2-category generated by a polygraph given in Section 2.2 is algebraic but requires to consider morphism modulo a congruence which is difficult to work with. On the other hand, string diagrams described in Section 2.3 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 *categorical nets*, 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.

Categorical nets are based on the idea that a term generated by a particular signature S is itself an object of the same nature as a signature whose elements are labelled by the elements of the signature S. For example, consider the 1-polygraph S with

$$E_0^S = \{A, B\} \quad \text{and} \quad E_1^S = \{f : A \to B, g : B \to A\} \tag{11}$$

which can be represented graphically by



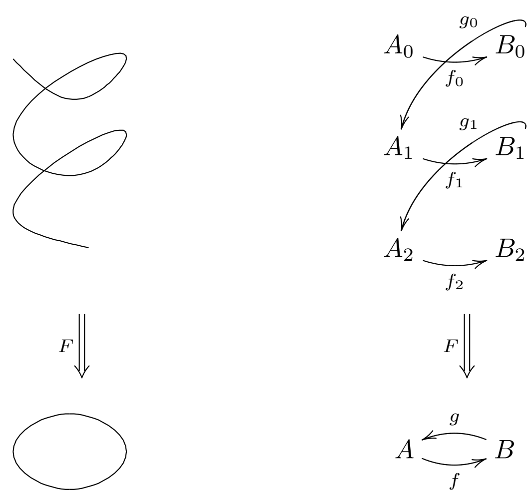
We will see the term t defined as $f \circ g \circ f \circ g \circ f : A \to A$, which is generated by the previous signature S, as a polygraph P such that

$$E_0^P \quad = \quad \{A_0, B_0, A_1, B_1, A_2, B_2\}$$

and

$$E_1^P = \{f_0 : A_0 \to B_0, g_0 : B_0 \to A_1, f_1 : A_1 \to B_1, g_1 : B_1 \to A_2, f_2 : A_2 \to B_2\}$$

such that the A_i, B_i, f_i and g_i are labelled by A, B, f and g respectively (formally, these labels are given by a morphism F from the polygraph P to the polygraph 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 the following figure.



(the picture on the left represents the circle S^1 together with part of its universal covering, which is an infinite spiral). We will not make explicit use of this similarity with geometry, 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 2-$\mathbf{Net}_S$ whose k-cells are themselves k-polygraphs. We will make use of the notion of slice category, which is defined as follows [Mac71]:

Definition 10 (Slice category)**.** Suppose that we are given category $\mathcal{C}$ and an object A of $\mathcal{C}$. The *slice category* of $\mathcal{C}$ over A, written $\mathcal{C}\downarrow A$ is the category whose objects are pairs (B, f), where B is an object of $\mathcal{C}$ and $f : B \to A$ is a morphism of $\mathcal{C}$, and morphisms $h : (B, f) \to (C, g)$ are the morphisms $h : B \to C$ of $\mathcal{C}$ such that $f = g \circ h$.

We also recall that the categories of n-polygraphs are cocomplete, in particular pushouts always exist.

2.4.1 Categorical 0-nets

First, suppose that we are given a 0-polygraph S (i.e. a set E_0). An *atomic* 0-polygraph is a polygraph which contains only one 0-cell. The category 0-$\mathbf{Net}_S$ of *categorical* 0-*nets* on this polygraph is the subcategory of 0-$\mathbf{Pol}{\downarrow}S$, whose objects are atomic polygraphs over S and whose morphisms are the isomorphisms. This category 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 E_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, 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 $i \in \mathbb{N}$.

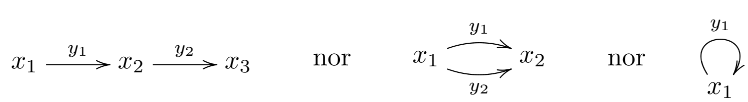
The category of categorical 0-nets on S is equivalent to the set E_0, seen as a category with only identities.

2.4.2 Categorical 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 the source and the target 0-cells of this 1-cell are distinct. Graphically, an atomic polygraph looks like

$$x_1 \xrightarrow{y} x_2$$

but not like



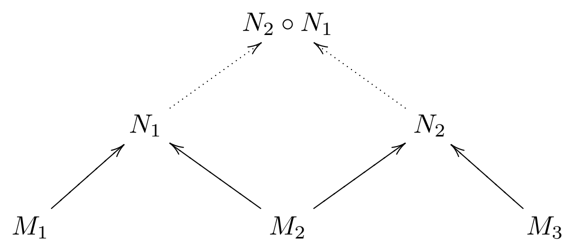
Suppose fixed a 1-polygraph S. The bicategory 1-$\mathbf{Net}_S$ of *categorical* 1-*nets* on S has the 0-nets M on $S/0$ as objects. The inclusion (10) of 0-$\mathbf{Pol}$ into 1-$\mathbf{Pol}$ induces an inclusion functor from 0-$\mathbf{Pol}{\downarrow}(S/0)$ to 1-$\mathbf{Pol}{\downarrow}S$, enabling us to see 0-nets on $S/0$ as elements of this last category. Morphisms $N : M_1 \to M_2$ of 1-$\mathbf{Net}_S$ are the smallest set of cospans

$$M_1 \xrightarrow{s} N \xleftarrow{t} M_2 \tag{12}$$

in 1-$\mathbf{Pol}{\downarrow}S$ such that

– every cospan (12) 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 a pushout



is a morphism,

– for every 0-net M, the cospan

$$M \xrightarrow{M} M \xleftarrow{M} M$$

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 11. Consider the polygraph S defined in (11) and the polygraphs N_1 and N_2 defined by

$$E_0^{N_1} = \{A_0, B_0\} \quad \text{and} \quad E_1^{N_1} = \{f_0 : A_0 \to B_0\}$$

and

$$E_0^{N_2} = \{B_0, A_0\} \quad \text{and} \quad E_1^{N_2} = \{g_0 : B_0 \to A_0\}$$

These polygraphs are elements of 1-$\mathbf{Pol} \downarrow S$ with the obvious 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

$$E_0^{N_2 \circ N_1} = \{A_0, B_0, A_1\} \quad \text{and} \quad E_1^{N_2 \circ N_1} = \{f_0 : A_0 \to B_0, g_0 : B_0 \to A_1\}$$

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 $t_0(f) = s_0(g)$. Categorical 1-nets can be characterized with this relation as follows.

Property 12. The 1-cells of the bicategory 1-$\mathbf{Net}_S$ are precisely the polygraphs $N : M_1 \to M_2$ of 1-$\mathbf{Pol}\downarrow S$ which are

1. *linear*: a 0-generator $x \in E_0^N$ is the source (resp. the target) of exactly one 1-generator $y \in E_1^N$, 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 is the same as a linear graph

$$x_0 \xrightarrow{y_1} x_1 \xrightarrow{y_2} x_2 \cdots x_{n-1} \xrightarrow{y_n} x_n \tag{13}$$

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

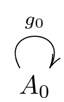
Property 13. The category on a signature S, obtained from the bicategory 1-$\mathbf{Net}_S$ by quotienting objects and morphisms by isomorphism of polygraphs, is isomorphic to the free category generated by the polygraph S.

Remark 14. 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 1-$\mathbf{Pol}{\downarrow}S$ such that

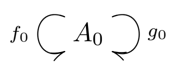
$$E_0^P = \{A_0\} \qquad E_1^P = \{f_0 : A_0 \to A_0\} \qquad E_0^Q = \{A_0\} \qquad E_1^Q = \{g_0 : A_0 \to A_0\}$$

Graphically, P and Q can be respectively pictured as

and



Notice that by Property 12, these polygraphs are not morphisms in 1-$\mathbf{Net}_S$ (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



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).

2.4.3 Categorical 2-nets

A 2-polygraph P is *atomic* when

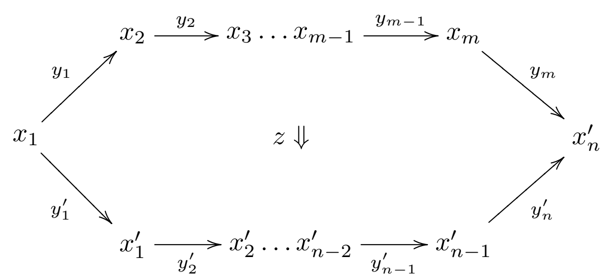
- it contains only one 2-generator z,

- the source and target $s_1(z)$ and $t_1(z)$ are 1-nets,

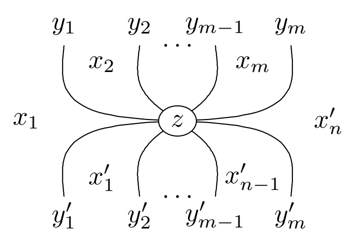
– $E_1^{s_1(z)} \cap E_1^{s_1(z)} = \emptyset$,

– and $E_0^{s_1(z)} \cap E_0^{s_1(z)} = \{s_0(s_1(z)), t_0(t_1(z))\}$.

Graphically, an atomic 2-polygraph looks like



in diagrammatic notation, or like

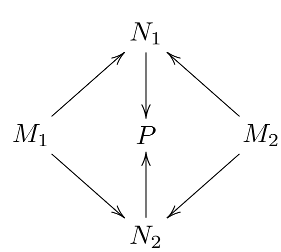


in string-diagrammatic notation, where the y_i and y'_i are all distinct and the x_i and x'_i are all distinct.

Suppose that we are given a 2-polygraph S as in (7). The 2-category (or more precisely tricategory) 2-$\mathbf{Net}_S$ of categorical 2-nets on S is defined by a generalization of the previous construction. Its underlying category is the category 1-$\mathbf{Net}_{S/1}$ of 1-nets on the polygraph $S/1$. Again, such 1-nets can be seen as objects in the 2-category 2-$\mathbf{Pol}{\downarrow}S$. The 2-cells $P : N_1 \Rightarrow N_2 : M_1 \to M_2$ of 2-$\mathbf{Net}_S$ will be cospans

$$N_1 \xrightarrow{\ s\ } P \xleftarrow{\ t\ } N_2 \tag{14}$$

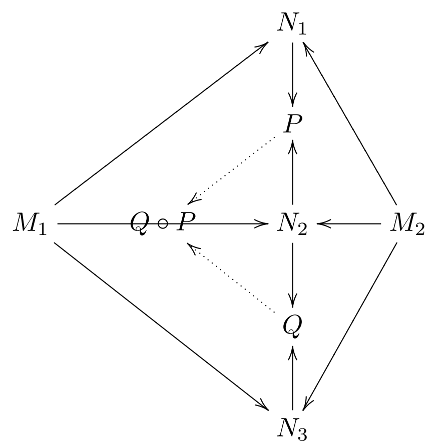
in the category 2-$\mathbf{Pol}{\downarrow}S$ such that N_1 and N_2 are both 1-nets from M_1 to M_2, thus inducing diagrams of the form



Vertical composition of two morphisms

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

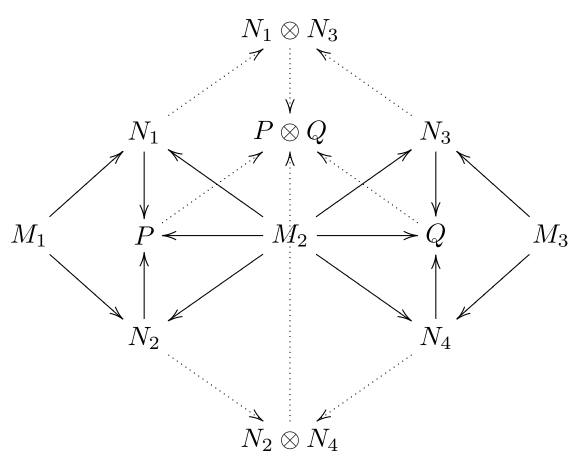
is given by pushout



of consecutive cospans of the form (14) and composition $P \otimes Q$ of two morphisms

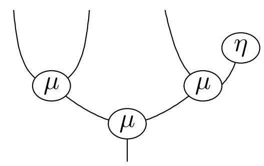
$$P : N_1 \Rightarrow N_2 : M_1 \to M_2 \quad \text{and} \quad Q : N_3 \Rightarrow N_4 : M_2 \to M_3$$

is given by the following sequence of pushouts



(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 2-$\mathbf{Net}_S$ 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 1-$\mathbf{Net}_{S/1}$ is a bicategory and composition of spans is not strictly associative, we have defined a tricategory. However, in the following we will consider 2-nets up to isomorphism (which corresponds to renaming of cells) and these form a (strict) 2-category. These 2-nets can be seen as an algebraic reformulation of Power's pasting schemes [Pow90a].

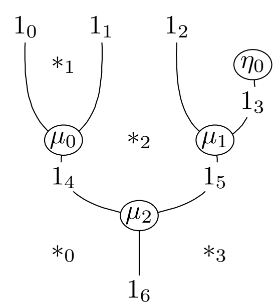
Example 15. The morphism $\mu \circ (\mu \otimes (\mu \circ (1 \otimes \eta)))$ in the theory M of monoids (see Example 8) whose string-diagrammatic notation is



can be represented by the polygraph whose generators are

$$
\begin{array}{rcl}
E_0 & = & \{*_0,\ *_1,\ *_2,\ *_3\} \\
E_1 & = & \{1_0 : *_0 \Rightarrow *_1,\ 1_1 : *_1 \Rightarrow *_2,\ 1_2 : *_2 \Rightarrow *_3, \\
 & & \ 1_3 : *_3 \Rightarrow *_3,\ 1_4 : *_0 \Rightarrow *_2,\ 1_5 : *_2 \Rightarrow *_3,\ 1_6 : *_0 \Rightarrow *_3\} \\
E_2 & = & \{\eta_0 : *_3 \Rrightarrow 1_3,\ \mu_0 : 1_0 \otimes 1_1 \Rrightarrow 1_4, \\
 & & \ \mu_1 : 1_2 \otimes 1_3 \Rrightarrow 1_5,\ \mu_2 : 1_4 \otimes 1_5 \Rrightarrow 1_6\}
\end{array}
$$

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



(15)

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

Since the definition of the category 2-$\mathbf{Net}_S$ is given by a free completion by certain pushouts of nets, it can be shown that

Theorem 16. The 2-category 2-$\mathbf{Net}_S$ on a signature S is equivalent to the free 2-category generated by the polygraph S.

Property 17. The *vertical ordering* relation $<_1$ is the smallest transitive relation on 2-generators E_2^P of a 2-polygraph P such that $z <_1 z'$ whenever $E_1^{t_1(z)} \cap E_1^{s_1(z')} \neq \emptyset$. The 2-polygraphs

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

in 2-$\mathbf{Net}_S$ are

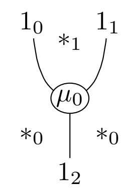
– *linear*: a 1-generator $y \in E_1^P$ is in the source (resp. in the target) of exactly one 2-generator $z \in E_2^P$, i.e. $y \in E_1^{s_1(z)}$ (resp. $y \in E_1^{t_1(z)}$), excepting for the elements of $E_0^{t(P)}$ (resp. $E_0^{s(P)}$) which are in the source (resp. target) of none,

– *acyclic*: the relation $<_1$ is irreflexive.

Remark 18. In the Example 15, 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 12 (i.e. the horizontal ordering $<_0$ is not irreflexive).

Remark 19. Property 17 does not give a characterization of 2-polygraphs which are 1-nets. For example, the polygraph

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



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

Recall that a 3-polygraph S freely generates a 3-category. Two coinitial 3-cells

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

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 $s_1 \circ_2 r_1 = s_2 \circ_2 r_2$. The polygraph S is *locally confluent* when every pair of coinitial 3-cells containing only one 3-generator in the generated 3-category is joinable, and *confluent* when every pair of 3-cells in the generated 3-category is joinable. It is *terminating* when there is no infinite sequence $\alpha_1 \Rrightarrow^{r_1} \alpha_2 \Rrightarrow^{r_2} \ldots$, where the r_i are 3-generators.

The usual Newman's lemma is still valid in this framework:

Lemma 20. A terminating polygraph is confluent if and only if it is locally confluent.

We have seen in Section 2.4.3, that the 2-cells of the 3-category generated by a 3-polygraph S can be seen as categorical 2-nets. The *size* $\|\alpha\|_2$ of a 2-cell $\alpha : f \Rightarrow g$ is the number of 2-generators occurring in a 2-net corresponding to α. In some simple cases, termination of polygraphs can be deduced from the following lemma:

Lemma 21. A 3-polygraph S such that for every 3-generator $r : \alpha \Rrightarrow \beta$ we have $\|\alpha\|_2 > \|\beta\|_2$ 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].

3.1 The multicategory of monoidal contexts

We introduce here the notion of *context* in a 2-category. 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 22 (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 $f_i : A_i^1, \ldots, A_i^{k_i} \to A_i$, for $1 \leq i \leq n$, and $f : A_1, \ldots, A_n \to A$, associates a composite operation

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

  that we often simply write $f(f_1, \ldots, f_n)$,
- an operation $\mathrm{id}_A : A \to A$, called *identity*, for every object A,

such that

- the composition is associative:

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

  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 $f : A_1, \ldots, A_n \to A$, we have $f \circ (\mathrm{id}_A, \ldots, \mathrm{id}_A) = f$.

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

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

Definition 23 (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 are 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 sets of morphisms

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

are the functions

$$\mathrm{Hom}_\mathcal{C}(f_1, g_1) \times \ldots \times \mathrm{Hom}_\mathcal{C}(f_n, g_n) \quad \to \quad \mathrm{Hom}_\mathcal{C}(f, g)$$

with composition induced by composition of functions, i.e.

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

in the smallest sets of such functions, closed under composition, such that

- for every 2-cell $\alpha : f \to g$ in $\mathcal{C}$ the constant function

  $$K_\alpha \quad : \quad 1 \quad \to \quad \mathrm{Hom}(f, g)$$

  which to the unique element of the terminal set 1 associates the morphism α is an element of $\mathcal{K}_\mathcal{C}(; f \Rightarrow g)$,

- for every object $f \Rightarrow g$, the identity function is an element of the set $\mathcal{K}_\mathcal{C}(f \Rightarrow g; f \Rightarrow g)$, and is the identity $\mathrm{id}_{f \Rightarrow g}$ of the operad on $f \Rightarrow g$,

- for every pair of objects $f \Rightarrow g$ and $g \Rightarrow f$, the vertical composition function

  $$K_\circ \quad : \quad \alpha, \beta \quad \mapsto \quad \beta \circ \alpha$$

  is an element of $\mathcal{K}_\mathcal{C}(f \Rightarrow g, g \Rightarrow h; f \Rightarrow h)$,

- for every horizontally composable pair of functions $f \Rightarrow g$ and $f' \Rightarrow g'$, the horizontal composition function

  $$K_\otimes \quad : \quad \alpha, \beta \quad \mapsto \quad \alpha \otimes \beta$$

  is an element of $\mathcal{K}_\mathcal{C}(f \Rightarrow g, f' \Rightarrow g'; (f \otimes f') \Rightarrow (g \otimes g'))$

- for every morphism

  $$K \in \mathcal{K}_\mathcal{C}(f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n; f \Rightarrow g)$$

  and every permutation $\pi : n \to n$, the function $K \circ \pi^{-1}$ is an element of

  $$\mathcal{K}_\mathcal{C}(f_{\pi(1)} \Rightarrow g_{\pi(1)}, \ldots, f_{\pi(n)} \Rightarrow g_{\pi(n)}; f \Rightarrow g)$$

  and induce the symmetric structure of the multicategory.

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. Similarly, if $\mathcal{C}$ is the underlying 2-category of a 3-polygraph, $K : f_1 \Rightarrow g_1 \Rrightarrow f \Rightarrow g$, and $r : \alpha \Rrightarrow \beta : f_1 \Rightarrow g_1$ is a 3-cell in the 3-category generated by the polygraph, we write $K(r) : K\alpha \Rrightarrow K\beta : f \to g$ for the obvious 3-cell.

The construction of the multicategory of contexts $\mathcal{K}_{\mathcal{C}}$ of a 2-category can be made more explicit, by formalizing the intuition that an operation

$$K \quad : \quad f_1 \Rightarrow g_1, \ldots, f_n \Rightarrow g_n \quad \Rrightarrow \quad f \Rightarrow g \tag{16}$$

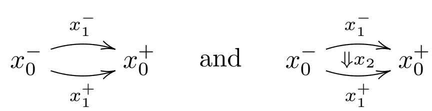
in this operad is a morphism of $\mathcal{C}$ involving "variables" $X_1, \ldots, X_n$ which are formal 2-cells of type $X_i : f_i \Rightarrow g_i$, each of this variable occurring exactly once in the operation. We explain briefly this construction, which is carried on in details in [GM08]. The *standard n-sphere* S_n (for $n \geq -1$) is the n-category generated by the n-polygraph whose set of i-generators (for $0 \leq i \leq n$) is $E_i = \{x_i^-, x_i^+\}$ with source and target functions

$$s_i(x_{i+1}^\varepsilon) = x_i^- \qquad \text{and} \qquad t_i(x_{i+1}^\varepsilon) = x_i^+$$

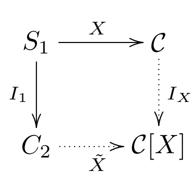
for $\varepsilon \in \{-, +\}$ and $0 \leq i < n$. The *standard n-cell* C_n (for $n \in \mathbb{N}$) is the n-category generated by the n-polygraph, whose underlying $(n-1)$-polygraph is the standard $(n-1)$-sphere, whose set of n-generators is $E_n = \{x_n\}$ with

$$s_{n-1}(x_n) = x_{n-1}^- \qquad \text{and} \qquad t_{n-1}(x_n) = x_{n-1}^+$$

as source and target functions. We write $I_n : S_n \to C_{n+1}$ for the obvious inclusion functor. For example, the categories S_1 and C_2 are respectively



A *variable* X in a 2-category $\mathcal{C}$ is a 2-functor $X : S_1 \to \mathcal{C}$, with S_1 seen as a 2-category with only identities as 2-cells. Given a variable X, the 2-category $\mathcal{C}[X]$ obtained from $\mathcal{C}$ by adjoining a formal variable is defined as the pushout



We sometimes leave implicit the inclusion functor I_X in the following. A 2-cell $\alpha : f \Rightarrow g : A \to B$ in $\mathcal{C}$ can be seen as a 2-functor $\alpha : C_2 \to \mathcal{C}$ with

$$\alpha(x_2) = \alpha \qquad \alpha(x_1^-) = f \qquad \alpha(x_1^+) = g \qquad \alpha(x_0^-) = A \qquad \alpha(x_0^+) = B$$

By universality of the pushout construction, for every 2-category $\mathcal{D}$, every 2-cell $\alpha : f \Rightarrow g$ of $\mathcal{D}$ induces a bijection between functors $F : \mathcal{C}[X] \to \mathcal{D}$ such that $F \circ \tilde{X} = \alpha$ and functors $G : \mathcal{C} \to \mathcal{D}$ such that $G \circ X = \alpha \circ I_1$. In particular, given a 2-cell α of $\mathcal{C}$ such that $\alpha \circ I_1 = X$, we write

$$-[\alpha/X] \quad : \quad \mathcal{C}[X] \quad \to \quad \mathcal{C}$$

for the functor associated to the identity $\mathrm{Id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$, which is called the *substitution functor* for α. We also write N for the 2-category with only one 0-cell, one 1-cell and $\mathbb{N}$ as set of 2-cells, with both horizontal and vertical composition given by addition, and $O : \mathcal{C}[X] \to N$ the functor associated to the constant functor $0 : \mathcal{C} \to N$ (which to every 2-cell associates 0) via the 2-cell 1 of N; intuitively, the functor O counts the number of *occurrences* of the variable X in a 2-cell of $\mathcal{C}[X]$. A 2-cell α of $\mathcal{C}[X]$ is *linear* (with respect to the variable X) when $O(\alpha) = 1$. Finally, a context (16) can be equivalently defined as a 2-cell $K : f \Rightarrow g$ in $\mathcal{C}[X_1, \ldots, X_n] = \mathcal{C}[X_1] \ldots [X_n]$, which is linear in each of the variables X_i, where X_i is the variable corresponding to the pair of parallel 1-cells f and g (i.e. such that $X_i(x_1^-) = f_i$ and $X_i(x_1^+) = g_i$). Since for any permutation π of n, the category $\mathcal{C}[X_1, \ldots, X_n]$ is canonically isomorphic to $\mathcal{C}[X_{\pi(1)}, \ldots, X_{\pi(n)}]$, we extend the notation of substitution functors

$$-[\alpha/X_i] \quad : \quad \mathcal{C}[X_1, \ldots, X_n] \quad \to \quad \mathcal{C}[X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n]$$

to the functors induced by the substitution functors

$$\mathcal{C}[X_1, ., X_n] \cong \mathcal{C}[X_1, ., X_{i-1}, X_{i+1}, ., X_n, X_i] \xrightarrow{-[\alpha/X_i]} \mathcal{C}[X_1, ., X_{i-1}, X_{i+1}, ., X_n]$$

Notice that these operations "commute" in the sense that

$$-[\alpha/X_i][\beta/X_j] \quad = \quad -[\beta/X_j][\alpha/X_i]$$

for $i \neq j$. Suppose that we are given contexts

$$K_i \quad : \quad f_i^1 \Rightarrow g_i^1, \ldots, f_i^{k_i} \Rightarrow g_i^{k_i} \quad \Rrightarrow \quad f_i \Rightarrow g_i$$

for $1 \leq i \leq n$ and write Y_i^j for the variable corresponding to the pair of parallel 1-cells f_i^j and g_i^j, i.e. K_i is a 2-cell in $\mathcal{C}[Y_i^1, \ldots, Y_i^{k_i}]$. Composition of contexts is given in this setting by

$$K \circ (K_1, \ldots, K_n) \quad = \quad K[K_1/X_1] \ldots [K_n/X_n]$$

3.2 Critical pairs

The previously introduced notions enable us to give a definition of critical pair in the setting of 3-polygraphs.

Definition 24 (Unifier)**.** A *unifier* of a two 2-cells

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

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

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

such that $K_1(\alpha_1) = K_2(\alpha_2)$. By extension, an unifier of two 3-generators

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

in a 3-polygraph S, is an unifier of their sources α_1 and α_2.

Definition 25 (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 an 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 those, which is

- *non-trivial*: there are no contexts

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

  such that

  $$K_1 = K_\circ(K_1', K_{K_2'(\alpha_2)}) \quad \text{and} \quad K_2 = K_\circ(K_{K_1'(\alpha_1)}, K_2')$$

  and the property is also satisfied when exchanging the roles of K_1 and K_2,
- *minimal*: for every unifier (K_1', K_2') of (r_1, r_2) such that $K_1 = K_1'' \circ K_1'$ and $K_2 = K_2'' \circ K_2'$, the unary contexts K_1'' and K_2'' are invertible contexts.

By abuse of language, we sometimes say that a 2-cell α is a critical pair to mean that there exists a critical pair (K_1, r_1, K_2, r_2) as in the above definition, which is clear from the context, such that $\alpha = K_1(\alpha_1) = K_2(\alpha_2)$.

Remark 26. In the definition of the non-triviality condition, it would be also tempting to require that there are no contexts

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

with $f_1' \otimes f_2' = f$ and $g_1' \otimes g_2' = g$, such that

$$K_1 = K_\otimes(K_1', K_{K_2'(\alpha_2)}) \quad \text{and} \quad K_2 = K_\otimes(K_{K_1'(\alpha_1)}, K_2')$$

However, this situation is already handled by the non-triviality condition since for example we would have in this case

$$K_1 = K_\circ(K_\otimes(K_1', K_{f_2'}), K_\otimes(K_{g_2'}, K_{K_2'(\alpha_2)}))$$

and

$$K_2 = K_\circ(K_\otimes(K_{K_1'(\alpha_1)}, K_{f_2'}), K_\otimes(K_{g_2'}, K_2'))$$

We have seen in Section 2.4.3 that the morphisms of the free 2-category generated by a polygraph S can be seen as 2-nets, i.e. 2-polygraphs over S, allowing us to consider morphisms of polygraphs between nets. In particular, the 2-nets β of the form $\beta = K(\alpha)$ can be seen as 2-nets β together with a monomorphism of 2-polygraphs $i : \alpha \Rrightarrow \beta$. This allows us to give a more concrete equivalent definition of critical pairs.

Definition 27 (Critical pair)**.** A *critical pair* $(r_1, i_2, \alpha, r_2, i_2)$ in a 3-polygraph S consists of two 3-generators

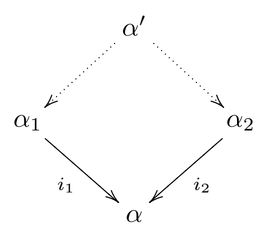
$$r_1 : \alpha_1 \Rrightarrow \beta_1 : f_1 \Rightarrow g_1 \quad \text{and} \quad r_2 : \alpha_2 \Rrightarrow \beta_2 : f_2 \Rightarrow g_2$$

a 2-net $\alpha : f \Rightarrow g$ and two monomorphisms of polygraphs

$$i_1 : \alpha_1 \Rrightarrow \alpha \quad \text{and} \quad i_2 : \alpha_2 \Rrightarrow \alpha$$

called *injections*, which are

– *non-trivial*: the 2-net α' defined as the pullback



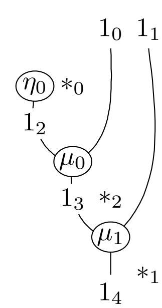
contains at least one 2-generator.

– *minimal*: the morphism $i_1 + i_2 : \alpha_1 + \alpha_2 \Rrightarrow \alpha$ is epi.

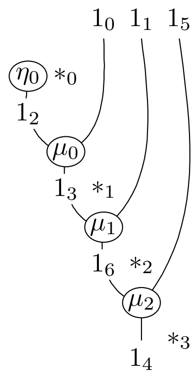
The usual property of critical pairs, which really justifies their name, extends to our framework:

Property 28. 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.

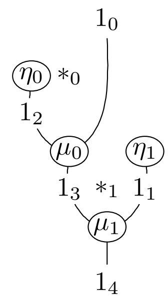
Example 29. For example consider the theory of monoids given in Example 8. The net



is an unifier of the rules a and l. However the nets

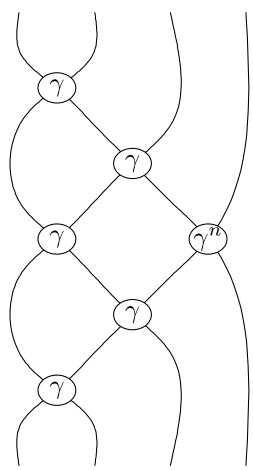


and



are not because they are respectively trivial and not minimal (with the obvious injections).

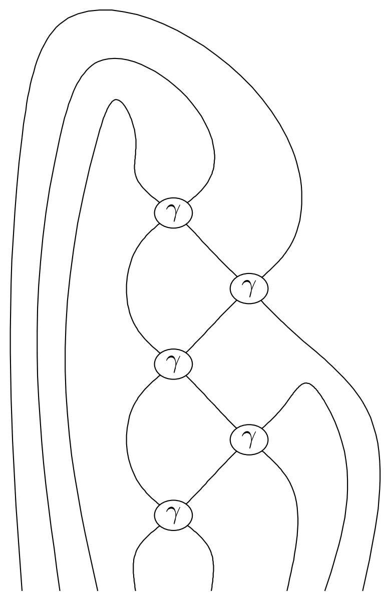
Example 30. Consider the 3-polygraph S of symmetries (Example 9). We write $\gamma^n : 1 \otimes 1 \to 1 \otimes 1$ for the morphism defined by induction on the integer n by $\gamma^0 = 1 \otimes 1$ and $\gamma^{n+1} = \gamma \circ \gamma^n$. Then for every integer n, the morphism



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.

4 Free compact 2-categories

The Example 30 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 right of Figure 2. This can be done by formally adding adjoints to the 2-categories generated by 2-polygraphs. Graphically, this corresponds to adding the possibility of "bending" wires: with this possibility, the mentioned diagram can be represented as



This section is devoted to the formalization of this idea.

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 31 (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

$$A \underset{g}{\overset{f}{\rightleftarrows}} B \quad (\bot)$$

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

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

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 32 (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,

$$(g \circ f)^{-1} = f^{-1} \circ g^{-1} \quad \text{and} \quad (g \circ f)^{+1} = f^{+1} \circ g^{+1}$$

and

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

and

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

– for every 0-cell A,

$$A^{-1} = A = A^{+1}$$

and

$$\eta^+_A = A = \varepsilon^+_A \quad \text{and} \quad \eta^-_A = A = \varepsilon^-_A$$

– 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

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

By abuse of notation, for any 1-cell $f : A \to B$ in a strictly compact 2-category and integer n, we write f^n for 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 which 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]. We recall briefly this construction here.

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

Definition 33 (Category with formal adjoints)**.** A category with formal adjoints $(\mathcal{C}, (-)^{-1}, (-)^{+1})$ is a category together with two functors

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

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,

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

– 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:

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

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

– $\varepsilon_{f^n} : f^n \otimes f^{n-1} \to A$, 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,

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

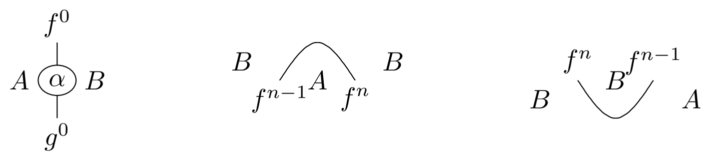
– for every pair of horizontally composable 2-cells α^0 and β^0,

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

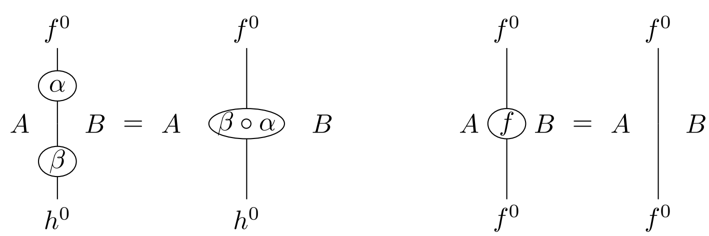
– for every 1-cell f^n,

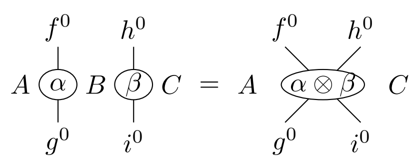
$$(f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^n) = f^{n-1} \quad \text{and} \quad (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) = f^n$$

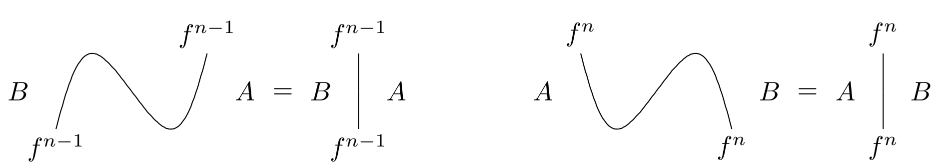
Graphically, if we write respectively



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 particular, if $\mathcal{C}$ is the 2-category generated by a 2-polygraph P, the compact 2-category $\mathcal{A}(\mathcal{C})$ is generated by the 3-polygraph Q such that

– $E_0^Q = E_0^P$

– $E_1^Q = \{ f^n \mid f \in E_1^P, n \in \mathbb{Z} \}$

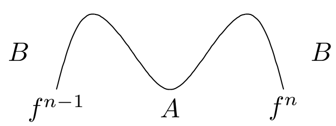
– $E_2^Q = \{\ \alpha^0 \mid \alpha \in E_2^P\ \} \uplus \{\ \eta_{f^n}, \varepsilon_{f^n} \mid f^n \in E_1^Q\ \}$

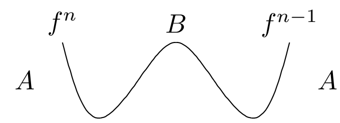
– $E_3^Q = \{\ l_{f^n}, r_{f^n} \mid f^n \in E_1^Q\ \}$

where

$$\begin{array}{rlrl} s_2(l_{f^n}) = & (f^{n-1} \otimes \varepsilon_{f^n}) \circ (\eta_{f^n} \otimes f^n) & t_2(l_{f^n}) = & f^{n-1} \\ s_2(r_{f^n}) = & (\varepsilon_{f^n} \otimes f^n) \circ (f^n \otimes \eta_{f^n}) & t_2(r_{f^n}) = & f^n \end{array}$$

and other cells have the obvious source and target. By Lemma 21, the polygraph Q is terminating and by Lemma 20 it is confluent since all its critical pairs, which are of the form



and


for some 1-cell f^n, are joinable. If $f_1, \ldots, f_m$ and $g_1, \ldots, g_n$ are parallel lists of composable morphisms of $\mathcal{C}$, then it can easily be shown that the 2-cells

$$\alpha : f_1^0 \otimes \ldots \otimes f_m^0 \Rightarrow g_1^0 \otimes \ldots \otimes g_n^0 \tag{17}$$

in the 3-category generated by Q, which are normal forms with respect to the previous rewriting system, do not contain any 2-generator η_{f^n} or ε_{f^n}. From this, we can deduce that the 2-cells (17) in bijection with the 2-cells

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

of $\mathcal{C}$, which shows that the embedding of $\mathcal{C}$ into $\mathcal{A}(\mathcal{C})$ is full and faithful. Moreover, the argument can easily be generalized to any category $\mathcal{C}$ (not necessarily generated by a 2-polygraph):

Property 34. 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" $\mathcal{A}(\mathcal{C})$.

Remark 35. The string diagrams with winding numbers, that we use here to represent morphisms in compact 2-categories, were studied and formalized by Joyal and Street [JS98, Sel08].

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.

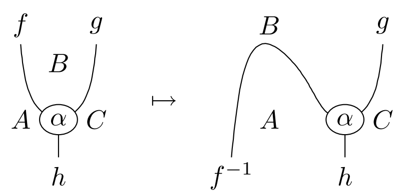
Property 36. If $\mathcal{C}$ is a compact 2-category, the sets

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

are naturally isomorphic by the function

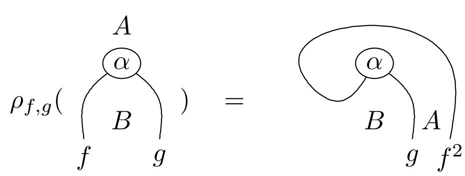
$$\alpha \quad \mapsto \quad (f^{-1} \otimes \alpha) \circ (\eta_f \otimes g)$$

Graphically,



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 36 and called *rotation*, between $\mathrm{Hom}(A, f \otimes g)$ and $\mathrm{Hom}(B, g \otimes f^2)$. Graphically,



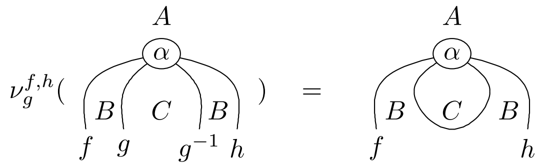
(we sometimes simply write ρ_f when g is clear from the context). We also write

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

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 $\mathrm{id}_f : f^{-1} \otimes f$, 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 $\otimes_B : R(f) \times R(g) \to R(f \otimes g)$ 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 37. 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.

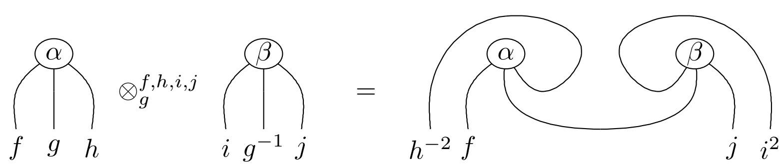
Remark 38. Suppose that $\mathcal{C}$ is a rotative 2-category. To every pair of two cells

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

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 $h : B \to A$,

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

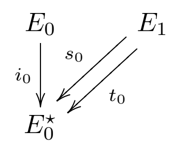
and moreover, we could have equivalently formalized the notion of rotative category using only this generalized composition operation instead of rotation, composition and hiding. It is more concise, which is why we use it in the following, but leads to a less nice axiomatics.

Remark 39. 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 23 (its objects are the 1-cells of $\mathcal{R}$). 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

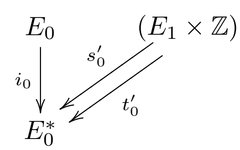
A *compact 2-polygraph* can be defined as in Section 2.1, where E_1^* (and s_0^* and t_0^*) is generated by a free category with formal adjoints construction on the graph (5). However, since compact 2-categories are equivalent to rotative 2-categories, it will prove simpler to define a compact 2-polygraph as follows.

A *compact 1-polygraph* is a diagram



in **Set**, where $E_0^\star = E_0$ 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 $E_1^\star$ for the set of its morphisms, $i_1 : E_1 \to E_1^\star$ 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 $t_0^\star \circ i_1 = t_0$.

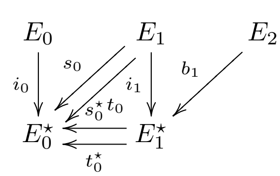
Remark 40. The category (with formal duals) thus generated is isomorphic to the category generated by the polygraph



with

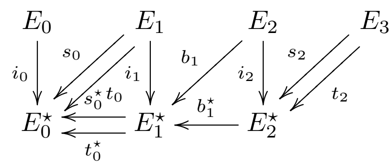
$$(s'_0(f,n), t'_0(f,n)) = \begin{cases} (s_0(f), t_0(f)) & \text{if } n \text{ is even,} \\ (t_0(f), s_0(f)) & \text{if } n \text{ is odd.} \end{cases}$$

A *compact 2-polygraph* is a diagram



(18)

in **Set**, consisting of a compact 1-polygraph together with the category with formal duals it generates, along with a set E_2 and a function $b_1 : E_2 \to E_1^\star$ 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 $E_2^\star$. We write $i_2 : E_2 \to E_2^\star$ 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



in **Set**, consisting of a compact 2-polygraph together with the rotative 2-category it generates, along with a set E_3 and two functions $s_2, t_2 : E_3 \to E_2^\star$ such that $s_2 \circ b_1^\star = t_2 \circ b_1^\star$.

We write n-**cPol** 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 categorical nets

The construction of categorical nets can be adapted to the setting of rotative 2-categories (and compact 2-categories). As explained in Remark 39, a multicategory of contexts of a compact 2-category can be defined and it can be constructed concretely using categorical nets. We give here a variant of this construction which is suitable for 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 (18) 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

$$p \quad = \quad x_0, (y_1, w_1), x_1, (y_2, w_2), x_2 \ldots, x_{n-1}, (y_n, w_n), x_n \tag{19}$$

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 E_0^S and to every y_i associates an element $\tau_p(x_i)$ of E_1^S, 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 $E_0^p = \{x_i\}$ and $E_1^p = \{y_i\}$ 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, which is defined in the obvious way. 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 (19), 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 (19) 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 $0 < i < n$,

$$x_i \neq x_{i+1} \quad \text{and} \quad y_i \neq y_{i+1}$$

Nets. A *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* $\tau_N(z) \in E_2^S$ 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 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

$$E_0^{p_1} \cap E_0^{p_3} \qquad E_0^{p_1} \cap E_0^{p_4} \qquad E_0^{p_2} \cap E_0^{p_3} \qquad E_0^{p_2} \cap E_0^{p_4}$$

are all included in E_0^p and the sets

$$E_1^{p_1} \cap E_1^{p_3} \qquad E_1^{p_1} \cap E_1^{p_4} \qquad E_1^{p_2} \cap E_1^{p_3} \qquad E_1^{p_2} \cap E_1^{p_4}$$

are all included in E_1^p (i.e. all unbound 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 $E_k^{b_i} \cap E_k^b$ for some index i and *unbound* otherwise. Suppose that we are moreover given n operations

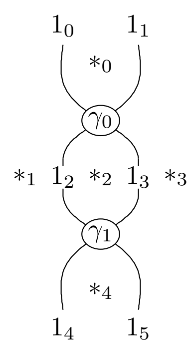
$$K_i \quad : \quad b_1^1, \ldots, b_n^{k_i} \quad \Rrightarrow \quad b_i$$

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 injective 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. Two operations borders b and b' are *α-equivalent* when there exists a 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 a 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 41. Consider the signature of symmetries defined in the Example 9 (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

$$E_2^N = \{\gamma_0, \gamma_1\} \qquad E_1^N = \{1_0, 1_1, \ldots, 1_5\} \qquad E_0^N = \{*_0, *_1, \ldots, *_4\}$$

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 $\mathcal{K}_S$:

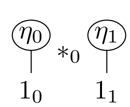
$$N \;:\; \quad \Rrightarrow \quad *_3, (1_1, -1), *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3 \tag{20}$$

$$N \;:\; \quad \Rrightarrow \quad *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3, (1_1, 1), *_0 \tag{21}$$

$$N \;:\; *_2 \quad \Rrightarrow \quad *_3, (1_1, -1), *_0, (1_0, -1), *_1, (1_4, 0), *_4, (1_5, 0), *_3 \tag{22}$$

Remark 42. 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 43. 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 8. 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 E_2^N for the set of 2-cells of N, and E_1^N and E_0^N 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 $E_i^M \cap E_i^N = \emptyset$, with $i = 0$, 1 or 2. A net M is *included* in a net N, what we write $M \subseteq N$, when $E_i^M \subseteq E_i^N$, for every cell $x \in E_i^M$, $\tau_M(x) = \tau_N(x)$, with $i = 0$, 1 or 2, and for every 2-cell $z \in E_2^M$, $b_M(z) = b_N(z)$.

Connected nets. Two 2-cells of a net N are immediately connected whenever they share 1-generators on their border, i.e. $E_1^{b_N(z_1)} \cap E_1^{b_N(z_2)} \neq \emptyset$. 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 41, the net (20) is not connected but the net (22) is.

4.5.2 Operations on nets

Mergings. A *merging* is a function m which to each i-generator x associates an i-generator $m(x)$, for $i = 0$, 1 or 2. The *domain* of a merging m is the set of generators x such that $m(x) \neq x$. A merging is *finite* when its domain is finite. It can be checked that we only use finite mergings in the following (so these are easy to represent and manipulate with a computer). We usually only specify the image of the elements of the domain of a merging m.

We sometimes write $x \mapsto x'$ for the merging m which is the identity excepting on x where $m(x) = x'$. Obviously, merging extend as a morphism on borders, nets, etc. and we write for example $m(N)$ for the image of a net N under merging m.

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 path 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

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

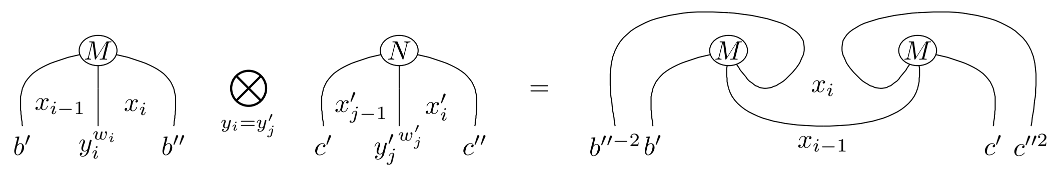
with

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

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

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

Graphically, this corresponds to the extended form of composition of morphisms which was introduced in Remark 38:



Remark 44. 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

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

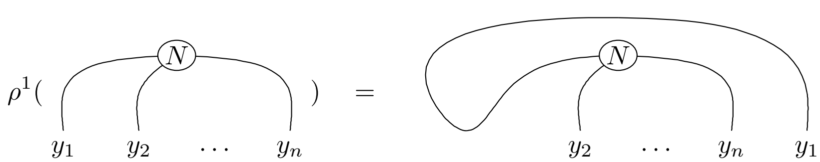
We write $\rho^1(N) : b_1, \dots b_n \Rrightarrow b'$ for the net N (in which only the border has been changed) with

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

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 $\rho^n_i(N)$ 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

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

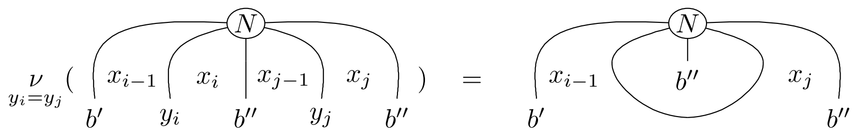
with $w_j = w_i - 1$. We write

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

for the image of the net under the merging defined by

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

Graphically,



Remark 45. 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 46. 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 we wanted 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 27 to this framework (we see the two nets as left members of two rewriting rules).

Definition 47 (Critical pair)**.** A *critical pair* between two nets

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

is a net

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

satisfying

- *inclusion*:

$$M \subseteq P \quad \text{ and } \quad N \subseteq P$$

- *non-triviality*:

$$E_2^M \cap E_2^N \neq \emptyset$$

– *minimality*: for $i = 0$, 1 or 2,

$$E_i^M \cup E_i^N = E_i^P$$

Remark 48. 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

5.1 An 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 $\text{father}_N(y)$ (resp. $\text{son}_N(y)$) for the 2-cell z such that y occurs in its border as an output (resp. as an input); by convention we write $\text{father}_N(y) = \bot$ (resp. $\text{son}_N(y) = \bot$) 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 $\text{father}_N(y) = \bot$ (resp. $\text{son}_N(y) = \bot$) 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 fresh_atomic(α) returns an atomic 2-net (in a sense similar to the definition of Section 2.4.3) whose only 2-generator is of type α and whose generators are fresh.

– The function border_index(b, y) gives the index of a 2-generator y in a border b.

– The function 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 merging 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 merging m such that $m(M) = \nu_{y_1=y_2}(M)$ described in Section 4.5.2.

5.1.2 The algorithm

Suppose that we are given two connected nets

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

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*

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

of the algorithm is a tuple of lists such that the elements of S_i and U_i are elements of $E_i^N \times E_i^P$, 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 merging m, we write $S[m]$ for the for the state where every generator x has been replaced by $m(x)$, 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

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

where (z_1, z_2) is an 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
   $(P : b_1, \ldots, b_n \Rrightarrow b), m := P \otimes_{y_2 = y'_2} A$
   $S := S[m]$
   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 $\mathrm{winding}(e, y'_2) \neq \mathrm{winding}(e, y_2) - 1$ then *Failure*
   let $m = \mathrm{merging}(P, y_2, y'_2)$ in
   $S := S[m]$
   if $\mathrm{son}_N(y_1) \neq \bot$ 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 an 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 49. 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).

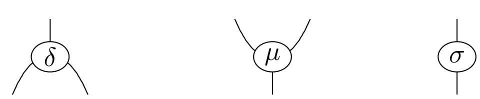
5.1.3 An example

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

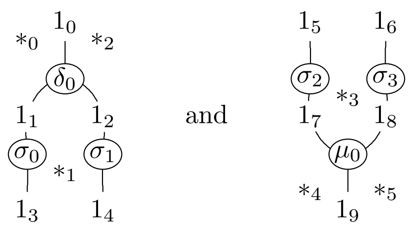
- one 0-generator $*$,
- one 1-generator 1,
- three 2-generators

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

  respectively depicted as

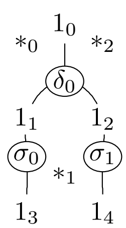


Suppose moreover that we want to unify two morphisms corresponding respectively to the nets M and N whose graphical representation are respectively

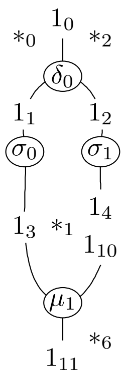


We now describe the main steps during the execution of the algorithm starting from the unification position $\sigma_0 \stackrel{?}{=} \sigma_2$. We write D-1 as a short notation for the rule Duplicate-1, etc.

We start from the "unifier" P, pictured as

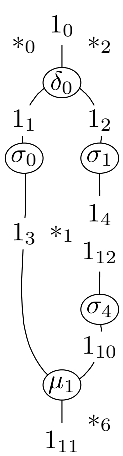


which is equal to M. There is a trivial injection $i_M : M \Rrightarrow P$ (the identity) and our algorithm will "grow" it until there is also an injection $i_N : N \Rrightarrow P$. The rule T-2 first checks that the types of σ_0 and σ_2 coincide, which is the case (they are both equal to σ). Then, the 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



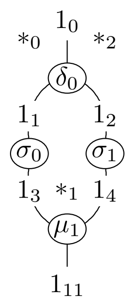
together with the unification target $\mu_0 \stackrel{?}{=} \mu_1$. By P-2, this leads to setting $i_N(\mu_0) = \mu_1$ and creates two new unification targets $1_9 \stackrel{?}{=} 1_{11}$ and $1_8 \stackrel{?}{=} 1_{10}$. The first unification target will eventually lead to setting $i_N(1_9) = 1_{11}$ 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 $\text{father}_N(1_8) = \sigma_3$ and $\text{father}_P(1_{10}) = \bot$:

1. A fresh atomic 2-net of type σ is adjoined to P which becomes



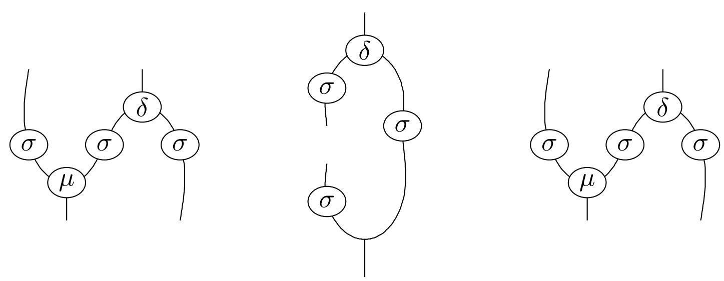
   and the unification target remains $i_N(1_8) = 1_{10}$. By the P-i rules, this eventually leads to setting $i_N(1_8) = 1_{10}$, $i_N(\sigma_3) = \sigma_4$ and $i_N(1_6) = 1_{12}$. And the P thus computed is an unifier of M and N.

2. The "unifier" P becomes $\nu_{1_4=1_{10}}(P)$, that is



   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 an unifier of M and N.

The other unifiers of M and N are

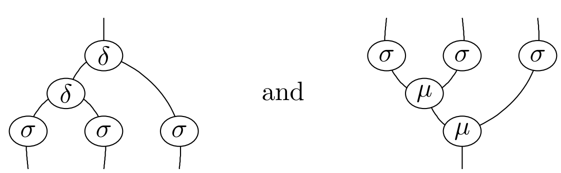


and are computed by starting the algorithm on the unification positions $\sigma_3 \stackrel{?}{=} \sigma_0$, $\sigma_3 \stackrel{?}{=} \sigma_1$ and $\sigma_2 \stackrel{?}{=} \sigma_1$ respectively.

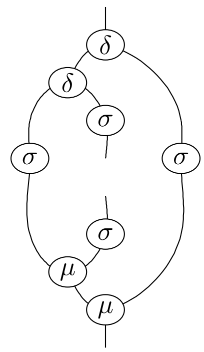
5.1.4 Remarks

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

The need of the operadic structure. The example given in Section 5.1.3 is simple enough not to really need the operadic structure. However, consider the two following morphisms M and N in the same signature:



An unifier of these two morphisms is



which really requires the operadic structure in order to describe the "hole" in it.

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.

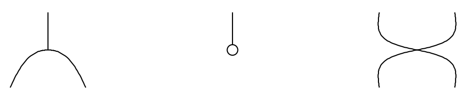
Confluence on metaterms. As explained before, in order to compute the unifier of two morphisms in a 2-category $\mathcal{C}$, we embed this category into a "bigger universe" – the free compact 2-category $\mathcal{A}(\mathcal{C})$ – and compute the critical pairs in $\mathcal{A}(\mathcal{C})$. The formal 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 $\mathcal{A}(\mathcal{C})$, 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 9. It can be shown to be confluent [Laf03], however the critical pair shown on the right of Figure 2 is not joinable in the free compact 2-category (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.

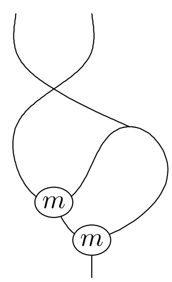
Recovering the usual unification algorithm. Burroni has shown [Bur93] that there is a forgetful functor U from equational term-theories to polygraphic theories, which to equational theory on a term-signature (Σ_n) associates a polygraph S containing: one 0-generator $*$, one 1-generator $1 : * \to *$,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

$$\delta : 1 \to 2 \quad \text{and} \quad \varepsilon : 1 \to 0 \quad \text{and} \quad \gamma : 2 \to 2 \tag{23}$$

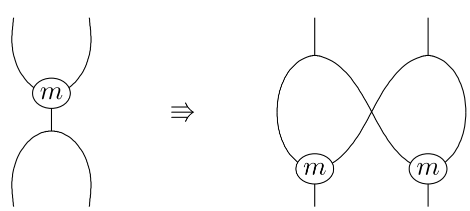
(which should be seen as explicit duplication, erasure and swapping of variables) with the relations corresponding to the relations of the equational term-theory, equations expressing that the generators (23) satisfy the laws of commutative comonoids, and equations expressing the compatibility of the generators (23) with operations coming from the (Σ_n). The generators (23) are usually respectively pictured as



Example 50. 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 categorical 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

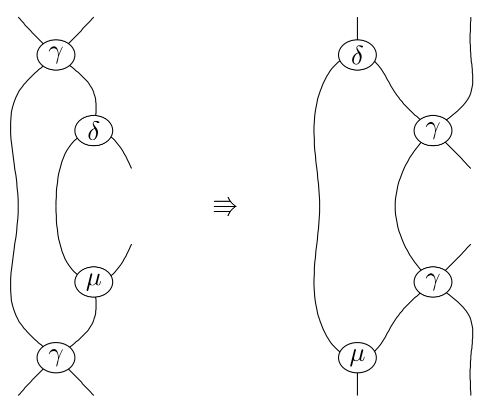
We have made a toy implementation of the algorithm described in Section 5.1.2 in less than 2000 lines of OCaml. It has been used to successfully recover the unifiers of many rewriting systems defined in [Laf03] and, even though we did not particularly focus on efficiency, computing the unifiers themselves is very fast (typically less than a second on a desktop computer) and negligible compared to the compilation time of the produced LaTeX file (which contains graphical representations of the unifiers).

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, 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.

[GM08] Y. Guiraud and P. Malbos. Higher-dimensional categories with finite derivation type. Arxiv preprint `arXiv:0810.1442`, 2008.

[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.

[JS98] A. Joyal and R. Street. Planar Diagrams and Tensor Algebra. Unpublished manuscript, available on Ross Street's website, 1998.

[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.