

Coherent Interaction Graphs

Lê Thành Dũng NGUYỄN^{*†}

Laboratoire d’informatique de Paris Nord (LIPN)
Université Paris 13, Sorbonne Paris Cité, Villetaneuse, France
nltld@nguyentito.eu

Thomas SEILLER[†]

CNRS, UMR 7030 – LIPN
CNRS, UMR 7030 – LIPN
seiller@lipn.fr

We introduce the notion of coherent graphs, and show how those can be used to define dynamic semantics for Multiplicative Linear Logic (MLL) extended with non-determinism. Thanks to the use of a coherence relation rather than mere formal sums of non-deterministic possibilities, our model enjoys some finiteness and sparsity properties.

We also show how studying the semantic types generated by a single “test” within our model of MLL naturally gives rise to a notion of proof net, which turns out to be exactly Retoré’s R&B-cographs. This revisits the old idea that multiplicative proof net correctness is interactive, with a twist: types are characterized not by a set of counter-proofs but by a single non-deterministic counter-proof.

1 Introduction

Dynamic semantics of proofs originate in Girard’s *Geometry of Interaction (GoI) program* [7], whose aim was to provide a semantic account for the process of cut-elimination. Indeed, while the proofs-as-programs correspondence expresses that β -reduction of lambda-terms corresponds to cut-elimination, its extension to categorical models (the so-called Curry–Howard–Lambek correspondence) arguably fails to fully reflect these dynamics, as it represents those operations as a simple equality.

Since then, GoI has been developed into many directions to give various accounts of cut-elimination without syntactic rewriting. One aspect which has received less attention is the construction of models of linear logic where morphisms are generalized programs on which a GoI-style execution procedure can be defined, and serves to compose morphisms. Girard’s successive papers on GoI all investigate such models, using operator algebras as the spaces of generalized programs; recently, Seiller has simplified Girard’s models using graphings (a measure-theoretic extension of graphs) instead of operators [24]. Note that discrete graphs suffice to obtain a model of linear logic without exponentials or quantifiers [21, 23].

The aforementioned models all start from an untyped universe of programs, and interpret formulae as specifications for the behavior of programs: they can be seen as realizability models with operators/graphs as realizers. Furthermore, these specifications will be given as batteries of “tests” or “counter-proofs” which are themselves generalized programs.

This idea comes from the theory of multiplicative proof nets, where correct proofs have to be distinguished out of a set of “proof structures” by means of a *correctness criterion*. Girard observed that by representing a proof structure as a permutation σ , he could express the correctness criterion as the cyclicity of $\sigma\tau_i$ for all i , where $\{\tau_i \mid i \in I\}$ is a set of permutations depending on the formula being proved. This led to a prototypical GoI model [6], limited to Multiplicative Linear Logic (MLL), with permutations as

^{*}Part of this work was carried out when the first author was a student at École normale supérieure de Paris.

[†]Partially supported by the ANR project ELICA (ANR-14-CE25-0005).

programs. This model internalizes “the proof σ passes the test τ ” as a symmetric relation $\sigma \perp \tau$, defined as “ $\sigma\tau$ cyclic”. In general, in GoI models, semantic types are defined using this kind of symmetric relation, called an *orthogonality*.

Contents of the paper. In this paper, we revisit this connection between proof net correctness and GoI orthogonality in the MLL fragment. One issue is that, for a given MLL formula, the number of corresponding tests is exponential in the size of the formula – and by replacing the formula by a huge test of sets, we have forgotten that they all come from the same succinct data. We argue that a formula should correspond to a *non-deterministic superposition* of tests, and propose a variant of Seiller’s interaction graphs which admits *sparse* representations of non-deterministic sums of programs; the main ingredient is a *coherence relation* between edges, hence the title. Instead of an exponential family of tests, we get a single non-deterministic test of polynomial size.

Remarkably, this development – which does not depend on knowing a priori anything about MLL correctness criteria – leads us to recover a previously known but rarely used version of proof nets, introduced by Retoré [20], together with its correctness criterion. Let us note that Ehrhard suggested [3] exploring connections between this criterion and GoI. This result which provides a correspondence between cographs and formulae, suggest further work providing a tighter correspondence between proof and graph theory by exploiting the notion of *modular decomposition* of graphs (see e.g. [10]).

The coherence allows for the representation of sums of programs, and therefore provides a GoI account of MLL+SUM. This suggests extensions of the framework to e.g. Differential Linear Logic (with or without promotion) [5, 4] which requires the use of the SUM. But this is not the only possible use of a coherence relation: we will also describe in this paper how one can, in coherent interaction graphs, express a restriction to *simple* paths – i.e. paths visiting each vertex at most once – to ensure that our generalized programs remain finite objects when they are composed together. This was in fact the motivation for coherent graphs when they were originally introduced in Seiller’s PhD thesis [22].

Notations. We write $\mathcal{P}(A)$ for the powerset of A . We distinguish two notions of disjoint union: $A \sqcup B$ is $A \cup B$ with the implied condition that $A \cap B = \emptyset$, while $A + B$ is the coproduct in the category of sets (one may define $A + B = \{0\} \times A \sqcup \{1\} \times B$).

By a graph G , we mean a directed multigraph, i.e. a set of vertices $V(G)$ and a set of edges $E(G)$ with source and target maps $s_G, t_G : E(G) \rightarrow V(G)$; $s_G \times t_G$ need not be injective. We write $G \equiv G'$ when $V(G) = V(G')$ and there is a bijection $\varphi : E(G) \rightarrow E(G')$ such that $s_G = s_{G'} \circ \varphi$, $t_G = t_{G'} \circ \varphi$, that is, when there is a graph isomorphism between G and G' acting as the identity on vertices. This is the right notion of “equality” (not merely isomorphism!) of graphs for our purpose. We denote \emptyset_V the graph with no edges on the vertex set V .

A *path* is a finite sequence $v_0, e_1, v_1, \dots, e_n, v_n$ with $v_i \in V(G)$, $e_j \in E(G)$, $s_G(e_j) = v_{j-1}$, $t_G(e_j) = v_j$. Cycles are defined as paths $v_0, e_1, v_1, \dots, e_n, v_n$ with $v_n = v_0$.

Two edges e and f are *incident* if they have a common endpoint. This does not depend on the directions of e and f .

2 From execution of matchings to interaction graphs

In this section, we recall the second author’s previous work on interaction graphs [21, 23], in a particular case (empty pole), and omitting proofs.

2.1 Multiplicative Linear Logic proofs as matchings

Let F be a formula of (unit-free) MLL. Any cut-free proof of F determines a fixed-point-free involution – the σ from the introduction – on the atoms of F , exchanging two occurrences of atoms when they were introduced by the same axiom rule. It is well-known that for most purposes, one may in fact forget about the syntactic proof and work only with the involution – in fact this is one way of looking at the theory of MLL proof nets. In particular, cut-elimination can be carried out directly on involutions [18].

To do so, it is more convenient to see these fixed-point-free involutions as *matchings*, by which we mean¹ undirected graphs whose vertices all have degree 1: the vertices are the atoms, and the edges are given by the orbits of the involution. Given two cut-free proofs $\pi : (\vdash \Gamma, F)$ and $\pi' : (\vdash F^\perp, \Delta)$, inducing the respective matchings M and M' , to find the matching for $\mathbf{cut}(\pi, \pi')$, we first *plug* M and M' together:

Definition 1. Let G, H be two graphs; their vertex sets $V(G)$ and $V(H)$ might not be disjoint. $G \square H$ then denotes the graph $(V(G) \cup V(H), E(G) + E(H))$, equipped with the partition $\{E(G), E(H)\}$ of its edge set, in order to remember the provenance of edges.

A path in $G \square H$ is *alternating* if, for any two consecutive edges, exactly one of them is in $E(G)$.

Remark 2. There is an obvious notion of ternary plugging of graphs, with a tripartition of its edge set, which will also turn out to be useful later. It will be denoted $F \square G \square H$ for the graphs F, G, H .

Here, $V(M) \cap V(M') \neq \emptyset$ because we identify the vertices corresponding to atoms in F with their duals in F^\perp . Translating cut-elimination on the level of matchings just consists of following the *alternating paths* in the resulting graph $M \square M'$: see the example in Figure 1. This operation leads to the definition of a graph whose edges are exactly the alternating paths between M and M' ; the latter will be called the *execution* of M and M' , and denoted as $M :: M'$ (Definition 3).

The alternation condition can be justified through an analogy with game semantics. An involution σ of a set S can be thought of as a “strategy” over the “arena” S : if my opponent plays x , I answer with $\sigma(x)$ – in other words, I follow the incident edge of the matching and answer with the other endpoint. Then the condition reflects an alternation of moves between two players; execution is “composition” (taking $G \square H$ and following paths) and “hiding” (keeping only the symmetric difference $V(G) \triangle V(H)$).

In general, plugging two matchings results in an undirected graph whose vertices have degree 1 or 2, i.e. a disjoint union of alternating paths and cycles. When restricting to interpretations of MLL proofs, *cycles never appear*. (This corresponds to the strong normalization property for MLL proof nets.)

2.2 Interaction graphs in a nutshell

Let us now generalize by considering arbitrary *directed graphs* instead of matchings.

Definition 3. The *execution* $G :: H$ of two graphs G, H has as vertex set the symmetric difference $V(G) \triangle V(H)$ and contains, for each alternating path from u to v in $G \square H$, an edge from u to v ($u, v \in V(G :: H)$).

Execution defines a well-behaved composition for graphs thanks to this associativity property:

Proposition 4. *If $V(F) \cap V(G) \cap V(H) = \emptyset$, then $(F :: G) :: H \equiv F :: (G :: H)$.*

Remark 5. Note that this is not just an isomorphism: on both sides of the \equiv sign, the vertex set is the same, namely the union of $V(F)$, $V(G)$ and $V(H)$ minus all pairwise intersections.

¹Properly speaking, these are actually called *1-regular graphs* in graph theory. The usual meaning of “matching” (resp. “perfect matching”) is a subset of edges of a given graph such that each vertex is incident to at most (resp. exactly) one matching edge – that is, perfect matchings correspond to 1-regular *subgraphs*.

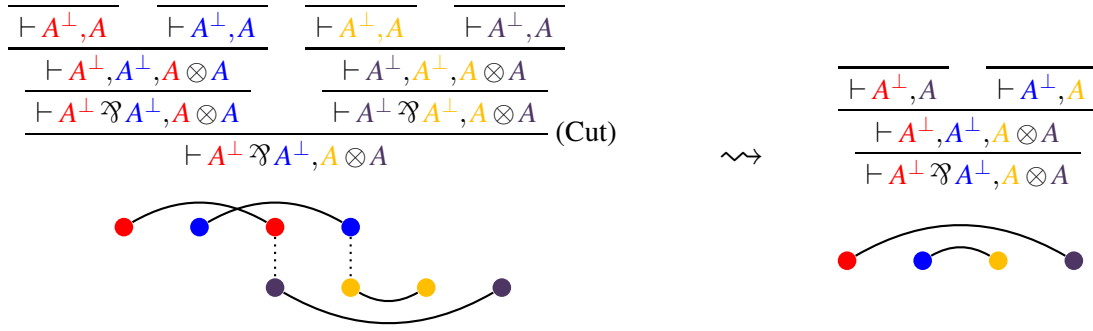


Figure 1: Cut-elimination reflected on matchings

A special case of execution allows us to represent the tensor product of MLL as an operation on interaction graphs. For two graphs G and H , if $V(G) \cap V(H) = \emptyset$, then $G :: H$ is in fact a sort of disjoint union, with all alternating paths having length 1. This defines the tensor product on graphs. (In general, to ensure $V(G) \cap V(H) = \emptyset$, we will resort to replacing G and H by isomorphic copies.)

Definition 6. For two graphs G and H such that $V(G) \cap V(H) = \emptyset$, we define $G \otimes H$ as the graph $G :: H$.

This would be enough to build a degenerate model of MLL – a compact closed category – with vertex sets as objects and graphs as morphisms. But instead, we want a model in which $\wp \neq \otimes$.

2.3 Building a model with orthogonality

This requires something akin to the correctness criterion which distinguishes \wp from \otimes in proof nets. In our case, it will be through a notion of *orthogonality*, accounting for linear negation at the level of graphs – as mentioned in the introduction (see also [21, 18]), this is related to correctness criteria. But our definition of orthogonality can also be motivated by “interactive” considerations: we saw earlier that plugging two matchings coming from MLL proofs with the right types never led to cycles.

Definition 7. For two graphs G and H with $V(G) = V(H)$, we write $G \perp H$ when $G \square H$ contains no alternating cycle. If X is a set of graphs on a common vertex set, we define $X^\perp = \{G \mid \forall H \in X, G \perp H\}$.

In our model, the objects interpreting MLL types will be sets of graphs over a common vertex set which can be written as X^\perp for some X ; they are called *conducts*². The idea is that X is a set of tests, and the conduct $\mathbf{A} = X^\perp$ is the set of programs passing those tests. The condition $\exists X. \mathbf{A} = X$ is equivalent to $\mathbf{A} = \mathbf{A}^{\perp\perp}$; thus, $(-)^{\perp}$ can be used as an involutive linear negation.

Definition 8. A *conduct* \mathbf{A} on a set $V(\mathbf{A})$ is a set of graphs with vertex set $V(\mathbf{A})$ such that $\mathbf{A} = \mathbf{A}^{\perp\perp}$.

The constructions on graphs then lift to constructions on conducts.

Definition 9. Let \mathbf{A}, \mathbf{B} be conducts with $V(\mathbf{A}) \cap V(\mathbf{B}) = \emptyset$. We define $\mathbf{A} \otimes \mathbf{B} = \{G \otimes H \mid G \in \mathbf{A}, H \in \mathbf{B}\}^{\perp\perp}$

It is then natural to define the morphisms from \mathbf{A} to \mathbf{B} as the graphs belonging to the conduct $\mathbf{A} \multimap \mathbf{B} = (\mathbf{A} \otimes \mathbf{B}^\perp)^\perp$. Unfolding this definition, one gets the realizability-style characterization of $\mathbf{A} \multimap \mathbf{B}$ as the set of graphs mapping elements of \mathbf{A} to elements of \mathbf{B} .

Proposition 10. Let \mathbf{A}, \mathbf{B} be conducts such that $V(\mathbf{A}) \cap V(\mathbf{B}) = \emptyset$.

$$\mathbf{A} \multimap \mathbf{B} = \{F \mid \forall G \in \mathbf{A}, F \perp G \otimes \emptyset_{V(\mathbf{B})} \text{ and } F :: G \in \mathbf{B}\}$$

²This word is a synonym of “behavior”, meant to suggest that a type is a set of programs behaving in the same way.

More precisely, the programs in $\mathbf{A} \multimap \mathbf{B}$ are those which send all inputs in \mathbf{A} to outputs in \mathbf{B} , with the side condition $F \perp G \otimes \emptyset_{V(\mathbf{B})}$ meaning that there must be no alternating cycle between the program and the possible inputs. This side condition also appears in the so-called adjunction between the tensor product and the execution:

Proposition 11 (Adjunction). *Let F , G and H be three graphs. Suppose that $V(F) = V(G) \sqcup V(H)$. Then $F \perp G \otimes H$ if and only if $F \perp G \otimes \emptyset_{V(H)}$ and $F :: G \perp H$.*

Together, the above two facts lead to the monoidal closure of the category of conducts and graphs. Most of the ingredients for a model of MLL are now present; the construction will be detailed when we carry it out in the setting of coherent graphs, which is the topic of the next section.

Remark 12. A way to get rid of the side condition is to keep track of the cycles appearing during the execution and incorporating them in our notion of generalized program – which would then consist of a graph together with a “set of cycles”, called a *project*. The orthogonality of two projects $a \perp b$ can then be formulated as the fact that the set of cycles of the project $a :: b$ is empty – this is a *focussed orthogonality* in the sense of Hyland and Schalk [14], and our construction can then be seen as an instance of *double glueing* carried out in a compact closed category.

The project-based construction of a model of MLL is much more general, since it allows one to define orthogonality as any reasonable predicate on the set of cycles, not just emptiness. This leads to a family of models parameterized by the choice of orthogonality. The interested reader can find details of this more general construction in earlier papers by the second author [21, 23]. For reasons of simplicity, and because the selected results we will detail in the last section do not require this more general setting, we will not use projects in this paper and restrict to the specific case described above. It should however be noted that the extension from graphs to coherent graphs explained in the next section could very well be performed in the more general setting of projects.

3 Enriching interaction graphs with coherence

We are now in a position to define the main object of study of this paper, by endowing the edges of interaction graphs with a coherence space structure. The definition of execution and orthogonality will be extended to these objects, and this will suffice to build a model of MLL from them.

3.1 Motivation: sparse non-deterministic proofs

Let’s say we want to interpret non-deterministic superpositions of proofs. More formally, this corresponds to proofs in a sequent calculus enriched with the SUM rule

$$\frac{\vdash \Gamma \quad \dots \quad \vdash \Gamma}{\vdash \Gamma}$$

An obvious solution would be to represent the set of possibilities as a formal sum of proofs – as is done for instance in differential linear logic [5, 4]. Indeed, in [17], which introduces this rule, proof equivalence is extended with associativity and distributivity over arbitrary contexts of SUM [17, Fig. 6]. These rules can turn any proof using SUM into an equivalent proof with a single SUM rule at the root.

However, this may lead to an exponential blow-up of the size of the proof. (Maurel’s motivation for studying the SUM rule in [17] is to characterize the complexity class NP; for this purpose, such a size increase must be avoided.) We would like to have concise interpretations of proofs in which small sub-proofs involve non-deterministic choice, without having to duplicate the context.

The same issues arise in the theory of proof nets for Multiplicative-Additive Linear Logic: like the SUM rule, the &-introduction rule involves a superposition of two proofs. Hughes and van Glabbeek's slice nets [12] use the "formal sum" approach, and therefore, a sequent calculus proof may have an exponentially bigger translation. In other systems of MALL proof nets, such as monomial nets [8] and conflict nets [11], this is avoided by morally sharing parts of the net between different terms of the formal sum, by means of some kind of *coherence relation*. (For the former, the appendix [8, A.1] provides an alternative presentation, equivalent to monomial weights, using a coherence space; in the latter, the conflict relation corresponds to what we call incoherence.)

Taking inspiration from this, we will extend interaction graphs to equip their sets of edges with a coherence space structure. This will allow us to interpret the SUM rule as a "incoherent union" of graphs in an interaction graph model. However, it will not be sufficient to get a model of MALL, because of obstructions³ which are common to GoI models.

Nonetheless, the consideration of this coherence space structure provides an interesting construction which could be used to built GoI models of extensions of MLL+SUM, such as differential linear logic. Moreover, the introduction of coherence allows for restrictions on the execution of two graphs that ensures finiteness of the result. Indeed in the general construction on graphs considered above, the execution of two finite graphs F, G may not be finite. However, the execution of two finite coherent graphs will be a finite coherent graph.

3.2 Coherent graphs: basic properties

Definition 13. A *coherence relation* is a symmetric reflexive binary relation. A *coherence space* X is a set $|X|$ (the *web* of X) equipped with a coherence relation written \circ_X (or \circ when X is implied by the context). We use the following notations: $x \frown y : \Leftrightarrow x \circ y \wedge x \neq y$, and $x \smile y : \Leftrightarrow x \not\circ y \wedge x \neq y$.

The following binary operations are defined on coherence spaces A, B :

- $\&$: $|A \& B| = |A| \sqcup |B|$, $a \frown_{A \& B} b$ for all $a \in A, b \in B$, and the rest is induced from \circ_A, \circ_B
- \oplus : $|A \oplus B| = |A| \sqcup |B|$, $a \smile_{A \oplus B} b$ for all $a \in A, b \in B$ and the rest is induced from \circ_A, \circ_B

Lastly, a *clique* is a subset of the web whose elements are pairwise coherent.

Definition 14. A *coherent graph* is a graph G together with a coherence relation \circ_G on its set of edges $E(G)$. We denote by $\text{Coh}(G)$ the coherence space $(E(G), \circ_G)$ associated with G .

Definition 15. Let G and H be coherent graphs. The plugging $G \square H$ is endowed with the coherence relation on edges such that $\text{Coh}(G \square H) \simeq \text{Coh}(G) \& \text{Coh}(H)$.

An alternating path (or cycle) in $G \square H$ is *coherent* if all its edges are pairwise coherent, i.e. if it is a clique for the associated coherence relation. Two coherent paths (or cycles) π, π' are *mutually coherent* if and only all edges in π are coherent with all edges in π' .

The *execution* $G :: H$ has as vertex set $V(G) \triangle V(H)$ and contains, for each *coherent alternating path* from u to v in $G \square H$ with $u, v \in V(G) \triangle V(H)$, an edge from u to v . It comes with the coherent graph structure induced by mutual coherence of paths in $G \square H$.

³The natural encoding of additives would be:

- for $A \oplus B$, $\text{inl}(G) = G \otimes \emptyset_B$ and $\text{inr}(H) = \emptyset_A \otimes H$,
- for $A \& B$, $G \& H = (G \otimes \emptyset_B) + (\emptyset_A \otimes H)$: a proof of $A \& B$ consists of a superposition of a proof of A and a proof of B .

But consider two linear maps $f : A \multimap C$ and $g : B \multimap C$, their copairing $\langle f, g \rangle : A \oplus B \multimap C$, and some $a : A$. Then, in general, the interpretation of $f(a)$ and that of $\langle f, g \rangle(\text{inl}(a))$ differ: the latter contains any edge of g whose endpoints are both in C .

This also occurs with the representation of additives by means of *sliced* interaction graphs; see the discussion in [23, §5.2] for more details and a concrete example, which can be translated to coherent graphs.

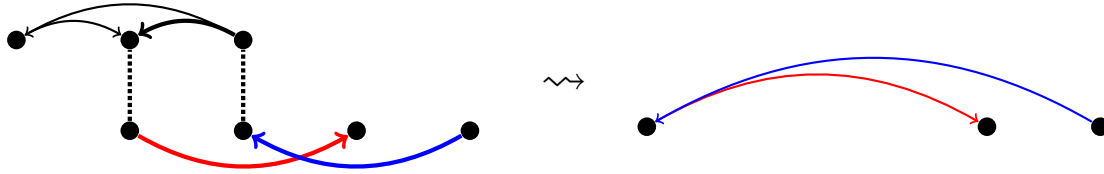


Figure 2: Execution of two coherent graphs

When $V(G) \cap V(H) = \emptyset$, we write $G \otimes H$ instead of $G :: H$. This is justified by the shape of $G \otimes H$: $V(G \otimes H) = V(G) \sqcup V(H)$ and $\text{Coh}(G \otimes H) \simeq \text{Coh}(G) \& \text{Coh}(H)$.

Example 16. Consider the two coherent graphs on the left of Figure 2. The color scheme is meant to show the cliques in the coherence relation: while all the edges of the top graph are coherent, the blue edge and the red edge in the bottom graph are incoherent. Plugging them together gives a graph with: **red** \circ black, **blue** \circ black, **red** \smile **blue**.

The execution is then computed by following the coherent alternating paths in this plugging, giving the two edges of the graph on the right. Note that the bolded path on the left is incoherent – it contains both a red and a blue edge – and therefore does not yield an edge in the execution.

Since the two coherent paths in the plugging are mutually incoherent, the execution consists of two incoherent edges, which have been drawn with different colors to illustrate this.

This definition of execution is still adequate to compose morphisms:

Proposition 17 (Associativity for coherent graphs). *Let F , G and H be three coherent graphs. Suppose that $V(F) \cap V(G) \cap V(H) = \emptyset$. Then $(F :: G) :: H \equiv F :: (G :: H)$.*

Proof. Both sides of the \equiv sign are ways of computing the graph whose vertices are $V(F) \triangle V(G) \triangle V(H)$, and whose edges correspond to the coherent alternating paths in $F \square G \square H$, with the coherence relation given by mutual coherence of paths. \square

And the operation mentioned earlier to represent non-determinism can now be formalized as follows:

Definition 18. The *incoherent sum* $G \boxplus H$ of two coherent graphs G and H with $V(G) = V(H) = V$ is defined as $V(G \boxplus H) = V$, $\text{Coh}(G \boxplus H) = \text{Coh}(G) \oplus \text{Coh}(H)$ (therefore $E(G \boxplus H) = E(G) + E(H)$).

For example, the bottom left graph of Figure 2 can be seen as the incoherent sum of the red subgraph and the blue subgraph – both subgraphs having 4 vertices. Likewise, the graph on the right can be decomposed into an incoherent sum.

3.3 An application of coherence: simple paths

Although we haven't yet redefined orthogonality and conducts for coherent graphs, what we have seen until now suffices to showcase an application of the coherence space structure on edges, initially considered in Seiller's PhD thesis [22].

The motivation for the extension of graphs with a notion of coherence was to limit the execution, which often led to infinite graphs. Indeed, let F and G be two graphs, such that $F \square G$ contains an alternating path of the form $\pi C \pi'$, where C is an alternating cycle and π, π' are two alternating paths, with endpoints in $V(F) \triangle V(G)$. Then the execution $F :: G$ contains infinitely many edges, even when F and G are finite, since all the $\pi C^n \pi'$, $n \in \mathbb{N}$, are alternating paths.

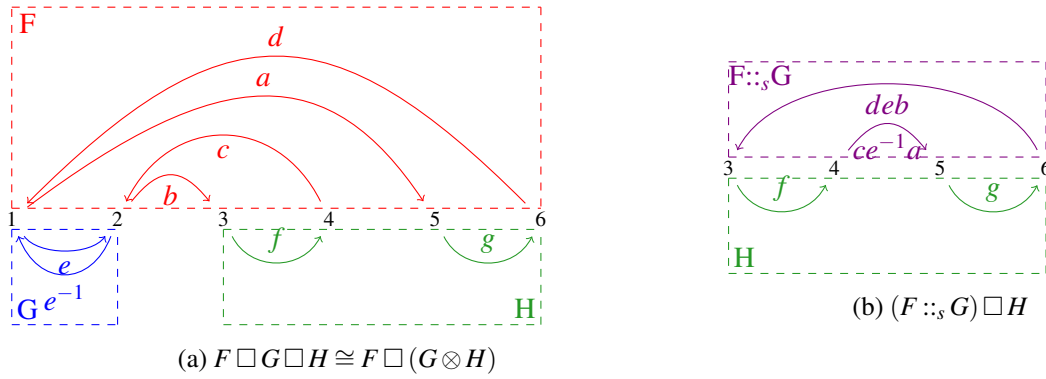


Figure 3: Counter-example of the adjunction for simple paths and cycles

In the model defined in section 2, this doesn't happen when composing morphisms because the “side condition” ensures that $F \square G$ does not contain alternating cycles. However, when looking at more general models [21, 23], we need another way to keep the execution finite. It is then natural to consider restricting our attention to simple paths and cycles.

Definition 19. A path (resp. cycle) is *simple* if it does not visit any vertex more than once.

But restricting the execution (resp. orthogonality) to simple paths (resp. cycles) breaks the adjunction.

Example 20. Figure 3 shows how the adjunction fails when one considers only simple paths and cycles. Let us write $G_1 ::_s G_2$ for the *simple execution* between G_1 and G_2 – i.e. the execution restricted to the edges coming from simple paths – and $G_1 \perp_s G_2$ when there is no alternating simple cycle in $G_1 \square G_2$.

On the left, we have no alternating simple cycle in $F \square G \square H$, so $F \perp_s G \otimes H$. On the other hand, one can see on the right that there is a simple alternating cycle C between the simple execution $F ::_s G$ and H , so $F ::_s G \not\perp_s H$. In fact, this cycle lifts to a non-simple cycle in $F \square G \square H$: the edges deb and $ce^{-1}a$ correspond to paths in $F \square G$ which both visit the vertices 1 and 2.

This example therefore explains why the introduction of coherence is necessary to restrict to simple paths: coherence keeps track of possible dependencies between paths (i.e. two paths that used the same edge) that may be forgotten and lead to troubles in the model (i.e. break associativity, which is the model's counterpart of the Church-Rosser property).

Coherent graphs were therefore a way of making the adjunction work with simple paths.

Remark 21. To ensure the finiteness of execution, another natural choice would be to restrict to paths and cycles without repeating *edges*, instead of vertices. This is done in a similar way to what we are about to show here, but using a non-reflexive coherence relation [22, §5.2].

Definition 22. Let G be a graph. The *simple coherence* relation on $E(G)$ is defined as follows: two edges are coherent if and only if they do *not* have a common vertex.

Definition 23. A coherent graph G is *simple* if \circlearrowright_G is included in the simple coherence on $E(G)$.

For instance, any coherence on a matching is simple, and the *chordless coherence* defined in section 4 is included in the simple coherence.

Proposition 24. Let G and H be two simple coherent graphs with $V(G) = V(H)$. A coherent alternating path or cycle in $G \square H$ is simple. The converse holds if \circlearrowright_G and \circlearrowright_H are equal to the simple coherences.

Proof. We prove the proposition for paths; a similar argument works for cycles.

Let π be an alternating path in $G \square H$, and suppose it is not simple. Among all the vertices visited at least twice, let v be the one with the earliest first visit; this first visit involves an edge e with $t(e) = v$. Suppose w.l.o.g. $e \in E(G)$. The second visit to e involves an edge $e' \in E(G)$ (either incoming or outgoing) because of alternation; $e \neq e'$, or else $s(e)$ would contradict the minimality of v . And since $t(e) = v \in \{s(e'), t(e')\}$, $e \smile_G e'$ by definition of simplicity of \circlearrowright_G . Thus, π is incoherent.

Conversely, in an alternating simple path, the source and destination are incident to a single edge, while every intermediate vertex is incident to exactly one edge of G and one edge of H . So the path cannot contain two incident edges from the same graph. \square

Thus, as long as we execute two finite simple coherent graphs together, we always get finite graphs. It only remains to check that simplicity is preserved by execution to show that we have devised an universe closed under execution of generalized programs which are all finite graphs.

Proposition 25. *If G and H are simple, then $G :: H$ is simple.*

Proof. Let $e \neq e'$ have a common source vertex in $G :: H$; we wish to show $e \smile e'$. By definition, e and e' correspond to paths in $G \square H$, respectively π et π' . Let ρ be their longest common prefix: $\pi = \rho \sigma$ and $\pi' = \rho \sigma'$. Since $\pi \neq \pi'$, at least one of σ and σ' must be non-empty. Furthermore, if only one of them were empty, the destination of π would be a vertex in $V(G) \cap V(H)$ which is impossible.

Therefore, there exists $f \neq f'$ such that $\sigma = f\psi$, $\sigma' = f'\psi'$. Then $f \smile f'$ by simplicity of \circlearrowright_G si $f, f' \in E(G)$, or of \circlearrowright_H si $f, f' \in E(H)$ (necessarily, f and f' come from the same graph; this is where the common starting vertex assumption comes into play). Thus, π and π' are not mutually coherent.

If e and e' have the same target vertex, the proof is the same in the opposite graph; if $s(e) = t(e')$ or $t(e) = s(e')$, it suffices to look at the pair of edges incident to the common vertex. \square

3.4 A model made out of coherent graphs

Now, we carry out the orthogonality-based construction of a model of MLL, which was previously sketched, in the setting of coherent graphs.

Definition 26. For two coherent graphs G, H with $V(G) = V(H)$, we write $G \perp H$ when $G \square H$ contains no *coherent* alternating cycle.

As before, the tensor and the execution satisfy an adjunction with respect to \perp :

Proposition 27. *Let F, G and H be three coherent graphs. Suppose that $V(F) = V(G) \sqcup V(H)$. Then $F \perp G \otimes H$ if and only if $F \perp G \otimes \emptyset_{V(H)}$ and $F :: G \perp H$.*

Proof. A coherent alternating cycle between F and $G \otimes H$ either is between F and G or involves at least one edge of H . The former case corresponds to $F \perp G \otimes \emptyset_{V(H)}$. Cycles fitting in the latter case correspond exactly to alternating cycles between $F :: G$ and H . \square

The notion of conduct is the same as before, and actually the next few definitions and theorems are rather formal and are not affected by adding a coherence relation to graphs.

Definition 28. A *conduct* \mathbf{A} on a vertex set $V(\mathbf{A})$ is a set of coherent graphs with vertex set $V(\mathbf{A})$ such that $\mathbf{A} = \mathbf{A}^{\perp\perp}$.

Definition 29. Let \mathbf{A} and \mathbf{B} be two conducts such that $V(\mathbf{A}) \cap V(\mathbf{B}) = \emptyset$. We define $\mathbf{A} \otimes \mathbf{B} = \{G \otimes H \mid G \in \mathbf{A}, H \in \mathbf{B}\}^{\perp\perp}$.

It follows that $\mathbf{A} \bowtie \mathbf{B} = \{G \otimes H \mid G \in \mathbf{A}, H \in \mathbf{B}\}^\perp$, and $\mathbf{A} \multimap \mathbf{B} = \mathbf{A}^\perp \bowtie \mathbf{B}$. As in the previous section, one easily obtains the following characterisation of the linear maps.

Proposition 30. *Let \mathbf{A} and \mathbf{B} be two conducts such that $V(\mathbf{A}) \cap V(\mathbf{B}) = \emptyset$. Then*

$$\mathbf{A} \multimap \mathbf{B} = \{F \mid \forall G \in \mathbf{A}, F \perp G \otimes \emptyset_{V(\mathbf{B})} \text{ and } F :: G \in \mathbf{B}\}$$

From these, one can define a categorical model of MLL, establishing that the interpretation of MLL proofs by coherent graphs is sound. This can be further refined by showing that cut elimination is soundly represented by the execution of two graphs.

Details of the proofs of these results are ghastly, as one needs to consider *delocations*. Indeed, as each conduct has a location, i.e. a specified set of vertices, one needs to consider isomorphic copies of conducts when defining all the basic operations. Morally, the categorical model consists of the category whose objects are conducts and morphisms from \mathbf{A} to \mathbf{B} are coherent graphs in $\mathbf{A} \multimap \mathbf{B}$. In details, the conduct $\mathbf{A} \multimap \mathbf{B}$ may not be defined if by some bad luck we ended up with a situation where $V(\mathbf{A}) = V(\mathbf{B})$. This is circumvented by defining the set of morphisms from \mathbf{A} to \mathbf{B} are coherent graphs in $\phi(\mathbf{A}) \multimap \psi(\mathbf{B})$, where ϕ and ψ rename the sets of vertices to ensure disjointness. This leads to complications in the definition of composition. However, the details of the involved combinatorics is provided in full in the second author's first paper on interaction graphs [21] and the addition of a coherence relation on edges of graphs do not change the proof: understanding the word "graph" as meaning "coherent graph", the original proof can be used word to word to obtain the following result, as only the associativity and adjunction properties are used.

Similarly, as the tensor product is not a total operation, it needs to be dealt with through delocations. Again, the method is exactly that of the previous work mentioned above.

Theorem 31. *The following defines a *-autonomous category: conducts as objects, coherent graphs as morphisms – modulo delocations and composed by execution –, delocated tensor as monoidal product, and $(-)^{\perp}$ as dualization.*

Now, recall that we motivated the introduction of the coherence relation by mentioning a SUM rule, which is supposed to be interpreted by the incoherent sum \boxplus . But the category of coherent conducts and graphs does *not* interpret this rule: the commutations which distributes SUM over arbitrary contexts do not correspond to equalities of coherent graphs. However, one can get a model of MLL+SUM by identifying two coherent graphs when they have the same “deterministic sub-graphs”, that is, the same maximal cliques. Note this identifies coherent graphs which are indistinguishable w.r.t. orthogonality. I.e. this equivalence relation is finer than what Seiller calls *universal equivalence* [22] and which is defined as $G \sim H$ if for all graph F the set of alternating cycles in $G \square F$ is equal to the set of alternating cycles in $H \square F$. The universal equivalence is quite natural, and implies that the equivalent graphs F, G are observationally indistinguishable in *all* Interaction Graphs models definable. Thus, the quotient needed to represent faithfully the SUM rule is quite natural to consider and provides a very satisfactory model, as does the quotient w.r.t. universal equivalence.

Theorem 32. *Let \mathcal{R} be the following equivalence relation on graphs: $G \mathcal{R} H$ iff $V(G) = V(H)$ and there is a bijection between the maximal cliques (for inclusion) of $E(G)$ and those of $E(H)$, commuting with the source and target maps. (That is, G and H “have the same” maximal cliques.)*

\mathcal{R} is a congruence on the hom-sets of the category in the previous theorem, and the quotient is a model of MLL+SUM, interpreting SUM as \boxplus .

Of course, each equivalence class then has a canonical representative which is just the incoherent sum of the maximal cliques. However, the use of a coherence relation remains interesting since it potentially allows for much more concise representatives – recall that *sparsity* of non-deterministic programs was an initial motivation.

4 Principal conducts and cographic proof nets

4.1 Principal conducts

Our goal is now to study the interpretation of MLL types in the model defined by Theorem 31 – that is, without the quotient – and recover a notion of proof net together with an associated correctness criterion. To do so we will look at the generation of conducts by single “tests” in their orthogonal.

Definition 33. Let \mathbf{A} be a conduct. A *generator* of \mathbf{A} is a graph $G \in \mathbf{A}^\perp$ such that $\mathbf{A} = \{G\}^\perp$.

Let \mathbf{A} be generated by a set X , that is, $\mathbf{A} = X^\perp$; then it admits a generator, namely, the incoherent sum of all graphs in X (even if X is infinite, \boxplus makes sense as an infinitary operation). But this doesn't simplify much the description of \mathbf{A} . What we would like is to find generators with a reasonable size, e.g. with a number of edges polynomial in the number of vertices. Coherent interaction graphs allow us to build a model of MLL consisting entirely of conducts with such generators.

Definition 34. A conduct \mathbf{A} is *principal* if it admits a generator without parallel edges, i.e. multiple edges with both the same sources and the same targets. It is *bi-principal* if both \mathbf{A} and \mathbf{A}^\perp are principal.

Theorem 35. *Principal conducts are closed under \otimes and \wp , and bi-principal conducts constitute a non-trivial model of MLL.*

Thus, there is a model of MLL whose conducts all admit generators whose number of edges is at most quadratic in the number of vertices (with no parallel edges allowed, it is bounded by the number of ordered pairs of vertices).

Remark 36. Moreover, given a fixed assignment of atoms to bi-principal conducts, the number of vertices – and thus the number of edges – of a generator is polynomial in the size of the interpreted formula.

The non-triviality comes from the fact that the unique conduct on $\{*\}$ is bi-principal. This conduct does not contain any coherent graph, but this tells us that to get a truly non-empty bi-principal conduct, we can just take any provable formula and interpret every atom by $\{*\}$. The rest of the theorem is a consequence of the following explicit constructions on generators.

Proposition 37. *For all coherent graphs G and H with $V(G) \cap V(H) = \emptyset$, we have $\{G\}^\perp \wp \{H\}^\perp = \{G \otimes H\}^\perp$ and $\{G\}^\perp \otimes \{H\}^\perp = \{G \wp H\}^\perp$, where $V(G \wp H) = V(G) \sqcup V(H)$, and*

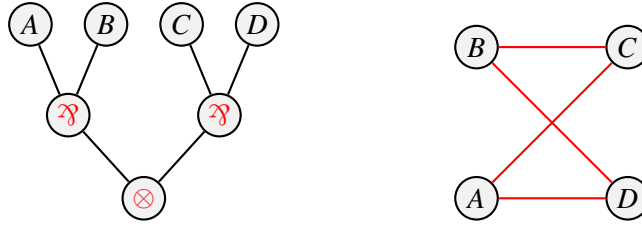
$$\text{Coh}(G \wp H) = \text{Coh}(G) \oplus \text{Coh}(H) \oplus (V(G) \times V(H) \sqcup V(G) \times V(H), \emptyset)$$

That is, for each $u \in V(G)$ and $v \in V(H)$, we add both (u, v) and (v, u) and make them incoherent with all other edges.

Proof. First, note that $\{G\}^\perp \wp \{H\}^\perp = \{X \otimes Y \mid X \in \{G\}^{\perp\perp}, Y \in \{H\}^{\perp\perp}\}^\perp$ by definition. By instantiating $X = G, Y = H$ and using the contravariance of $(-)^{\perp}$, we get $\{G\}^\perp \wp \{H\}^\perp \subseteq \{G \otimes H\}^\perp$.

Conversely, let $X \in \{G\}^{\perp\perp}, Y \in \{H\}^{\perp\perp}$ and $Z \perp G \otimes H$; our goal is to show $Z \perp X \otimes Y$. To do so, we use the adjunction: $Z \perp G \otimes H \implies Z :: G \perp H \implies Z :: G \in \{H\}^\perp \implies Z :: G \perp Y$ (because $Y \in \{H\}^{\perp\perp}$) $\implies Z \perp G \otimes Y$ etc. and in the end, $Z \perp X \otimes Y$.

As for $\{G \wp H\}^\perp$, the edges between $V(G)$ and $V(H)$ ensure that all graphs orthogonal to $G \wp H$ can be written as $X \otimes Y$ with $V(X) = V(G)$ and $V(Y) = V(H)$. $G \wp H \perp X \otimes Y$ entails $X \perp G$ and $Y \perp H$ (by looking at the coherent alternating cycles with vertices in $V(G)$, or in $V(H)$). Thus, we have $\{G \wp H\}^\perp \subseteq \{G\}^\perp \otimes \{H\}^\perp$. The reverse direction is proven by applying $(-)^{\perp\perp}$ to the inclusion $\{X \otimes Y \mid X \in \{G\}^\perp, Y \in \{H\}^\perp\} \subseteq \{G \wp H\}^\perp$ (note that $S^{\perp\perp\perp} = S^\perp$ for any set S). Note that this proof gives an “internal completeness” result as a bonus: every graph in this conduct decomposes as a tensor. \square

Figure 4: The cograph associated to the MLL formula $(A \wp B) \otimes (C \wp D)$

4.2 Cographic proof nets

Among bi-principal conducts, we have all those built from $\{*\}$ using the multiplicative connectives. Let us take a closer look at what those look like. The following shows that the coherence relation of the canonical generator of such a conduct is entirely determined by its vertices and edges.

Definition 38. Let G be a graph. The *chordless coherence* relation is defined as follows: $e, f \in E(G)$ are incoherent if and only if either e and f are incident, or some $g \in E(G)$ is incident to both e and f .

Proposition 39. Any generator obtained from the one-vertex graph by the constructions of the previous proposition is equipped with the chordless coherence relation.

The proof is a straightforward induction. Our choice of naming is justified by the following.

Proposition 40. Let G and H be two graphs equipped with the chordless coherence relation. An alternating cycle in $G \square H$ is coherent if and only if it is simple and there is no edge of $G \square H$ between two non-consecutive vertices of the cycle (such an edge is called a chord in graph theory). The same holds for alternating paths in $G \square H$ with endpoints in $G \triangle H$.

Proof. Let π be an alternating cycle in $G \square H$, with a chord $e \in \pi$, and let $u = s(e)$, $v = t(e)$. Suppose w.l.o.g. that $e \in E(G)$. Since π is alternating, there exist $e_u, e_v \in E(G) \cap \pi$ such that $u \in \{s(e_u), t(e_u)\}$ and similarly for v . Since u and v are non-consecutive, $e_u \neq e_v$. Then, because e is incident to both, $e_u \smile_G e_v$, making π incoherent. Conversely, if π is incoherent, by finding the cause of incoherence between two edges of π according to the definition, one can exhibit either a repeated vertex or a chord. \square

Thus, it remains only to understand the set of graphs (without coherence) generated by \otimes and \wp . First, one can remark that those canonical generators are all symmetric directed graphs; it will be more convenient to consider them as *undirected* graphs. In fact, it is common to associate an undirected graph to a classical propositional formula by interpreting \vee as a disjoint union and \wedge as the dual operation (see e.g. [1]). This is exactly what we do with MLL formulae; note that the \wp on conducts is interpreted as a \otimes on generators, and thus corresponds to the classical disjunction – which is indeed the classical counterpart of \wp . (See the example of Figure 4.) The graphs obtained this way are *cographs* [2], a well-known class of undirected graphs with many different characterizations.

In the study of Multiplicative Linear Logic, the cograph associated to a formula has previously been used in Retoré’s version of proof nets, which he called “R&B-cographs” [20]:

Definition 41. A *cographic proof* is a vertex set V , together with a pair of graphs (G, M) such that $V(G) = V(M) = V$, G is a cograph and M is a matching; additionally, V is labeled by MLL atoms in such a way that if two vertices are adjacent in M , then their labels are dual.

This is indeed a faithful representation of proofs:

- from a cograph, one can recover the original formula up to associativity/commutativity of \wp/\otimes ,

- we saw earlier that the matching associated to a proof captures its computational content.

But all cographic proofs do not come from proofs – actually, they correspond to potentially incorrect *proof structures*, so this is really just another way to describe proof nets (but one that arises naturally from our GoI construction!). Retoré also proposed a correctness criterion for cographic proofs; it was later rediscovered by Ehrhard [3] and also underlies Hughes’s “combinatorial proofs” for classical logic [13]:

Theorem 42 ([20, 3]). *Let (G, M) be a cographic proof and Φ the MLL formula whose cograph is G . Then M corresponds to a proof of Φ in MLL with the Mix rule if and only if there is no chordless alternating cycle between G and M .*

This “no chordless alternating cycle” condition is just orthogonality between coherent interaction graphs: $G \perp M$ where G is equipped with the chordless coherence relation! The “only if” condition can be recovered immediately by interpreting the corresponding proof in our model, sending the atoms of Φ to $\{*\}$ so that G is exactly the canonical generator of the bi-principal conduct corresponding to Φ , and M is the element of this conduct corresponding to the proof. (For this, our model needs to interpret the Mix rule; this is the case since $\mathbf{A} \otimes \mathbf{B} \subseteq \mathbf{A} \wp \mathbf{B}$ holds for any two conducts \mathbf{A}, \mathbf{B} .)

As for the “if” direction, it amounts to a *sequentialization theorem* on proof nets, and thus it can’t simply fall out of the above work on coherent graphs: a proof of sequentialization must necessarily involve some hard combinatorics. But translated into our GoI model, it almost gives a *full completeness* theorem. Any M which is orthogonal to a cograph G is a proof of the corresponding formula, up to two morphological conditions: M must be a matching, and its edges must have dual atoms as endpoints.

5 Perspectives

The obvious direction for future work is to extend the model to larger fragments of linear logic. One could extend our model of MELL to handle exponentials, for instance using *graphings* such as in previous work [24]. But to keep finiteness and sparsity properties, it might be preferable to find a way to use finite objects instead of measure spaces of infinite cardinality. The ability to take non-deterministic sums in our model also suggests an extension to differential linear logic [5, 4]. It would be interesting to understand the “correctness criterion” for differential nets with promotion boxes arising from GoI orthogonality; we might recover Pagani’s visible acyclicity [19] which is already characterised interactively.

The cographic representation of formulae and proofs could also lead to the discovery of new operations in a multiplicative setting. For instance, *modular decomposition* of graphs (see e.g. [10]) extends to arbitrary graphs the correspondence between cographs and formulae. This may be used to define generalized multiplicative connectives.

References

- [1] Seth Chaiken, Neil V. Murray & Erik Rosenthal (1989): *An application of P4-free graphs in theorem-proving*. *Annals of the New York Academy of Sciences* 555(1), pp. 106–121, doi:10.1111/j.1749-6632.1989.tb22442.x.
- [2] D. G. Corneil, H. Lerchs & L. Stewart Burlingham (1981): *Complement reducible graphs*. *Discrete Applied Mathematics* 3(3), pp. 163–174, doi:10.1016/0166-218X(81)90013-5.
- [3] Thomas Ehrhard (2014): *A new correctness criterion for MLL proof nets*. In: *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS ’14, Vienna, Austria, July 14 - 18, 2014*, pp. 38:1–38:10, doi:10.1145/2603088.2603125.

- [4] Thomas Ehrhard (2017): *An introduction to differential linear logic: proof-nets, models and antiderivatives*. *Mathematical Structures in Computer Science*, pp. 1–66, doi:10.1017/S0960129516000372.
- [5] Thomas Ehrhard & Laurent Regnier (2006): *Differential interaction nets*. *Theoretical Computer Science* 364(2), pp. 166–195, doi:10.1016/j.tcs.2006.08.003.
- [6] Jean-Yves Girard (1987): *Multiplicatives*. *Rendiconti del Seminario Matematico. Università e Politecnico di Torino* 45(Special Issue).
- [7] Jean-Yves Girard (1989): *Towards a Geometry of Interaction*. In J. W. Gray & A. Scedrov, editors: *Categories in Computer Science and Logic: Proc. of the Joint Summer Research Conference*, American Mathematical Society, Providence, RI, pp. 69–108, doi:10.1090/conm/092/1003197.
- [8] Jean-Yves Girard (1996): *Proof-nets: The parallel syntax for proof-theory*. In: *Logic and Algebra*, Marcel Dekker, pp. 97–124.
- [9] Jean-Yves Girard (2011): *Geometry of Interaction V: Logic in the hyperfinite factor*. *Theoretical Computer Science* 412(20), pp. 1860–1883, doi:10.1016/j.tcs.2010.12.016.
- [10] Michel Habib & Christophe Paul (2010): *A survey of the algorithmic aspects of modular decomposition*. *Computer Science Review* 4(1), pp. 41–59, doi:10.1016/j.cosrev.2010.01.001.
- [11] Dominic Hughes & Willem Heijltjes (2016): *Conflict nets: efficient locally canonical MALL proof nets*. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2016*, ACM, New York, U. S. A., pp. 437–446, doi:10.1145/2933575.2934559.
- [12] Dominic J. D. Hughes & Rob J. Van Glabbeek (2005): *Proof Nets for Unit-free Multiplicative-additive Linear Logic*. *ACM Transactions on Computational Logic* 6(4), pp. 784–842, doi:10.1145/1094622.1094629.
- [13] Dominic J.D. Hughes (2006): *Proofs Without Syntax*. *Annals of Mathematics* 143(3), pp. 1065–1076, doi:10.4007/annals.2006.164.1065.
- [14] Martin Hyland & Andrea Schalk (2003): *Glueing and orthogonality for models of linear logic*. *Theoretical Computer Science* 294(1), pp. 183–231, doi:10.1016/S0304-3975(01)00241-9.
- [15] G. M. Kelly & M. L. Laplaza (1980): *Coherence for compact closed categories*. *Journal of Pure and Applied Algebra* 19, pp. 193–213, doi:10.1016/0022-4049(80)90101-2.
- [16] Ganna Kudryavtseva & Volodymyr Mazorchuk (2009): *On three approaches to conjugacy in semigroups*. *Semigroup Forum* 78(1), pp. 14–20, doi:10.1007/s00233-008-9047-7.
- [17] François Maurel (2003): *Nondeterministic Light Logics and NP-Time*. In: *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 241–255, doi:10.1007/3-540-44904-3_17.
- [18] Alberto Naibo, Mattia Petrolo & Thomas Seiller (2016): *On the Computational Meaning of Axioms*, pp. 141–184. Springer International Publishing, doi:10.1007/978-3-319-26506-3_5.
- [19] Michele Pagani (2012): *Visible acyclic differential nets, Part I: Semantics*. *Annals of Pure and Applied Logic* 163(3), pp. 238–265, doi:10.1016/j.apal.2011.09.001.
- [20] Christian Retoré (2003): *Handsome proof-nets: perfect matchings and cographs*. *Theoretical Computer Science* 294(3), pp. 473–488, doi:10.1016/S0304-3975(01)00175-X.
- [21] Thomas Seiller (2012): *Interaction graphs: Multiplicatives*. *Annals of Pure and Applied Logic* 163(12), pp. 1808–1837, doi:10.1016/j.apal.2012.04.005.
- [22] Thomas Seiller (2012): *Logique dans le Facteur Hyperfini: Géométrie de l'Interaction et Complexité*. Thèse de doctorat, Aix-Marseille Université. Available at <https://tel.archives-ouvertes.fr/tel-00768403/document>.
- [23] Thomas Seiller (2016): *Interaction graphs: Additives*. *Annals of Pure and Applied Logic* 167(2), pp. 95–154, doi:10.1016/j.apal.2015.10.001.
- [24] Thomas Seiller (2017): *Interaction graphs: Graphings*. *Annals of Pure and Applied Logic* 168(2), pp. 278–320, doi:10.1016/j.apal.2016.10.007.