

MELL proof-nets in the category of graphs

Giulio Guerrieri, Giulia Manara, Luc Pellissier, Lorenzo Tortora de Falco,
Lionel Vaux Auclair

► **To cite this version:**

Giulio Guerrieri, Giulia Manara, Luc Pellissier, Lorenzo Tortora de Falco, Lionel Vaux Auclair. MELL proof-nets in the category of graphs. 5th International Workshop on Trends in Linear Logic and Applications (TLLA 2021), Jun 2021, Rome (virtual), Italy. lirmm-03271478

HAL Id: lirmm-03271478

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-03271478>

Submitted on 25 Jun 2021

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

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



MELL proof-nets in the category of graphs

Giulio Guerrieri* Giulia Manara† Luc Pellissier‡
Lorenzo Tortora de Falco§ Lionel Vaux Auclair†

Abstract

We present a formalization of proof-nets (and more generally, proof-structures) for the multiplicative-exponential fragment of linear logic, with a novel treatment of boxes: instead of integrating boxes into to the graphical structure (*e.g.*, by adding explicit auxiliary doors, plus a mapping from boxed nodes to the main door, and *ad hoc* conditions on the nesting of boxes), we fix a graph morphism from the underlying graph of the proof-structure to the tree of boxes given by the nesting order. This approach allows to apply tools and notions from the theory of algebraic graph transformations, and obtain very synthetic presentations of sophisticated operations: for instance, each element of the Taylor expansion of a proof-structure is obtained by a pull-back along a morphism representing a thick subtree of the tree of boxes. A treatment of cut elimination in this framework currently under development.

Linear Logic (LL) [13] has been introduced by Girard as a refinement of intuitionistic and classical logic that isolates the infinitary parts of reasoning under two modalities: the *exponentials* ! and ?. These modalities give a logical status to operations of resource/hypothesis management such as *copying/contraction* or *erasing/weakening*: a proof without exponentials corresponds to a program/proof that uses its arguments/hypotheses *linearly*, i.e. exactly once, while an exponential proof corresponds to a program/proof that can use its arguments/hypotheses at will.

Among other contributions, this refinement provided by LL brought about a reflection on the following central questions in proof-theory: What is a proof? How can we represent a proof? Here we aim to contribute to the way proofs can be represented in LL, pushing forward with Girard's graphical spirit [13].

Proof-nets and proof-structures. One of the features of LL is that it allows us to represent its proofs as *proof-nets*, a graphical syntax alternative to sequent calculus. Sequent calculus is a standard formalism for several logical systems. However, sequent calculus forces an order among inference rules even when they are evidently

*University of Bath, Department of Computer Science, Bath, UK

†Aix-Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France

‡Université Paris Est Créteil, LACL, F-94010 Créteil, France

§Università Roma Tre, Dipartimento di Matematica e Fisica, Rome, Italy

independent, a drawback called *bureaucracy*. Proof-nets, instead, are a geometrical, parallel and bureaucracy-free representation of proofs as labeled directed *graphs*. In proof-nets deductive rules are disposed on the plane, in parallel, and connected only by their causal relation. Clearly, not all graphs that can be written in the language of LL are proof-nets, i.e. represent a proof in LL sequent calculus. Proof-nets are special inhabitants of the wider land of *proof-structures*: they can be characterized, among proof-structures, by abstract (geometric) conditions called correctness criteria [13]. The procedure of cut-elimination can be applied directly to proof-structures, and proof-nets can also be seen as the proof-structures with a good behavior with respect to cut-elimination [2]. Cut-elimination defined on proof-structures is more elegant than in sequent calculus because it drastically reduces the need for commutative steps, the non-interesting and bureaucratic burden in a cut-elimination procedure for sequent calculus. Indeed, in proof-structures there is no last rule, and so most commutative cut-elimination cases just disappear.

Boxes. Regrettably, the one above is a faithful picture of the advantages of proof-structures only in the multiplicative fragment of LL (MLL) [5], which does not have the exponentials and so it is not sufficiently expressive to encode classical or intuitionistic logic (or the λ -calculus). To handle the exponentials, Girard was forced to introduce *boxes*. They come with the *black-box principle*: “boxes are treated in a perfectly modular way: we can use the box B without knowing its content, i.e., another box B' with exactly the same doors would do as well” [13].

According to this principle, boxes forbid interaction between their content and their outer environment. This is evident in the definition of correctness criteria for MELL (the multiplicative-exponential fragment of LL) and in the definition of cut-elimination steps for MELL. Let us consider cut-elimination. Cut-elimination steps for the $!$ -modality in MELL require us to duplicate or erase whole sub-proofs. Proofs in sequent calculus are tree-shaped and bear a clear notion of last rule, the root of the tree. As a consequence, given a $!$ -rule r in a sequent calculus proof, there is an evident sub-proof ending with r , the sub-tree rooted in r , which is the content of the box associated with r . So, non-linear cut-elimination steps can easily be defined by duplicating or erasing such sub-trees. In *proof-structures*, instead, the situation radically changes, because a proof-structure in general has many last rules, one for each formula in the conclusions. Given a rule r it is not clear how to find a sub-proof-structure ending with r . Thus, in order to define cut-elimination steps for the $!$ -modality in MELL proof-structures—which requires to identify some sub-proof-structure—some information has to be added to graphs.

Bureaucracy comes back! The typical solution betrays Girard’s spirit and re-introduces part of the bureaucracy in MELL proof-structures, pairing each $!$ -rule with an *explicit box* containing the sub-proof that can be duplicated or erased during cut-elimination. In some fragments of MELL (for instance the intuitionistic one corresponding to the λ -calculus [22] or more generally the polarized one [17, 1])

where proof-structures still have an implicit tree-like structure (since among the conclusions there is always exactly one distinct output, the analogue of sequent calculus last rule), an explicit box is actually not needed, and boxes can be recovered by a correctness criterion. But here we are interested in the full (classical) MELL fragment, where linear negation is involutive and classical duality can be interpreted as the possibility of juggling between different conclusions. And we want to disentangle the identification of boxes from correctness. Concretely, in the literature mainly two kinds of solution that make use of explicit boxes can be found:

1. A MELL proof-structure is a directed graph (defined *non-inductively*) together with some additional information to identify the content and the border of each box. This additional information can be provided either informally, just drawing the border of each box in the graph [13, 6, 19], but then the definition of MELL proof-structure is not rigorous; or in a more formal way [9, 16, 7], but then the definition is highly technical and *ad hoc*;
2. A MELL proof-structure is an *inductive* directed graph [18, 20, 23, 8], where with any vertex v of type $!$ is associated another directed graph representing the content of the box of v . This inductive solution can be taken to extremes by representing proof-structures with term-like syntax [10, 4].

The drawback of Item 1 is that the definition of MELL proof-structure is not easily manageable because either it is not precise or it is too tricky. Item 2, instead, provides more manageable definitions of MELL proof-structures, but another drawback arises: Girard's idea of proofs as graphs in watered-down, and it makes way for more *ad hoc* means to represent proofs.

Our contribution. Following [14, 15], we present here a purely graphical definition of MELL proof-structures, so as to keep Girard's original intuition of a proof-structure as a graph even in MELL. Our definition follows the non-inductive approach seen in Item 1, and improves its state of the art. It is completely based on standard notions coming from *graph theory*, being *formal* (with an eye towards complete computer formalization) but still *easy to handle*, and it avoids *ad hoc* technicalities to identify the border and the content of a box. We use n -ary vertices of type $?$ collapsing weakening, dereliction and contraction (like in [6]): in this way, we get a *canonical* representation of MELL proof-structures.

Roughly, we define a MELL proof-structure R as made of three components:

- a directed labeled *graph* $|R|$ representing R without boxes,
- a *tree* \mathcal{A}_R representing the nesting order of the boxes of R ,
- a *graph morphism* box_R from $|R|$ to the reflexive-transitive closure \mathcal{A}_R° of \mathcal{A}_R , which allows us to recognize the content and border of all boxes in R .

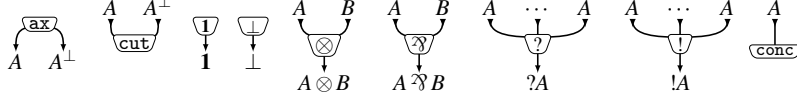


Figure 1: Vertices of a DiLL graph, with their labels and their typed inputs and outputs (ordered left-to-right). Vertices of type `concl` are often omitted.

We now give a precise definition of a MELL proof-structure, and more generally of a DiLL proof-structure [11, 20, 23], which extends MELL by allowing for ! the duals of dereliction, contraction and weakening for ?. We suppose known the definitions of directed graph, rooted tree, and morphism of these structures. An *input* (resp. *output*) of a vertex v is an edge incoming in (resp. outgoing from) v .

Definition 1 (Proof-structure). A DiLL graph G is a (finite) directed graph with:

- edges e labeled by a MELL formula $c(e)$, the type of e , where MELL formulas are defined by the grammar $A, B ::= X \mid X^\perp \mid \mathbf{1} \mid \perp \mid A \otimes B \mid A \wp B \mid !A \mid ?A$, and the linear negation $(\cdot)^\perp$ is defined via De Morgan laws $\mathbf{1}^\perp = \perp$, $(A \otimes B)^\perp = A^\perp \wp B^\perp$ and $(!A)^\perp = ?A$, so as to be involutive, i.e. $A^{\perp\perp} = A$;
- vertices v labeled by $\ell(v) \in \{\text{ax}, \text{cut}, \mathbf{1}, \perp, \otimes, \wp, ?, !, \text{concl}\}$, the type of v ; all the vertices verify the conditions of Figure 1; the input of any vertex of type `concl` is a conclusion of G ; an input of a vertex of type `!` is a `!`-input of G ;
- an order $<_G$ that is total on the conclusions of G and on the inputs of each vertex of type `?`, `!`.

A (DiLL) proof-structure is a triple $R = (|R|, \mathcal{A}, \text{box})$ where:

- $|R|$ is a DiLL graph, called the underlying graph of R ;
- \mathcal{A} is a rooted tree, called the box-tree of R ;
- $\text{box}: |R| \rightarrow \mathcal{A}^\circ$ is a morphism of directed graphs, called the box-function of R (\mathcal{A}° is the reflexive-transitive closure of \mathcal{A}), such that:
 - it induces a partial bijection from the `!`-inputs of $|R|$ and the edges in \mathcal{A} ,¹
 - for any edge from v' to v , if $\text{box}(v) \neq \text{box}(v')$ then $\ell(v) \in \{!, ?\}$.²

A proof-structure $R = (|R|, \mathcal{A}, \text{box})$ is:

1. MELL if all vertices in $|R|$ of type `!` have exactly one input, and the partial bijection induced by box from the `!`-inputs of $|R|$ to the edges in \mathcal{A} is total;
2. DiLL₀ (or resource or box-free) if the box-tree \mathcal{A} of R is just one vertex.

An example of a MELL proof-structure is given in Figure 2.

¹The `!`-inputs that box maps to the edges of \mathcal{A} are the ones associated with a box. In DiLL, a vertex of type `!` may have several inputs, and not all of them are necessarily associated with a box.

²Roughly, it says that the border of a box is made of (inputs of) vertices of type `!` or `?`.

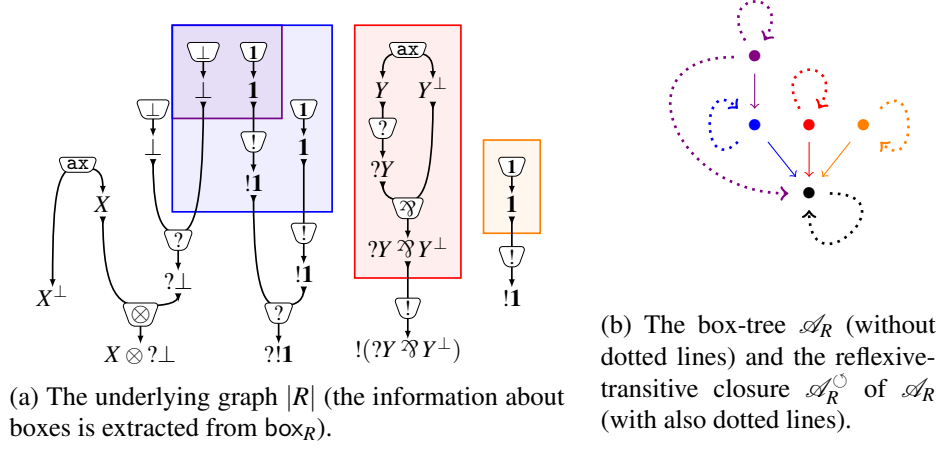


Figure 2: An MELL proof-structure $R = (|R|, \mathcal{A}_R, \text{box}_R)$: box_R is given by colors.

Our proof-structures are still manageable: in this framework, sophisticated operations on them such as *cut-elimination* or the *Taylor expansion* [12], as well as *correctness graphs* to characterize the proof-structures that are proof-nets (i.e. that correspond to proofs in the LL sequent calculus), are easy to define formally. Also, *identity* of proof-structures is defined naturally as a graph isomorphism.

Taylor expansion via pullbacks. As a test of the usability of our formalism for MELL proof-structures, we give an elegant definition of their *Taylor expansion*, by means of *pullbacks*. The Taylor expansion $\mathcal{T}(R)$ [12] of a MELL proof-structure R is a possibly infinite set of *box-free* proof-structures: roughly, each element of $\mathcal{T}(R)$ is obtained from R by replacing each box B in R with n_B copies of its content (for some $n_B \in \mathbb{N}$), recursively on the depth of R . Note that n_B depends not only on B but also on which “copy” of all boxes containing B we are considering.

Usually, the Taylor expansion of MELL proof-structure is defined globally and inductively [19, 21]: with a MELL proof-structure R is directly associated its Taylor expansion (the whole set!) by induction on the depth of R . A drawback of this approach is that, for each element of $\mathcal{T}(R)$, the way the different copies of the content of a box are merged has to be defined “by hand”, which is syntactically heavy.

Following [16, 14, 15], we adopt an alternative non-inductive approach: the Taylor expansion is defined pointwise. Indeed, a MELL proof-structure R has a tree structure \mathcal{A}_R made explicit by its graph morphism box_R . The definition of the Taylor expansion uses this tree structure: first, we define how to “expand” a tree via the notion of *thick subtree* [3] (roughly, it states the number of copies of each box to be taken, recursively): see Figure 3 for an example. We then take all these expansions of the box-tree \mathcal{A}_R of a proof-structure R and we *pull them back* to the underlying graph, finally we forget the tree structures associated with them. Figure 4 shows the element of the Taylor expansion of R in Figure 2 obtained from the thick subtree in Figure 3. Thus, pullbacks gives us an abstract and elegant

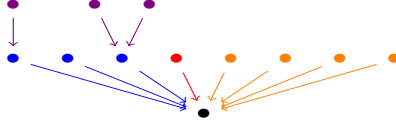


Figure 3: Example of a thick subtree of the tree \mathcal{A}_R in Figure 2 (the mapping to edges and vertices of \mathcal{A}_R is given by colors).

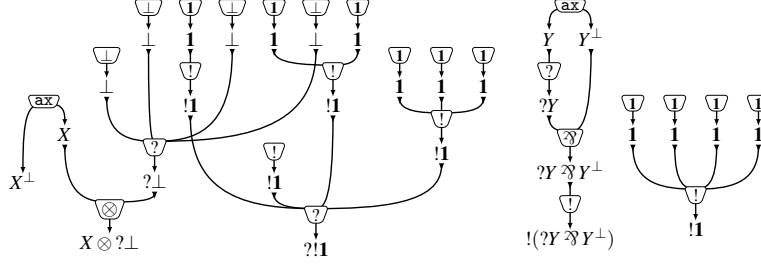


Figure 4: The element of the Taylor expansion of the MELL proof-structure R in Figure 2 obtained from the thick subtree of \mathcal{A}_R in Figure 3.

way to define the merging of the different copies of the content of a box in an element of $\mathcal{T}(R)$. The use of pullbacks is possible because all the components of our definition of a MELL proof-structure live in the category of directed graphs.

Cut-elimination via pullbacks. In a MELL proof-structure R , cut-elimination steps not involving the exponentials $!$ and $?$ are easy to define: they only entail *local* modifications on the underlying graph $|R|$ of R . An *exponential* cut-elimination step, instead, is complex because it may erase or duplicate the content of a box, or nest the content of a box inside another box: they involve *global* modifications not only in the underlying graph $|R|$ of R , but also in its box-tree \mathcal{A}_R and in its box-function box_R . Defining formally such operations “by hand” is syntactically heavy. In our formalism, exponential cut-elimination steps can be defined in two stages, where the heavy part is still managed in a concise way by the notion of pullback.

1. The cut between a $!$ -node (the main door of a box B) and a $?$ -node with n premises, induces a specific thick subtree τ of \mathcal{A}_R , the one that takes n copies of B . The *pullback* from $|R|$ and τ defines a new proof-structure R' (with tree structure τ), where the box B has been duplicated n times (erased if $n = 0$), but the cut has not been eliminated yet.
2. The result of the cut-elimination step is then obtained from R' by *local* changes in its underlying graph (by eliminating the cut-node similarly to multiplicative steps) and in the tree τ (by equalizing some nodes, to take into account the new nesting order of boxes).

Conclusions. In our framework, operations or relations on MELL proof-structures such as cut-elimination, or the Taylor expansion, or identity, which are intuitively

clear but tricky to define in a rigorous way, find an elegant, formal and not *ad hoc* definition. Our way to represent MELL proof-structures, and to define the Taylor expansion and the identity of MELL proof-structures has been introduced in [14] and used in [15]. The definition of cut-elimination is part of an ongoing work that we hope can be easily generalized to *parallel* cut-elimination steps. The goal is to ground the syntax of MELL proof-structures and the study of their operational properties on rigorous bases, by means of handy and elegant tools, pushing forward with Girard's original spirit of representing proofs as graphs.

References

- [1] B. Accattoli. Compressing polarized boxes. In *28th Annual Symposium on Logic in Computer Science (LICS 2013)*, pages 428–437. IEEE, 2013.
- [2] D. Béchet. Minimality of the correctness criterion for multiplicative proof nets. *Mathematical Structures in Computer Science*, 8(6):543–558, 1998.
- [3] P. Boudes. Thick subtrees, games and experiments. In *Typed Lambda Calculi and Applications, 9th International Conference (TLCA 2009)*, volume 5608 of *Lecture Notes in Computer Science*, pages 65–79, 2009.
- [4] J. Chouquet and L. Vaux Auclair. An application of parallel cut elimination in unit-free multiplicative linear logic to the taylor expansion of proof nets. In *27th EACSL Annual Conference on Computer Science Logic, CSL 2018*, volume 119 of *LIPICs*, pages 15:1–15:17. Schloss Dagstuhl, 2018.
- [5] V. Danos and L. Regnier. The structure of multiplicatives. *Arch. Math. Log.*, 28(3):181–203, 1989.
- [6] V. Danos and L. Regnier. Proof-nets and the Hilbert Space. In *Proceedings of the Workshop on Advances in Linear Logic*, pages 307–328. Cambridge University Press, 1995.
- [7] D. de Carvalho. Taylor expansion in linear logic is invertible. *Logical Methods in Computer Science*, 14(4):1–73, 2018.
- [8] D. de Carvalho, M. Pagani, and L. Tortora de Falco. A semantic measure of the execution time in linear logic. *Theor. Comput. Sci.*, 412(20):1884–1902, 2011.
- [9] D. de Carvalho and L. Tortora de Falco. The relational model is injective for multiplicative exponential linear logic (without weakenings). *Ann. Pure Appl. Logic*, 163(9):1210–1236, 2012.
- [10] T. Ehrhard. A new correctness criterion for MLL proof nets. In *23rd Conference on Computer Science Logic and 29th Symposium on Logic in Computer Science (CSL-LICS '14)*, pages 38:1–38:10. ACM, 2014.

- [11] T. Ehrhard and L. Regnier. Differential interaction nets. *Theor. Comput. Sci.*, 364(2):166–195, 2006.
- [12] T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 403(2-3):347–372, 2008.
- [13] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.
- [14] G. Guerrieri, L. Pellissier, and L. Tortora de Falco. Proof-net as graph, Taylor expansion as pullback. In *Logic, Language, Information, and Computation - 26th International Workshop, WoLLIC 2019*, volume 11541 of *Lecture Notes in Computer Science*, pages 282–300. Springer, 2019.
- [15] G. Guerrieri, L. Pellissier, and L. Tortora de Falco. Glueability of resource proof-structures: Inverting the Taylor expansion. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, volume 152 of *LIPICs*, pages 24:1–24:18. Schloss Dagstuhl, 2020.
- [16] G. Guerrieri, L. Pellissier, and L. Tortora de Falco. Computing connected proof(-structure)s from their Taylor expansion. In *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*, volume 52 of *LIPICs*, pages 20:1–20:18. Schloss Dagstuhl, 2016.
- [17] F. Lamarche. Proof nets for intuitionistic linear logic: Essential nets. Technical report, inria-00347336, 2008.
- [18] O. Laurent. Polarized proof-nets and lambda- μ -calculus. *Theor. Comput. Sci.*, 290(1):161–188, 2003.
- [19] D. Mazza and M. Pagani. The separation theorem for differential interaction nets. In *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference (LPAR 2007)*, volume 4790 of *Lecture Notes in Computer Science*, pages 393–407. Springer, 2007.
- [20] M. Pagani. The cut-elimination theorem for differential nets with promotion. In *Typed Lambda Calculi and Applications, 9th International Conference, (TLCA 2009)*, volume 5608 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2009.
- [21] M. Pagani and C. Tasson. The inverse Taylor expansion problem in linear logic. In *Proceedings of the 24th Annual Symposium on Logic in Computer Science (LICS 2009)*, pages 222–231. IEEE Computer Society, 2009.
- [22] M. Solieri. Geometry of resource interaction and Taylor-Ehrhard-Regnier expansion: a minimalist approach. *Mathematical Structures in Computer Science*, 28(5):667–709, 2018.
- [23] P. Tranquilli. Intuitionistic differential nets and lambda-calculus. *Theor. Comput. Sci.*, 412(20):1979–1997, 2011.