

A correctness criterion free from switchings

Marc Bagnol¹, Amina Doumane², and Alexis Saurin³ *

¹ IML, Université d'Aix-Marseille, bagnol@iml.univ-mrs.fr

² PPS, Université Paris Diderot & École centrale Paris, amina.doumane@student.ecp.fr

³ PPS, CNRS, Université Paris Diderot & INRIA, alexis.saurin@pps.univ-paris-diderot.fr

Abstract

We present a new correctness criterion for Multiplicative Linear Logic (MLL) proof nets. Our criterion generalizes a criterion recently introduced by Mogbil and Jacobé de Naurois in order to characterize the space complexity of the correctness problem for M(E)LL: while the Mogbil-Naurois criterion relies on a single switching, our criterion completely abstracts from the notion of switchings.

This is done in two steps. First we define dependency graphs on proof nets rather than correction graphs, as in Mogbil-Naurois, making them switching-independent. Then, we partition our criterion in two parts: one part deals exclusively with tensor, axiom and cut inferences while the other one deals with the structure of par inferences by requiring dependency graphs to be acyclic flowgraphs (or SDAG).

We finally compare both notions of dependency graphs and show that our criterion lifts naturally to the exponential case.

Linear logic, MLL, Proof nets, Correctness criterion, Danos-Regnier criterion, Dependency graphs, Switchings

1 Introduction

The benefits of Curry-Howard. Since the discovery of Curry-Howard correspondence [10], programming language theory and proof theory have been tightly intertwined. Among the numerous and fruitful back-and-forths between proofs and programs, linear logic [6] certainly stands as exemplary.

With linear logic came a very canonical representation of proofs (for fragments) of linear logic: proof nets are a graphical notation for proofs, resulting in very canonical proof objects (contrarily to sequent proofs) in which cut-elimination is very elegant, and simple. As such, they are certainly one of the most original innovations of linear logic.

Proof-nets and logical correctness. By representing proofs as graphs instead of the more usual sequential notations, we step back from the intuition of what a proof is [1]. As a direct consequence of moving from inductive objects (*e.g.* sequent calculus proof trees) to more geometrical objects (proof structures), it may not be easy to detect a logical incorrectness. Indeed, contrarily to most deductive systems [2] logical correctness is not local anymore. To give a status to those (possibly incorrect) objects, one speaks of *proof structures*, reserving the term *proof nets* to those objects which actually come from a sequent calculus proof.

One is thus willing to have conditions of a graphical and geometric nature ensuring the correctness of proof structures. Several correctness criteria have been introduced in the

*An extended version with supporting proofs can be found in [1] at <http://www.pps.univ-paris-diderot.fr/~saurin/Publi/DepGraphLong.pdf>

¹Typically a reasoning which is conducted progressively, in an ordered way, from its starting point to its conclusion, structured with intermediate lemmas and where each step shall be logically valid.

²In sequent calculus, correctness results from each inference rule being locally correct.

literature. Among the best-known criteria, one can refer to the original long-trip criterion (LT) [6], Danos-Regnier criterion (DR) [4], counter-proofs criterion (CP) [2, 7], contractibility (C) [3], graph-parsing criterion (GP) [8, 9], Dominator Tree (DT) [11] and more recently Mogbil-Nauois criterion (MN) [5].

On the structure and efficiency of correctness criteria. Danos-Regnier criterion is certainly the most standard correctness criterion for its beauty and because it is convenient to reason about. However, it is not efficient in order to effectively test the correctness of a proof structure since one shall check the connectedness and acyclicity of 2^n correction graphs, where n is the number of $\wp$ nodes in the structure. One can hardly hope to improve this bound by restricting the criterion to a well-chosen subset of switchings: one can indeed show that there exist proof structures having an arbitrary large number of $\wp$ of which all correction graphs but one are connected and acyclic (see [1]), unless one can use an additional information on the topology of the net. (For instance, under planarity assumptions one can restrict to considering only two switchings for deciding the correctness of a proof structure [12].). From this point of view, MN-criterion is slightly odd since it uses an arbitrarily picked switching in order to build the *dependency graph*.

Considering the above mentioned remark on the fact that incorrect proof structure may have an arbitrarily large proportion of connected and acyclic correction graphs, it is surprising that a single switching can ensure correctness. This is the starting point of our work: we wanted to understand what was the true use of picking a switching in MN-criterion for deciding correctness. In the rest of the paper, we will show that the switching can be completely forgotten and that one can reason directly on the proof structure.

Organization of the paper. This paper is a contribution to the problem of correctness of proof nets. We propose a new correctness criterion for M(E)LL.

After recalling the basics of linear logic and proof nets in Section 2 and correctness criteria in Section 3, we define in Section 4 a new correctness criterion which abstracts from switchings. This is done in two steps. First we define dependency graphs on proof nets rather than correction graphs in 4.2, as in Mogbil-Nauois, making them switching-independent. Then, we partition our criterion in two parts: one part deals exclusively with tensor, axiom and cut inferences (in 4.3) while the other one deals with the structure of par inferences by requiring dependency graphs to be acyclic flowgraphs. Finally we compare the two variants of dependency graphs in 4.5. A long version [1] supports the present paper with proofs and extension to MELLL.

2 Linear logic and proof nets

MLL. MLL formulas are built from the following grammar:

$$A, B := X \mid X^\perp \mid A \otimes B \mid A \wp B \quad (X \in \mathcal{V})$$

MLL is usually presented via a sequent calculus: an MLL sequent is a finite unordered list of MLL formulas, written $\vdash \Gamma$ and a proof is a tree with nodes labelled by (ax) , (cut) , $(\otimes)$, $(\wp)$ and edges are labelled by sequents as follows:

Identity Group:	$\frac{}{\vdash A, A^\perp} (ax) \quad \frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta} (cut)$
Multiplicative Group:	$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} (\otimes) \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} (\wp)$

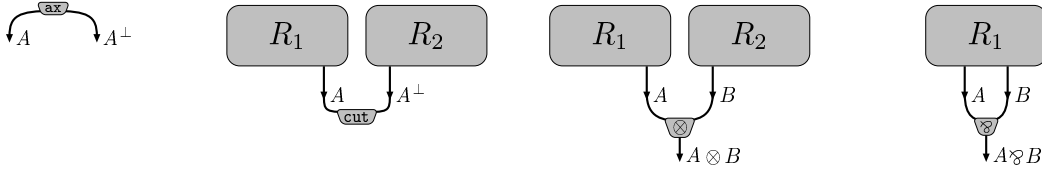
Proof structures and proof nets. Proof structures allow to present MLL proofs in a non-sequential way.

Definition 1 (Proof structure). *A proof structure is a finite undirected graph where vertices are labelled by the names of MLL inference rules or the special label $\mathbf{c}$ (for the conclusions of the proof) and edges are labelled with formulas of MLL. Moreover, edges which are adjacent to a vertex are partitioned into premises and conclusions according to the following rules:*

- Nodes of label $\otimes$ (resp. $\wp$) have two premises and one conclusion. If A is the label of the first premise and B the label of the second premise, then the conclusion is labelled $A \otimes B$ (resp. $A \wp B$);
- Nodes of label $\mathbf{ax}$ have no premise and two conclusions. If the label of the first conclusion is A , the label of the second conclusion is $A^\perp$;
- Nodes of label $\mathbf{cut}$ have two premises and no conclusion. If the first premise is labelled A , the second premises is labelled $A^\perp$;
- Nodes labelled $\mathbf{c}$ have one premise and no conclusion³;
- Every edge is premise of one of its endpoints and conclusion of the other.

Definition 2 (Proof net). *To any MLL proof, one associates a proof structure in a trivial way. A **proof net** is any proof-structure which is image of a sequent proof in this way.*

Proof nets are inductively defined on the structure of sequent proofs by case on the last rule of the proof:



Remark 1. *In the graphical representation of proof nets, we put arrows on edges to represent the information on premise/conclusion, but we consider the graph as undirected, in particular with respect to any notion such as paths, cycles, ...*

The graph in figure 1 is indeed a proof structure but it cannot be associated with a MLL proof. A proof structure therefore does not necessarily correspond to a sequent calculus proof; such a proof structure is called non-sequentialisable. There is a number of methods to distinguish sequentialisable proof structures – proof nets – from non sequentialisable ones; such a method is called a *correctness criterion*.

3 Correctness Criteria

In this section, we shall mainly focus on Danos-Regnier and contractibility which will be used in the latter part of the paper.

³We shall often omit those nodes in the graphical representation of nets: they will be depicted as pending edges.

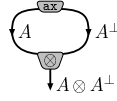


Figure 1: A PROOF STRUCTURE WHICH IS NOT A PROOF NET.

3.1 Danos-Regnier Criterion

Definition 3 (Switching). A **switching** of a proof structure R is the choice, for every $\wp$ node of the graph, of one of its premises. More formally, a switching of R is a map from the $\wp$ nodes of R to $\{l, r\}$.

Given a switching s of a proof structure R , a $\wp$ node n will be said to be switched to the right (resp. to the left) if the right premise (resp. left) has been selected, that is if $s(n) = r$ (resp. l).

Definition 4 (Correction graph). A **Correction graph** of a proof structure $R = (V_R, E_R)$ and a S of R is the undirected graph $S(R) = (V_{S(R)}, E_{S(R)})$ defined as $V_{S(R)} = V_R$ and $E_{S(R)}$ is the subset of edges of R containing all edges from R but for the left (resp. right) premise of a $\wp$ node n when $S(n) = r$ (resp. l) and such that the labels are the inherited from R .

Definition 5 (Danos-Regnier criterion (DR)). A proof structure satisfies the Danos-Regnier criterion if every correction graph is connected and acyclic; in that case, it is said to be DR-correct.

Theorem 1. A proof structure is a proof net if, and only if, it is DR-correct.

Danos-Regnier criterion is one of the most standard correctness criteria for its beauty and because it is convenient to reason about, but it has an exponential complexity. We now introduce the contractibility [3] criterion.

3.2 Contractibility criterion

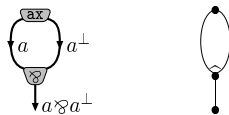
The correctness criterion we now consider directly expresses a topological property of the proof structure in terms of contractibility.

Definition 6 (Paired graph). A **paired graph** is given by a graph $G = (V, E)$ together with a set $P(G)$ of **paired edges**, that are undirected pairs of edges which share at least one endpoint. A node which is the endpoint of paired edges is called a **paired node**, the other endpoints are called the **premises**.

Remark 2. Note that a paired node may have 0, 1 or 2 premises.

Definition 7 ($C(R)$). To a proof structure R , one associates a paired graph, written $C(R)$, which is simply R together with the set of paired edges given as the set of pairs of edges which are premises of a $\wp$ node.

Example. We show below the unique proof net $R_{a \wp a^\perp}$ for the sequent $\vdash a \wp a^\perp$ and the paired graph $C(R_{a \wp a^\perp})$ which is associated to $R_{a \wp a^\perp}$ (paired edges are distinguished by a $\frown$):



Definition 8 (Contraction rules). *One defines two graph-rewriting rules on paired graphs as follows (note that in rule R_2 , the two nodes shall be distinct):*



Definition 9 (Contractibility). *A proof structure R is **contractible** if*

$$C(R) \rightarrow^* \bullet.$$

Contractibility characterizes proof nets, it provides a correctness criterion:

Theorem 2. *A proof structure is a proof net if, and only if, it is contractible.*

4 Dependency graph criterion

In this section, we will introduce a new correctness criterion. Since it is based on Mogbil-Naurois criterion, we first introduce in [4.1](#) this recently introduced criterion before defining the new, switching-independent, criterion. This is done stepwise. First in section [4.2](#), we shall define a new notion of dependency graph which does not rely on switchings, thus obtaining a necessary condition for a proof structure to be a proof net. Finally, we make free of switching in section [4.3](#) where we consider why we really need the first part of the criterion, showing that it can be replaced by a necessary condition far more simple and which does not rely on switchings. Those two necessary conditions put together result in a sufficient condition as we will show in section [4.4](#). We conclude the section by comparing the two versions of dependency graphs in subsection [4.5](#).

4.1 Mogbil-Naurois Criterion

Mogbil and Jacobé de Naurois were interested in characterizing the space-complexity of correctness of M(E)LL proof structures, showing it is NL-complete [4](#).

Definition 10 (Elementary path). *A path in a undirected graph is **elementary** when it does not enter twice the same edge.*

Definition 11 (Dependency graph of a correction graph). *Given R a proof structure and $\mathcal{S}$ a switching of R , the dependency graph of $\mathcal{S}(R)$, written $D(\mathcal{S}, R)$ is an oriented graph (V, E) defined as follows:*

- *The set of nodes V consists of the set of conclusions of $\wp$ nodes of R together with an additional node s .*
- *Let x be a $\wp$ node in R , x_r (resp. x_l) its right (resp. left) premise in R .*
 - *There is an edge $(s \rightarrow x)$ if there exists an elementary path $x_l, \dots, x_r$ in $\mathcal{S}(R)$ which goes through no $\wp$.*
 - *Let y be another $\wp$ node in R . There is an edge $(y \rightarrow x)$ if there exists an elementary path $x_l, \dots, x_r$ in $\mathcal{S}(R)$ containing y .*

⁴NL denotes the class of problems which can be decided in logarithmic space by a non-deterministic Turing machine.

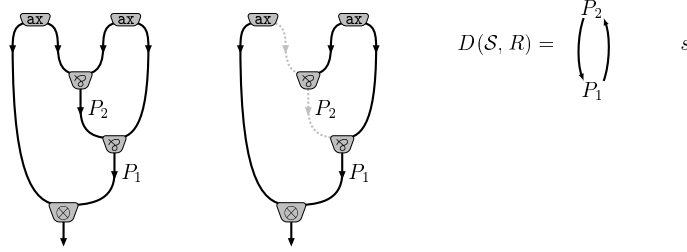


Figure 2: AN INCORRECT STRUCTURE R , A SWITCHING $\mathcal{S}$ OF R AND THE ASSOCIATED DEPENDENCY GRAPH (CYCLIC AND WITH NO SOURCE NODE).

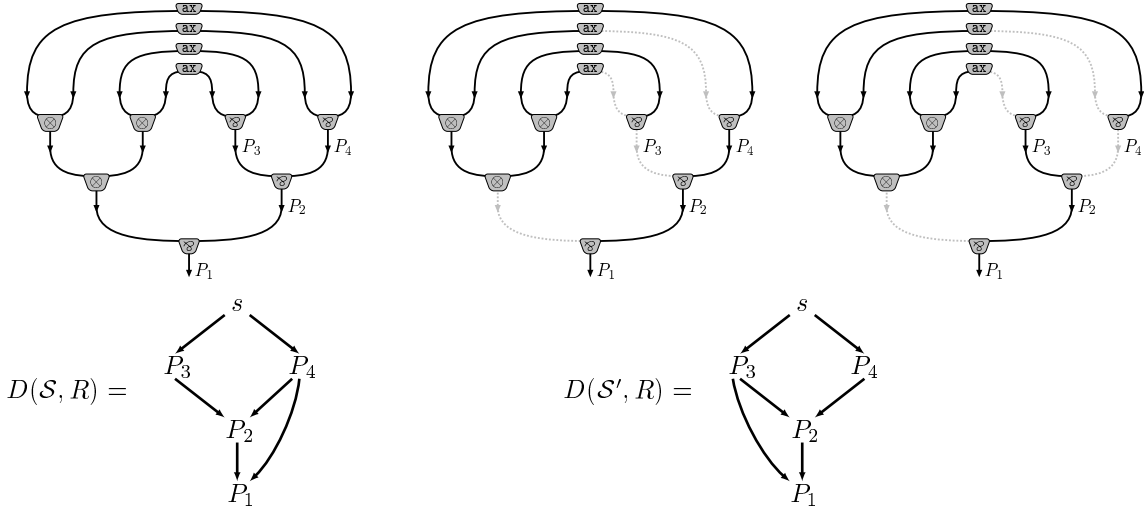


Figure 3: NET R , SWITCHINGS $\mathcal{S}$ $\mathcal{S}'$, THE ASSOCIATED DEPENDENCY GRAPHS.

Definition 12 (*SDAG graphs*). A graph G is *SDAG* if: it is *acyclic* and it contains a node s , the *source node*, such that for every node n of G , different from s , there exists a path from s to n .

Definition 13 (*Mogbil-Naurois Criterion*). A proof structure satisfies the *Mogbil-Naurois criterion (MN)* if there exists a connected and acyclic switching $\mathcal{S}$ such that $D(\mathcal{S}, R)$ is *SDAG* of source s . Such a proof structure is said to be **MN-correct**.

Theorem 3. A proof structure is a proof net if, and only if, it is *MN-correct*.

Figures 2 and 3 illustrate how the criterion discriminates between correct and incorrect structures. Note that on figure 3 the dependency graph depends on the switching. We will come back to this in section 4.5.

Compared to Danos-Regnier, the use of switchings in (MN)-criterion is quite odd: it only requires to analyze one switching and the corresponding correction graph. Moreover, the choice of this switching is itself arbitrary.

It is therefore natural to wonder the true role of this switching: is it really necessary? We answer this question in the following by stating a dependency-graph based criterion which does not rely on any switching.

4.2 A switching-free dependency graph

We show how to define dependency graphs, directly on the proof structure rather than on its correction graphs.

Definition 14 (*s-path in a paired graph*). *A path in a paired graph is called an s-path if it is elementary and if it does not go successively through two edges which are paired together.*

Notation. *If p is such an s-path, we also write $p[e_1, \dots, e_n]$ or $p[E]$ to indicate the paired edges it contains ($\{e_1, \dots, e_n\}$ and E being the paired edges occurring in p).*

Definition 15 (*Dependency graph of a paired graph*). *Let G be a paired graph. The dependency graph, $D(G)$, of G is the oriented graph (V, E) defined as follows:*

- *The set of nodes V consists in the set $P(G)$ of paired edges of G together with an additional node s .*
- *Let $e = (e_1, e_2)$ be an element of $P(G)$.*
 - *There is an edge $(s \rightarrow e)$ in E if none of e_1, e_2 are loops and the premises of the common endpoint are connected by an s-path in G which goes through no paired edge.*
 - *Let $f = (f_1, f_2)$ be another element of $P(G)$. There is an edge $(f \rightarrow e)$ if the premise(s) of a common endpoint of e_1, e_2 are connected by an s-path containing f_1 or f_2 .*

Remark 3. *Notice that:*

- *In both cases of the previous definition, when e_1 and e_2 have both of their endpoints in common (let n_1, n_2 be those endpoints), there is an ambiguity on the definition of which premises we consider: n_1 has $\{n_2\}$ as set of premises and conversely n_2 has $\{n_1\}$ as set of premises. Anyway, in the case of an edge $(s \rightarrow e)$ there is an empty s-path connecting either premise to itself and in the other case we only ask that there is an s-path via f in either connecting either of the sets of premises.*
- *In the case of an edge $(s \rightarrow e)$, if there is an edge $(s \rightarrow e)$, that means that there is a s-path of the form $p[\emptyset]$ between the premises of common endpoints.*
- *In the case of an edge $(f \rightarrow e)$, if there is an edge $(f \rightarrow e)$, that means that there is a s-path of the form $p[\{y_i\} \cup F]$ between the premises of common endpoints.*

Definition 16 (*Dependency graph of a proof structure*). *Let R be a proof structure. The dependency graph, $D(R)$, of R is the dependency graph of its paired graph $C(R)$.*

Remark 4. *Notice that the nodes of the dependency graph of a proof structure R , which are the paired edges of its paired graph, are in one-to-one correspondence with the conclusions of $\bowtie$ nodes of R . One will use indifferently one or the other representation of the nodes for $D(R)$.*

Definition 17 (*Modified Mogbil-Nauois criterion*). *A proof structure satisfies the modified Mogbil-Nauois criterion (MN') if:*

- *its dependency graph $D(R)$ is SDAG of source s ;*
- *one of its correction graphs is connected and acyclic.*

(MN') is necessary for a proof structure to be sequentializable:

Theorem 4. *A proof net satisfies (MN').*

We show this result by considering a relation on $\wp$ nodes induced by the order in which inferences are used in one of the sequentializations of R , which will force acyclicity of the dependency graph. Once acyclicity established, the source is easily obtained from the fact that at most one node has no ancestor.

Definition 18 (Order of introduction). *Let π be an MLL proof. For every $\wp$ rule $\mathbf{r}_F$ introducing formula F , we note π_F the sub-tree of π rooted in the premise of $\mathbf{r}_F$. We define a partial order on the formulas introduced by $\wp$ inferences in π , that will be noted $<_\pi$ as follows: $F <_\pi G$ if $\mathbf{r}_F \in \pi_G$.*

It makes formal the relation “to be introduced above”.

The graph of this relation is noted $O^-(\pi)$ and one defines $O(\pi)$ as $O^-(\pi)$ augmented by adding a vertex s and, for all vertex e in $O^-(\pi)$, an edge $s \rightarrow e$.

Theorem 5. *Let π be an MLL proof and R the corresponding proof net. Then: $D(R) \subseteq O(\pi)$.*

Lemma 1. *Let π be an MLL proof. $O(\pi)$ is acyclic.*

Lemma 2. *Let G be a directed and acyclic graph. If every node of G has a parent but for one, written s , then G is SDAG of source s .*

4.3 On the importance of the witness of acyclicity and connectedness

The last bit of switching that remains is the witness of acyclicity and connectedness that is given by the arbitrary switching considered in the first part of the (MN) and (MN').

Actually, the important remark here is that if there exists a connected and acyclic switching, in any switching there is one more connected components than there are cycles. This number is characteristic of the proof structure and is actually independent of the switching itself as we shall make clear.

Actually, the difference between the number of connected components and the number of cycles can be characterized independently of the switching graphs: it is also $\#ax - \#\otimes$. From this remark follows immediately the last part of our criterion.

Definition 19. *Let G be an undirected graph and n, e be its numbers of nodes and edges. We set $\chi_G = n - e$ and call this quantity the **characteristic** of the graph.*

Theorem 6. *Let G be an undirected acyclic graph and c_G be the number of its connected components. One has: $\chi_G = c_G$.*

Proposition 1. *For every correction graph G of a proof net, one has $\chi_G = 1$.*

Proposition 2. *Every correction graph G of a proof structure R satisfies: $\chi_G = \#ax - \#\otimes$.*

We note γ_R this number and call it the characteristics of the structure.

Putting the two previous propositions together, a sequentializable proof structure must have one more axiom links than tensor links: $\#ax - \#\otimes = 1$.

Remark 5. *When a structure contains cuts, one has $\chi_G = \#ax - \#\otimes - \#\text{cut}$ for every correction graph G . The condition above becomes $\#ax - \#\otimes - \#\text{cut} = 1$.*

We can finally state our new criterion, *DepGraph*:

Definition 20 (DepGraph criterion). *A proof structure R is D-correct if*

- (1) $D(R)$ is SDAG, (2) R is connected and (3) $\gamma_R = 1$.

4.4 DepGraph is a correctness criterion

Connectedness and the characterization with axioms and tensors is of course necessary. There simply remains to show that D-correctness is a sufficient condition:

Theorem 7. *If a proof structure is D-correct, then it is a proof net.*

We will prove that a D-correct proof structure satisfies contractibility criterion, from which the result follows. We need the following lemma:

Lemma 3. *Contractibility steps preserve:*

1. *connectedness of the contracted graph;*
2. *the number $n - e_{npe} - e_{pe}$ (with n the number of nodes of the graph, e_{npe} the number of non-paired edges and e_{pe} the number of paired edges).*

Lemma 4. *Let G be a paired graph and G' be such that $G \rightarrow^* G'$. Let $\{b_1, \dots, b_m\}$ be the paired edges which have been contracted and let x and y be two nodes in G . If x and y are connected by an s -path $p[a_1, \dots, a_n]$ in G , then:*

- *either x and y have been merged;*
- *or there exists an s -path $q[\{a_1, \dots, a_n\} \setminus \{b_1, \dots, b_m\}]$ connecting x and y in G' .*

Conversely, every s -path of G' is the residual of an s -path in G .

4.5 Comparison between the two versions of dependency graphs

Example on figure 3 showed that Mogbil-Naurois dependency graphs were switching-dependent. We will show that, for proof nets, we almost have this invariance. More precisely, the transitive closures of the dependency graphs induced by the various switchings are equal and they are equal to the transitive closure of the dependency graph we introduced in this paper: all these graphs induce the same order.

Notations. *If $\mathcal{S}$ is a switching for a proof structure R and a a $\wp$ -link in R , we note $\mathcal{S}_a$ the switching $\mathcal{S}$ in which we have toggled the switching for a .*

Given a graph D , D^ is its transitive closure.*

Lemma 5. *Let z and a be two $\wp$ links of a proof net R and $\mathcal{S}$ be a switching.*

- *if $(z \rightarrow a) \in D(\mathcal{S}, R)$, then $(z \rightarrow a) \in D(\mathcal{S}_a, R)$*
- *if $(a \rightarrow z) \in D(\mathcal{S}, R)$, then $(a \rightarrow z) \in D(\mathcal{S}_a, R)$*

Theorem 8. *Let R be a DR-correct proof structure. For all switchings $\mathcal{S}, \mathcal{S}'$ of this structure, we have $D(\mathcal{S}, R)^* = D(\mathcal{S}', R)^*$.*

Only the last part of the result remains to be proved:

Theorem 9. *Let R be a proof net and $\mathcal{S}$ a switching for R . Then: $D(R)^* = D(\mathcal{S}, R)^*$.*

5 Conclusion

Comparison with (MN) criterion DepGraph criterion defined in section 4.2 is based on the notion of dependency graph introduced in [5] which we defined on the proof structure rather than on the correction graphs. Moreover, the different parts of our criterion clearly separate conditions on $\wp$ inferences from other inferences.

The proof of the sequentialization theorem evidence the fact that condition (SDAG) on the dependency graph is of a different nature than the two other conditions which constitute our criterion [20]. Indeed, when considering the relationships between DepGraph and contractibility, we see that (SDAG) prevents deadlocks that could result from bad configurations of paired edges, while the two other conditions ensure that the normal form is indeed reduced to a single point with no loop (connectedness plus characteristic equal to 1).

In particular, the natural order induced by the dependency graph of a correct structure provides useful information on the causality between $\wp$ connectives. Theorem 5 shows that every sequentialization of proof net shall at least be compatible with this order.

Perspectives A future direction of our work will be to investigate in more details the connections between dependency graph and causality in proofs. It would be interesting to prove a converse of theorem 5: if all sequentializations of a net have a given $\wp$ connective introduced below another one, this should be reflected in (a variant of) the dependency graph.

Theorems 8 and 9 are encouraging since the dependency graph actually reflects an invariant of the sequentializations, the order induced by the dependency graph.

References

- [1] Marc Bagnol, Amina Doumane, and Alexis Saurin. long version of this submission. available at <http://www.pps.univ-paris-diderot.fr/~saurin/Publi/DepGraphLong.pdf>.
- [2] Pierre-Louis Curien. Introduction to linear logic and ludics, part ii, 2006.
- [3] Vincent Danos. *Une application de la logique linéaire à l'étude des processus de normalisation (principalement du λ -calcul)*. Thèse de doctorat, Université Denis Diderot, Paris 7, 1990.
- [4] Vincent Danos and Laurent Regnier. The structure of multiplicatives. 28:181–203, 1989.
- [5] Paulin Jacobé de Naurois and Virgile Mogbil. Correctness of linear logic proof structures is NL-complete. *Theor. Comput. Sci.*, 412(20):1941–1957, 2011.
- [6] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1 – 101, 1987.
- [7] Jean-Yves Girard. *Le Point Aveugle: Cours de logique. Tome 1, Vers la perfection; Tome 2, Vers l'imperfection*. Hermann, 2006.
- [8] Stefano Guerrini. Correctness of multiplicative proof nets is linear. In *LICS*, pages 454–463. IEEE Computer Society, 1999.
- [9] Stefano Guerrini. A linear algorithm for MLL proof net correctness and sequentialization. *Theor. Comput. Sci.*, 412(20):1958–1978, 2011.
- [10] William A. Howard. The formulae-as-type notion of construction, 1969. In J. P. Seldin and R. Hindley, editors, *To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism*, pages 479–490. Academic Press, New York, 1980.
- [11] Andrzej S. Murawski and C.-H. Luke Ong. Fast verification of mll proof nets via imll. *ACM Trans. Comput. Logic*, 7(3):473–498, July 2006.
- [12] Misao Nagayama and Mitsuhiro Okada. A new correctness criterion for the proof nets of non-commutative multiplicative linear logics. *J. Symb. Log.*, 66(4):1524–1542, 2001.