On the dependencies of logical rules

Marc Bagnol[1], Amina Doumane[2], and Alexis Saurin[2] *

[1] IML, Université d'Aix-Marseille, bagnol@iml.univ-mrs.fr
[2] PPS, CNRS, Université Paris Diderot & INRIA,
{amina.doumane,alexis.saurin}@pps.univ-paris-diderot.fr

Abstract. Many correctness criteria have been proposed since linear logic was introduced and it is not clear how they relate to each other. In this paper, we study proof-nets and their correctness criteria from the perspective of *dependency*, as introduced by Mogbil and Jacobé de Naurois. We introduce a new correctness criterion, called DepGraph, and show that together with Danos' contractibility criterion and Mogbil and Naurois criterion, they form the three faces of a notion of dependency which is crucial for correctness of proof-structures. Finally, we study the logical meaning of the dependency relation and show that it allows to recover and characterize some constraints on the ordering of inferences which are implicit in the proof-net.

Keywords: Linear logic, MLL, Proof nets, Correctness criterion, Contractibility, Mogbil-Naurois Criterion, Permutability of inferences

1 Introduction

The benefits of Curry-Howard. Since the discovery of Curry-Howard correspondence [2], programming language theory and proof theory have been tightly intertwined. Among the numerous and fruitful back-and-forths between proofs and programs, linear logic [3] certainly stands as exemplary.

While working on second-order arithmetics, Girard introduced system F [4,5], a polymorphic λ-calculus. Studying the semantics of system F, he later introduced the coherent semantics [6] which led to the linear decomposition of implication ($A \Rightarrow B = \,!\,A \multimap B$), the cornerstone of linear logic [3] since this semantical observation turns to be syntactically reflected in a well-behaved proof system. With linear logic came a very canonical representation of proofs (for fragments) of linear logic: proof nets [3,7,8] 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. The beauty of proof-nets is especially striking in the multiplicative fragment with no logical constant (also said unit-free multiplicative logic) to which most of the paper will be dedicated.

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

Proof-nets and logical correctness. By moving from inductive objects (*e.g.* sequent calculus proof trees) to more geometrical objects (proof structures), correctness becomes a global property contrarily to sequent proofs where correctness was local (a proof is correct if every step in the argument is correct). 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. From this comes the need for conditions to ensure the logical correctness of proof structures. Several correctness criteria have been introduced in the literature. Among the best-known criteria, one can refer to the original long-trip criterion (LT) [3], Danos-Regnier criterion (DR) [8], counter-proofs criterion (CP) [9,10], contractibility (C) [11], graph-parsing criterion (GP) [12,13], Dominator Tree (DT) [14] and more recently Mogbil-Naurois criterion (MN) [15].

Relating correctness criteria. Actually, correctness criteria usually provide us with some specific viewpoints on the proof-theoretical or computational properties of proofs. For instance, they can (i) provide precise means to sequentialize a proof-net into a sequent proof, or (ii) tell us about the complexity of the correctness problem, or even (iii) say something about the structure of proofs.

Although correctness of proof-nets is now well-studied and understood, the question of comparing and relating those criteria attracted much less attention.

Contributions of the paper. The present paper is a contribution in this direction: we investigate a notion of dependency between inferences of a proof structure and use it to compare three correctness criteria (C, MN and DepGraph, a new criterion we introduce here) showing that they constitute three faces of this dependency relation.

We reformulate Contractibility in a big-step version from which arises the notion of dependency that one finds in MN criterion. This leads us to introduce a new criterion, DepGraph. We then show that these three criteria, arising from the notion of dependency, meet the three categories given above: we show that Contractibility gives actually a sequentialization of a proof-net, MN is a criterion with efficiency purposes and DepGraph emphasizes the structural properties of logic since (i) it deals separately with positive and negative inferences, suggestion possible connections with focusing, (ii) it is switching-independent, contrarily to MN, (iv) it makes use of a well-known necessary condition for correction following from Euler-Poincaré property [10] and finally (iv) we use its notion of dependency in order to characterize constraints on the order of introduction of inferences which are shared by all sequentializations of a given proof-net.

We focus on multiplicative and unit-free linear logic. Rather than a restriction of the results, this is a matter of presentation: DepGraph criterion can easily be extended to MELL, thus capturing typed lambda-calculus

Organization of the paper. In Section 2, we recall the basics of proof nets and correctness criteria and dedicate Section 3 to analyzing and comparing the three criteria mentioned above by (i) showing how contractibility is related with sequentialization, (ii) formulating a big-step notion of contractibility, (iii) justifying the occurrence of a dependency relation in proof-nets, (iv) introducing a new correctness criterion, DepGraph and (iv) comparing DepGraph with MN-criterion. We finally focus in Section 4 on the logical meaning of dependency graphs. Due to lack of space, proofs are omitted but can be found in an extended version with supporting proofs and more material, available online in [1].

2 Correctness problem of proof structures in linear logic

2.1 Linear logic and proof nets

MLL. In this paper, we will deal with multiplicative linear logic (MLL), which is a fragment of linear logic. MLL formulas are built from the following grammar:

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

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 $(ax), (cut), (\otimes), (\parr)$ and edges are labelled by sequents as follows:

Identity Group:	$\dfrac{}{\vdash A, A^{\perp}}\ (ax)$	$\dfrac{\vdash A, \Gamma \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ (cut)$
Multiplicative Group:	$\dfrac{\vdash A, \Gamma \qquad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta}\ (\otimes)$	$\dfrac{\vdash A, B, \Gamma}{\vdash A \parr B, \Gamma}\ (\parr)$

Sequent calculus induces a sometimes irrelevant order between inferences. This is evidenced by possible permutations between inferences of a sequent proof. We recall in figure 1 the main cases of these permutations, the other cases are much alike, included the permutations involving the cut inference.

$$\dfrac{\dfrac{\vdash A, C, \Gamma \qquad \vdash D, \Delta}{\vdash A, C \otimes D, \Gamma, \Delta}\ (\otimes) \qquad \vdash B, \Sigma}{\vdash A \otimes B, C \otimes D, \Gamma, \Delta, \Sigma}\ (\otimes) \quad \leftrightarrow \quad \dfrac{\dfrac{\vdash A, C, \Gamma \qquad \vdash B, \Sigma}{\vdash A \otimes B, C, \Gamma, \Sigma}\ (\otimes) \qquad \vdash D, \Delta}{\vdash A \otimes B, C \otimes D, \Gamma, \Delta, \Sigma}\ (\otimes)$$

$$\dfrac{\dfrac{\vdash A, B, C, D, \Gamma}{\vdash A \parr B, C, D, \Gamma}\ (\parr)}{\vdash A \parr B, C \parr D, \Gamma}\ (\parr) \quad \leftrightarrow \quad \dfrac{\dfrac{\vdash A, B, C, D, \Gamma}{\vdash A, B, C \parr D, \Gamma}\ (\parr)}{\vdash A \parr B, C \parr D, \Gamma}\ (\parr)$$

$$\dfrac{\vdash A, \Gamma \qquad \dfrac{\vdash B, C, D, \Delta}{\vdash B, C \parr D, \Delta}\ (\parr)}{\vdash A \otimes B, C \parr D, \Gamma, \Delta}\ (\otimes) \quad \leftrightarrow \quad \dfrac{\dfrac{\vdash A, \Gamma \qquad \vdash B, C, D, \Delta}{\vdash A \otimes B, C, D, \Gamma, \Delta}\ (\otimes)}{\vdash A \otimes B, C \parr D, \Gamma, \Delta}\ (\parr)$$

Fig. 1: Key cases of inference permutations in the sequent calculus.

Proof structures and proof nets. Proof nets are canonical representations of MLL sequent proofs quotienting them by the previous permutation rules, resulting in a confluent cut-elimination and other very good properties. Proof structures allow to present MLL proofs in a non-sequential way and therefore those objects are not inductively presented anymore which makes the checking of the logical correctness of those object challenging, calling for correctness criteria.

In the following, we shall consider only cut-free proof structures. Indeed, cut behaves exactly as $\otimes$ from the view point of correctness and therefore introduces no difficulty nor interesting aspects in our developments.

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 **c** (for the conclusions of the proof) and edges are labelled with formulas of MLL. Moreover, edges which are adjacent to a vertice are partitioned into premises and conclusions according to the following rules:*

- *Nodes of label $\otimes$ (resp. $\parr$) 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. $A \parr B$);*
- *Nodes of label **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 labelled **c** have one premise and no conclusion[3].*
- *Every edge is premise of one of its endpoints and conclusion of the other.*

Definition 2 (Desequentialization). *To any MLL proof π, one associates a proof structure $[\pi]$, its desequentialization, by forgetting the order of the inference rules and keeping only the subformula ordering together with the axiom links.*

Definition 3 (Proof net). *A* **proof net** *is any proof-structure which is the desequentialization of some sequent proof.*

By the previous definition, one immediately get an inductive characterization of proof nets. Proof nets are those proof structures which can be obtained inductively as in figure 2

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, ...

2.2 Correctness criteria

The graph in figure 3 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 sequentializable proof structures –

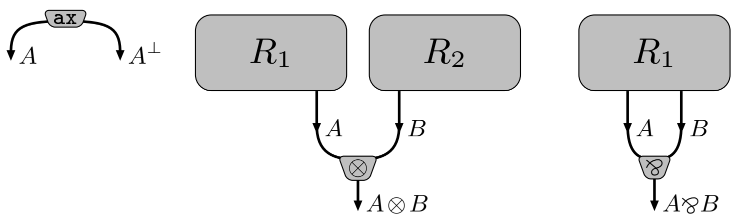


Fig. 2: Inductive characterization of proof nets.

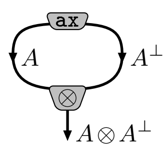


Fig. 3: A proof structure which is not a proof net.

proof nets – from non sequentialisable ones; such methods are called *correctness criteria*.

Several correctness criteria have been introduced in the literature. In the rest of this section, we shall present Danos-Regnier (DR) criterion which is one of the most popular criteria; then we present Contractibility and Mogbil-Naurois (MN) criterion which we will compare in the next section.

2.3 Danos-Regnier Criterion

Definition 4 (Switching). *A* **switching** *of a proof structure R is the choice, for every $⅋$ node of the graph, of one of its premises. More formally, a switching of R is a map from the $⅋$ nodes of R to $\{l, r\}$.*

Given a switching s of a proof structure R, a $⅋$ 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 $s(n) = l$ (resp. r).

Definition 5 (Correction graph). *A* **Correction graph** *of a proof structure $R = (V_R, E_R)$ and a $\mathcal{S}$ of R is the undirected graph $\mathcal{S}(R) = (V_{\mathcal{S}(R)}, E_{\mathcal{S}(R)})$ defined as $V_{\mathcal{S}(R)} = V_R$ and $E_{\mathcal{S}(R)}$ is the subset of edges of R containing all edges from R but for the left (resp. right) premise of a $⅋$ node n when $\mathcal{S}(n) = r$ (resp. l) and such that the labels are the inherited from R.*

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

Definition 6 (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.*

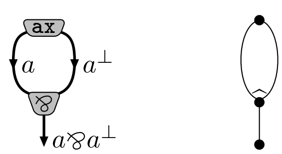
2.4 Contractibility

Contractibility criterion expresses a topological property of the proof structure, more precisely of an underlying graph structure, the paired graph which contains just enough information to distinguish premises of a $\parr$ from the other edges.

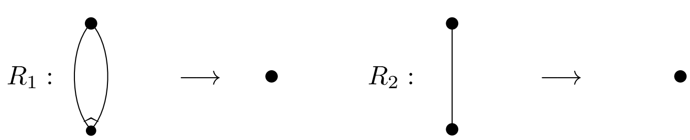
Definition 7 (Paired graph). *A* **paired graph** *is given by a graph $G = (V, E)$ together with a set $P(G)$ of* **paired edges**, *that are undirected pairs of edges which share at least one endpoint.*

Definition 8 ($C(R)$). *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 $\parr$ node.*

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



Definition 9 (Contraction rules). *One defines two graph-rewriting rules on paired graphs as follows (note that in both rules the two nodes shall be distinct):*



Definition 10 (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.*

2.5 Mogbil-Naurois criterion

We shall first present Mogbil-Naurois criterion, one of the most recent correctness criteria which characterized the space-complexity of the correctness problem.

Definition 11 (Elementary path). *A path in a undirected graph is* **elementary** *when it does not enter twice the same edge.*

Definition 12 (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 in the set of conclusions of $\parr$ nodes of R together with an additional node s.*
- *Let x be a $\parr$ node in R, x_r (resp. x_l) its right (resp. left) premise in R.*
  - *There is an edge $(s \rightarrow x)$ in E if there exists an elementary path $x_l, \ldots, x_r$ in $\mathcal{S}(R)$ which goes through no $\parr$ node.*
  - *Let y be another $\parr$ node in R. There is an edge $(y \rightarrow x)$ if there exists an elementary path $x_l, \ldots, x_r$ in $\mathcal{S}(R)$ containing y.*

Definition 13 (*SDAG* graphs). *A graph G is SDAG if: it is acyclic and it contains a node s, the source, such that all nodes of G are accessible from s.*

Definition 14 (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. Such a proof structure is said* **MN-correct***.*

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

One notices that dependency graphs are defined on correction graphs and thus they depend on the switching. 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 what is the exact role of this switching: is it really necessary? We answer this question in the following by going back to the origin of the idea of dependence, which was already present in the contractibility criterion as we shall see in section 3. From that point, we state a dependency-graph based criterion which does not rely on any switching.

3 On the three faces of contractibility

Despite the wide diversity of correctness criteria, their relationship remains poorly studied in the literature. In this section, we shall investigate the connections between three criteria: Mogbil-Naurois, Contractibility and DepGraph which is a new criterion that we introduce in the remainder.

3.1 Contractibilty and sequentialization

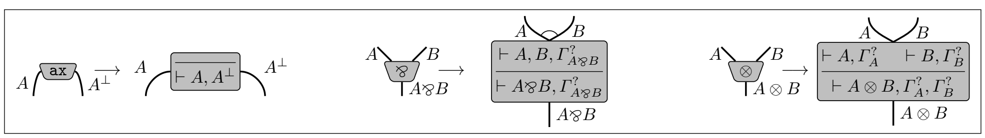
Before relating contractibility with the other two criteria, we make clear that it gives a genuine sequentialization. To do this, we simply label nodes of the paired graph of the proof structures with open proofs containing context variables. These open proofs correspond to partial sequentializations, which become larger and larger as contraction progresses, until reaching a full MLL proof. More precisely, these open proofs are constructed on sequents with context variables, generated by the following syntax (F is a formula and $\Gamma^?$ is a context variable):

$$S := \quad \emptyset \quad | \quad S, F \quad | \quad S, \Gamma^?$$

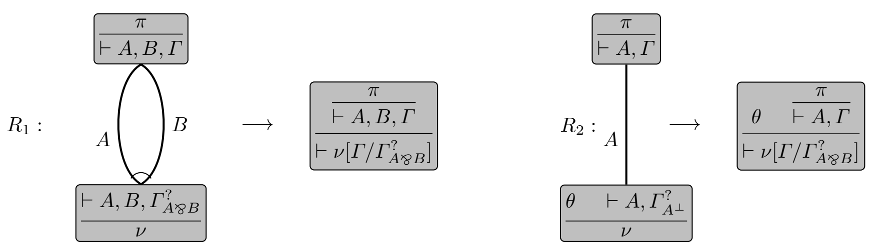
We consider these sequents up to commutativity. Open proofs are constructed by the following syntax:

$$\frac{}{\vdash A \parr A^\perp}\ (ax) \qquad \vdash S \qquad \frac{\vdash S_1, A \quad \vdash S_2, B}{\vdash S_1, S_2, A \otimes B}\ (\otimes) \qquad \frac{\vdash S, A, B}{\vdash S, A \parr B}\ (\parr)$$

Given a proof structure R, the labelled paired graph $C_l(R)$ is obtained by applying the following rules:



Labeled contractibility rules become:



If R is actually a proof net, the node at which its paired graph contracts is labeled by one of its sequentializations (see long version for an example):

Proposition 1. *Let R be a proof structure. If $C(R)$ contracts (by rules R_1 and R_2) to a point, then by following the same contraction path, $C_l(R)$ contracts to a point whose label is a sequentialization of R.*

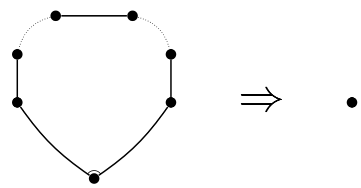
Notice that two different contraction paths may lead to different sequentializations of a proof net.

3.2 Big-step contractibility

We reformulate Contractibility in a big-step fashion to highlight the intrinsic notion of dependency present in this criterion.

One defines a new graph-rewriting rule R as follows:

Definition 15 (Big-step Contraction R). *An elementary cycle can be contracted to a point if it contains exactly two paired edges that are paired together that are adjacent in the cycle.*



This new notion of contractibility is easily seen to correspond to usual contractibility and thus induces a correctness criterion expressed as:

Theorem 4. *A proof structure is a proof net if, and only if, contraction R can be applied until:*

- *no paired edges are left and*
- *it leads to a tree of unpaired edges.*

3.3 Towards dependency graphs

This version of contractibility criterion induces a natural dependency relation between the $⅋$ nodes of the proof structure: when the premises of a $⅋$ node are connected by a path that does not go through any premise of an other $⅋$ node (see figure 4), one can contract directly this path; these are the nodes connected at the source in the dependency graph of MN-criterion. When, on the contrary, the path from the premises of a $⅋$ node ($⅋_1$) goes through one of the premises of another $⅋$ node ($⅋_2$) (see figure 4), we say that $⅋_1$ depends on $⅋_2$ because $⅋_1$ can only be contracted if $⅋_2$ is contracted before. In this way, we can construct a dependency graph which looks like the dependency graph of MN criterion, but this one is built directly on the proof structure rather than on a correction graph. The first condition of big-step contractibilty criterion says simply that this graph is SDAG. We will see how to transform the second condition in order to get a full correction criterion. Before moving to the study of this new criterion, let us simply remark that one can actually define a dependency relation between $⅋$ nodes of a proof structure R and any set of nodes of R as follows:

Definition 16 (Dependency graph of a proof structure, relatively to a set of nodes). *Let R be a proof structure and N a set of nodes of R. We denote by P the set of $⅋$ nodes of R. The dependency graph of R relatively to N, $D_N(R)$, is the oriented graph (V, E) defined as follows:*

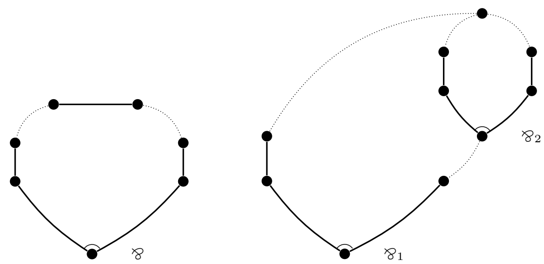


Fig. 4: Various dependency configurations.

- *$V = N \cup P \cup \{s\}$ where s is an additional node.*
- *Let p be an element of P.*
  - *There is an edge $(s \to p)$ in E if the premises of p are connected by an elementary path in R which goes through no ⅋ node.*
  - *Let q be an element of V. There is an edge $(q \to p)$ if the premises of p are connected by an elementary path containing q which does not go successively through the two premises of a ⅋ node.*

Remark 2. The intuition underlying this exdended notion of dependency graph is that in big-step contractibility, the contraction of the paired graph depends not only on depdency between paired edges, but also on the fact that the $\otimes$ nodes on the cycles actually can be contracted to a point (with no loop), thus making a ⅋ node depend on a $\otimes$ node.

Notation. *The previous definition has two natural instances: when we take N to be the set of the ⅋ nodes of a proof structure N, $D_N(R)$ is a graph which expresses the dependency relation between ⅋ nodes only. We note it by $D_{⅋}(R)$. When N is taken to be the set of all ⅋ and $\otimes$ nodes of a proof structure, $D_N(R)$ is a graph which expresses the dependency relations between the ⅋ nodes and the other ⅋ and $\otimes$ nodes. We denote it by $D_{⅋,\otimes}(R)$.*
In the following we shall consider only $D_{⅋}(R)$ until section 4 where $D_{⅋,\otimes}(R)$ will be considered. When there is no ambiguity will shall omit the subscript.

3.4 DepGraph criterion

As said before, the first condition of big-step contractibility expresses that $D_{⅋}(R)$ is SDAG: the existence of a contractibility sequence ensures that there is some ⅋ node having a cycle that does not contains any paired edges which is the condition to be connected to the source, while the acyclicity condition ensures that we will always find a ⅋ node with a cycle that can be contracted.

To get a full correction criterion, we will make use of a graph theoric property called Euler-Poincaré lemma, as suggested by Girard in [10].

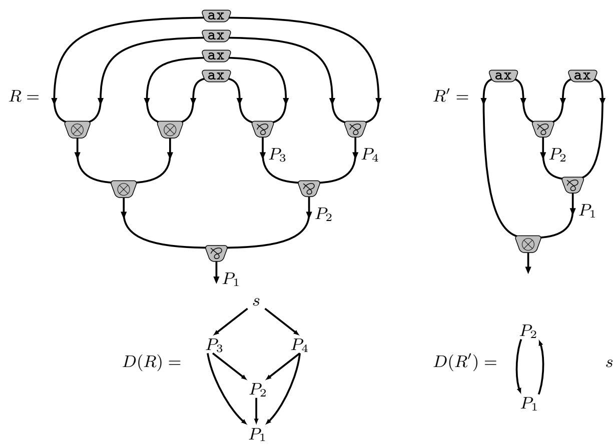


Fig. 5: EXAMPLES OF DEPENDENCY GRAPHS.

Definition 17. *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 G.*

Theorem 5 (Euler-Poincaré Lemma). *Let G be an undirected acyclic graph and c_G be its number of connected components. The following equality holds:*

$$\chi_G = c_G.$$

Proposition 2. *For every correction graph G of a proof net, one has $\chi_G = 1$.*

Proposition 3. *Every correction graph G of a proof structure R satisfies:*

$$\chi_G = \#ax - \#\otimes.$$

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

Remark 3. When a structure contains cuts, one has $\chi_G = \#ax - \#\otimes - \#cut$ for every correction graph G. The condition above becomes $\#ax - \#\otimes - \#cut = 1$.

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

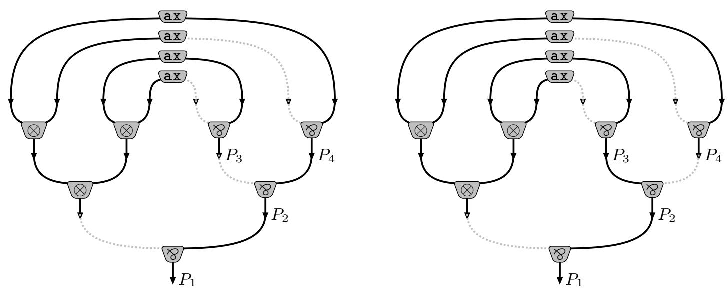
Definition 18 (DepGraph criterion). *A proof structure R is $D_{\parr}$-correct (or satisfies DepGraph criterion) if*

(1) $D_{\parr}(R)$ is a SDAG, (2) R is connected and (3) $\#ax - \#\otimes = 1$.

Theorem 6. *A proof structure is a proof net if, and only if, it is $D_{\parr}$-correct.*

3.5 Comparing the two notions of dependency graphs

Example on figure 6 shows that Mogbil-Naurois dependency graphs are switching-dependent. We will show that, for proof nets, they are almost invariant: the transitive closure of the dependency graphs induced different switchings are all equal and are equal to the transitive closure of the dependency graph we introduced in the previous section.



$$D(\mathcal{S}, R) = \begin{array}{c} s \\ \swarrow \quad \searrow \\ P_3 \qquad P_4 \\ \searrow \quad \swarrow \\ P_2 \\ \downarrow \\ P_1 \end{array} \qquad D(\mathcal{S}', R) = \begin{array}{c} s \\ \swarrow \quad \searrow \\ P_3 \qquad P_4 \\ \searrow \quad \swarrow \\ P_2 \\ \downarrow \\ P_1 \end{array}$$

Fig. 6: Switchings $\mathcal{S}$ $\mathcal{S}'$ of net R, the associated dependency graphs.

Notations. *If $\mathcal{S}$ is a switching for a proof structure R and a a $⅋$-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 1. *Let z and a be two $⅋$ links of a proof net R and $\mathcal{S}$ be a switching.*

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

Theorem 7. *Let R be a proof net and $\mathcal{S}, \mathcal{S}'$ be switchings of R. Then we have*

$$D(\mathcal{S}, R)^* = D(\mathcal{S}', R)^*.$$

Remark 4. The proof relies strongly on the fact that in a connected acyclic graph, there always exists a single elementary path between two nodes. The result would not hold if the structure were not correct.

Finally, we have:

Theorem 8. *Let R be a proof net and $\mathcal{S}$ a switching for R. Then: $D_{⅋}(R)^\star = D(\mathcal{S}, R)^\star$.*

4 On the order of introduction of connectives in sequentializations

In this section, we will investigate the logical meaning of dependency graphs introduced for DepGraph criterion. A crucial step in proving that a proof net satisfies DepGraph is to show that if π is a sequentialization of proof net R, every dependence in the dependency graph is also present in the order of introduction graph, more precisely:

Definition 19 (Order of introduction). *Let π be an MLL proof. For every $\parr$ or $\otimes$ 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 $\parr$ or $\otimes$ inferences in π, that will be noted $<_\pi$ as follows:*

$$F <_\pi G \text{ if } \mathbf{r}_F \in \pi_G$$

It formalizes 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 vertice s and, for all vertice e in $O^-(\pi)$, an edge $s \to e$.

To show that every proof net is $D_\parr$-correct, we established the following:

Lemma 2. *Let π be an MLL proof and R its desequentialization. Then*

$$D_\parr(R) \subseteq O(\pi).$$

As a consequence, $D_\parr(R) \subseteq O(R) := \cap_{\pi,[\pi]=R} O(\pi)$ where $O(R)$ can be seen as the essence of the sequentalizations of R. It is natural to wonder whether this inclusion can be sharpened in an equality characterizing $O(R)$ and that would rely on our notion of dependency. Actually, $D_\parr$ expresses only the relationship betweep $\parr$ nodes, and it is not enough to characterize $O(R)$. We will use instead the dependency graph $D_{\parr,\otimes}(R)$ to take in acount also the dependency relation between $\parr$ and $\otimes$ nodes.

Definition 20 (Subformula graph of a proof net). *Let R be a proof net. The subformula graph of R, $SF(R)$, is the directed graph (V, E) defined as follows:*

- *$V = P \cup T$ where P and T are respectively the set of $\parr$ nodes and $\otimes$ nodes.*
- *Let n and m be two elements of V. There is an edge $(m \to n)$ in E if the formula of the conclusion of m is a sub formula of the formula of the conclusion of n.*

Theorem 9. *Let R be a proof net. Then $(D_{\parr,\otimes}(R) \cup SF(R))^\star = O(R)$.*

Proposition 4. *Let π be an MLL proof and R its desequentialization. Then:*

$$D_{\parr,\otimes}(R) \subseteq O(\pi).$$

Corollary 1. *Let R be a proof net. One has $(D_{\parr,\otimes}(R) \cup SF(R))^\star \subseteq O(R)$.*

To prove the other inclusion, we will use the fact that two MLL proofs have the same proof net if and only if they are obtained one from the other by the permutation rules introduced in section 2.1. More precisely, we will make use of the two following lemmas, proven in the long version of this paper:

Lemma 3. *Let π be an MLL proof and $\mathbf{r}_1$, $\mathbf{r}_2$ two consecutive rules in π introducing the formulas F_1 and F_2, such that $\mathbf{r}_1$ is above $\mathbf{r}_2$. If $\mathbf{r}_1$, $\mathbf{r}_2$ are not permutable by the previous three rules, then:*

$$(F_1 \to F_2) \in O([\pi])$$

Definition 21. *Let π be an MLL proof and $\mathbf{r}_1$,$\mathbf{r}_2$ two inference rules in π introducing the formulas F_1 and F_2, such that $\mathbf{r}_{F_1}$ is above $\mathbf{r}_{F_2}$. We note by $]\mathbf{r}_2, \mathbf{r}_1[$ the sequence bottom-up of inference rules lying between $\mathbf{r}_1$ and $\mathbf{r}_2$ in the branch of π connecting $\mathbf{r}_1$ and $\mathbf{r}_2$. The distance between $\mathbf{r}_1$, $\mathbf{r}_2$ in π, noted by $d_\pi(\mathbf{r}_1, \mathbf{r}_2)$, is the length of $]\mathbf{r}_2, \mathbf{r}_1[$. The minimal distance between $\mathbf{r}_1$, $\mathbf{r}_2$, noted by $d_m(\mathbf{r}_1, \mathbf{r}_2)$, is defined by: $d_m(\mathbf{r}_1, \mathbf{r}_2) = inf_{\nu,[\nu]=[\pi]} d_\nu(\mathbf{r}_1, \mathbf{r}_2)$.*

Lemma 4. *Let π be an MLL proof and $\mathbf{r}_1$, $\mathbf{r}_2$ two inference rules in π introducing the formulas F_1 and F_2, such that $\mathbf{r}_1$ is above $\mathbf{r}_2$. We assume that $d_\pi(\mathbf{r}_1, \mathbf{r}_2) = d_m(\mathbf{r}_1, \mathbf{r}_2)$. If r is an inference rule introducing a formula F in π such that $\mathbf{r} \in]\mathbf{r}_2, \mathbf{r}_1[$, then $(F_1 \to F) \in O([\pi])$.*

We can finally prove the following (see [1] for a detailed proof):

Proposition 5. *Let R be a proof net. One has $O(R) \subseteq (D_{⅋,\otimes}(R) \cup SF(R))^\star$.*

5 Conclusion

Comparing correctness criteria. We have seen that Contractibility, Mogbil and Naurois's criterion and DepGraph are three faces of the same notion, dependency. More precisely, those three criteria can been understood as different concrete implementations of a *proto-criterion* related with dependency relation, along the different points of view developed in the introduction: we showed that (i) Contractibility gives actually a sequentialization of a proof-net from which arises dependency, (ii) MN is a criterion with efficiency purposes (working on the generalized dependency graph, it is not clear how to stay in NL since it requires to remember which premise of a ⅋ node has been visited, thus justifying the seemingly odd choice of a switching) while (iii) DepGraph emphasizes structural properties of logic by clearly separating conditions on ⅋ inferences from other inferences and by unveiling the meaning of its dependency graph which correponds (when considered together with the sub-formula relation) to the order of introduction of inferences common to all sequentializations of a given proof-net.

This last point actually evidences that, while they are completely parallel proof objects, proof-nets contain enough logical dependency to allow for the retrieval of inherently sequential information by computing the dependency relation which represent the true logical causality of sequential proofs.

Future works. The present work suggests two main directions for future works:

- The separation between positive and negative inferences which is the cornerstone of DepGraph criterion suggests connections with focusing. While proof-nets and focalized proofs are the results of diverging choices of proof-theoretical design (parallelism versus hypersequentiality), this suggests that they actually may be different aspects of the same phenomenon as already advocated in the study of multi-focusing [16].
- Another direction concerns the development and the validation of our comparative study of proof-nets. Indeed, the prototypical classification we suggested is mainly built on empirical considerations and we plan to investigate it more systematically in the future, in particular by considering connections with other criteria which seems to be related with the notion of dependency such as Di Giamberardino and Faggian's work on jumps [17], or Murawski and Ong's work on dominator's trees [14].

References

1. Bagnol, M., Doumane, A., Saurin, A.: long version of this submission. available at http://www.pps.univ-paris-diderot.fr/~saurin/Publi/DepGraphLong.pdf
2. Howard, W.A.: The formulae-as-type notion of construction, 1969. In Seldin, J.P., Hindley, R., eds.: To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism. Academic Press, New York (1980) 479–490
3. Girard, J.Y.: Linear logic. Theoretical Computer Science **50**(1) (1987) 1 – 101
4. Girard, J.Y.: Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur. Thèse de doctorat, Université Paris VII (1972)
5. Reynolds, J.C.: Towards a theory of type structure. In: Symposium on Programming. Volume 19 of Lecture Notes in Computer Science., Springer (1974) 408–423
6. Girard, J.Y.: The system F of variable types, fifteen years later. Theoretical Computer Science **45** (1986) 159–192
7. Girard, J.Y.: Proof-nets: the parallel syntax for proof-theory. In Ursini, A., Agliano, P., eds.: Logic and Algebra. Volume 180 of Lecture Notes In Pure and Applied Mathematics., New York, Marcel Dekker (1996) 97–124
8. Danos, V., Regnier, L.: The structure of multiplicatives. **28** (1989) 181–203
9. Curien, P.L.: Introduction to linear logic and ludics, part ii (2006)
10. Girard, J.Y.: Le Point Aveugle: Cours de logique. Tome 1, Vers la perfection; Tome 2, Vers l'imperfection. Hermann (2006)
11. Danos, V.: Une application de la logique linéaire à l'étude des processus de normalisation. Thèse de doctorat, Université Denis Diderot, Paris 7 (1990)
12. Guerrini, S.: Correctness of multiplicative proof nets is linear. In: LICS, IEEE Computer Society (1999) 454–463
13. Guerrini, S.: A linear algorithm for MLL proof net correctness and sequentialization. Theor. Comput. Sci. **412**(20) (2011) 1958–1978
14. Murawski, A.S., Ong, C.H.L.: Fast verification of mll proof nets via imll. ACM Trans. Comput. Logic **7**(3) (July 2006) 473–498
15. de Naurois, P.J., Mogbil, V.: Correctness of linear logic proof structures is NL-complete. Theor. Comput. Sci. **412**(20) (2011) 1941–1957
16. Chaudhuri, K., Miller, D., Saurin, A.: Canonical sequent proofs via multi-focusing. In: TCS 2008, IFIP. Vol. 273 of IFIP., Springer (2008) 383–396
17. Giamberardino, P.D., Faggian, C.: Jump from parallel to sequential proofs: Multiplicatives. In: CSL 2006, LNCS, Springer (2006) 319–333