Annals of Pure and Applied Logic 155 (2008) 173–182

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

Proof nets sequentialisation in multiplicative linear logic

Paolo Di Giamberardino [a,*], Claudia Faggian [b]

[a] *Dipartimento di Filosofia, Università Roma Tre – Institut de Mathématiques de Luminy, Italy*
[b] *Preuves Programmes et Systèmes, Paris 7, France*

ARTICLE INFO

Article history:
Received 30 September 2007
Received in revised form 9 April 2008
Accepted 9 April 2008
Available online 4 June 2008
Communicated by J.-Y. Girard

MSC:
03F52
03B47

Keywords:
Linear logic
Proof nets
Sequentialization

ABSTRACT

We provide an alternative proof of the sequentialisation theorem for proof nets of multiplicative linear logic. Namely, we show how a proof net can be transformed into a sequent calculus proof simply by properly adding to it some special edges, called *sequential edges*, which express the sequentiality constraints given by sequent calculus.

0. Introduction

Proof nets are a graph-like syntax introduced by Girard in [7] as an abstract representation of linear logic proofs. This representation has two main features: to provide a tool for studying normalization, and to give a canonical representation of proofs.

In a proof net, the rules of sequent calculus are represented by nodes, and the information about the order on which the rules are performed is reduced essentially to the one corresponding to subformula trees and to the one providing the axiom links.

The key result in the theory of proof nets is the *sequentialisation theorem*, which states that it is possible to recover a sequent calculus derivation from a proof net. The standard proof of sequentialisation is built on the property that – at each step – we can find a node (called *splitting*) which corresponds to the last rule of a sequent calculus derivation; such a derivation – which is a tree of rule occurrences – is reconstructed by iterated application of this property. To prove this property is the most delicate part of proving the sequentialisation theorem. In [7], Girard introduces the notion of *empire* to prove the existence of a *splitting Tensor* , while in [5] Danos derives the existence of a *splitting Par* from the *section theorem* in graph theory.

In this paper, we provide a proof of sequentialisation for multiplicative linear logic following another approach: from a purely graph theoretical point of view, we consider the sequentialisation procedure as a way to turn a directed acyclic graph into a tree, respecting the constraints given by sequent calculus (a similar approach is also used by Banach in [1]) We prove that this transformation can be performed simply by properly adding to a proof net edges which express the sequentiality constraints of sequent calculus: we call such edges *sequential edges*.

As a tool for sequentialisation we introduce *constrainted nets* (or C-*nets*). C-nets are multiplicative proof nets, enriched with sequential edges; we prove that by gradual insertion of edges in a C-net, one can move gradually from C-nets of minimal

* Corresponding author.
E-mail address: digiambe@uniroma3.it (P. Di Giamberardino).

0168-0072/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.apal.2008.04.002

sequentiality (i.e. without sequential edges) to C-nets of maximal sequentiality (i.e., no more sequential edges can be added). The former are proof nets in the usual sense, the latter directly correspond to sequent calculus derivations: in this way we obtain a very simple proof of the sequentialisation theorem.

Our main technical result is the *Arborisation Lemma*, which provides the way to add sequential edges to a proof net; we also show that both standard splitting lemmas are direct consequences of the Arborisation lemma.

The idea of using edges to represent sequentiality constraints underlies the notion of *jump*, introduced by Girard in [9] and [8] as a part of correctness criteria for proof nets. Girard then also suggested that it could be possible to retrieve a sequent calculus proof from a proof net simply by adding jumps. In previous work [6], following this suggestion and in the spirit of [4,3], we introduced a new class of multiplicative focusing proof nets, *J-proof-nets*, where object with different degree of parallelism live together, and where sequentiality could be graduated by adding or removing jumps.

Building on that work, here we generalize the proof of sequentialisation given in [6] to standard multiplicative proof nets by using sequential edges (which can be considered as a generalization of jumps).

The paper is divided into the following sections:

- In Section 1, after introducing some terminology about directed acyclic graphs we give some background on the syntax of multiplicative linear logic (MLL) and proof nets. We revise the notion of proof nets, in order to be able to add sequential edges.
- In Section 2, we give an example of our approach to sequentialisation.
- In Section 3, we define constrainted nets (C-nets) as proof nets enriched with sequential edges, we introduce the key notion of *skeleton* of a C-net, and show that if the skeleton is tree-like, then it corresponds to a sequent calculus derivation (which is actually a tree of rule occurrences).
- Section 4 is dedicated to the proof of the arborisation lemma.
- In Section 5, we give direct proofs of standard results, such as the existence of a splitting Tensor.
- In Section 6 we restrict our attention to the class of C-nets whose sequential edges correspond to standard jumps. This allows us to study the relationship between our approach and the notion of *empire*.

1. MLL: Sequent calculus and proof nets

1.1. MLL

Let $\mathcal{V} = \{X,\ Y,\ Z, \ldots\}$ be a numerable set of propositional variables; a formula F of MLL is defined as follows:

$F ::= X | X^{\perp} | F \parr F | F \otimes F.$

Linear negation is defined by:

$$X^{\perp\perp} = X$$
$$(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp} \qquad (A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp}.$$

The sequent calculus derivation rules of MLL are:

$$\frac{}{\vdash A,\ A^{\perp}}(Ax) \qquad \frac{\vdash \Gamma,\ A \quad \vdash \Delta,\ A^{\perp}}{\vdash \Gamma,\ \Delta}(Cut)$$
$$\frac{\vdash \Gamma,\ A \quad \vdash \Delta,\ B}{\vdash \Gamma,\ A \otimes B,\ \Delta}(\otimes) \qquad \frac{\vdash \Gamma,\ A,\ B}{\vdash \Gamma,\ A \parr B}(\parr).$$

Finally, we will also consider the *Mix rule*

$$\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}(Mix).$$

1.2. D.a.g.'s, paths, ports

Preliminaries on d.a.g.'s. Given a directed acyclic graph (d.a.g.) and an oriented edge l from a node x to a node y, we denote l by $x \to y$, and we say that $x \to y$ *exits* from x and *enters* into y; y is called the *target* of l and x is called the *source*.

When drawing a d.a.g we represent edges oriented up-down; we will say that a node is above or below another node.

We call predecessor of a node c, a node which *immediately* precedes c.

A *root* of a d.a.g. is a node with no predecessors.

Partial order associated to a d.a.g. It is standard to represent a *strict partial order* as a d.a.g., where we have an edge $b \to a$ whenever $a <_1 b$ (i.e. $a < b$, and there is no c such that $a < c$ and $c < b$.) Conversely (the transitive closure of) a d.a.g. G induces a strict partial order $\prec_G$ on the nodes of G.

Skeleton of a d.a.g. Given a d.a.g. G, an edge $a \to b$ is said *transitive* if there is a node c and a sequence of nodes $b_1, \ldots, b_n$ in R such that $a \to b_1 \to b_2 \to \cdots \to b_n \to c$ and $c \to b$.

The *skeleton* of a d.a.g. G (denoted $Sk(G)$) is the d.a.g. obtained from G by erasing all the edges which are transitive.

Paths and cycles. Given a directed graph G, a *path* p which *starts* from a node $b = a_1$ and *reaches* a node $c = a_n$ is a sequence $\langle a_1, \ldots, a_n\rangle$ of nodes such that for each a_i, a_{i+1}, there is an edge l from a_i to a_{i+1}, or from a_{i+1} and a_i; we say that the edge l is *used* by p. A *cycle* is a path $\langle a_1, \ldots, a_n\rangle$ such that $a_1 = a_n$.

Ports. Given a d.a.g., we will partition the set of the edges which enter a node b into subsets which we call *ports*; we will denote the ports of b by b^x, b^y, $\ldots$, b^z.

We say that an edge l from a node a to a node b *enters into b trough the port* b^x, written $a \to b^x$, if l belongs to b^x.

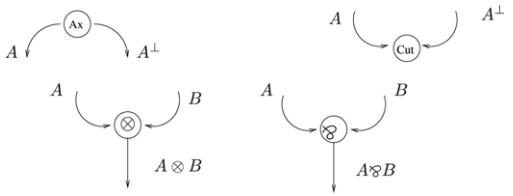
1.3. Proof structures and proof nets

We first give the standard definition of proof structure and proof net, where each node is labelled by a MLL rule; we then enrich the typing of the nodes with a partition of the entering edges into ports.

Proof structures. Proof structures are directed acyclic graphs with pending edges (that is some edges have a source but no target) whose nodes (also called *links*) are labelled by one of the symbols *ax*, *cut*, $\parr$, $\otimes$ (corresponding to MLL sequent calculus rules). The edges are typed by formulas of linear logic.

The label of a link imposes some constraints on both the number and the types of its entering edges (called *premises*) and exiting edges (called *conclusions*):

- the *ax*-link has two conclusions labeled by dual atomic formulas, but no premises;
- the *cut*-link has two premises labeled by dual formulas but no conclusions;
- the $\parr$-link has two ordered premises and one conclusion. If the left premise is labeled by the formula A and the right premise is labeled by the formula B, then the conclusion is labeled by the formula $A \parr B$;
- the $\otimes$-link has two ordered premises and one conclusion. If the left premise is labeled by the formula A and the right premise is labeled by the formula B, then the conclusion is labeled by the formula $A \otimes B$.



Each edge is the conclusion of a unique link and the premise of at most one link. Edges which are not the premise of any link are the *conclusions* of the proof structure.

Given a sequent calculus proof π of MLL (or MLL + Mix), we can associate to it a proof structure π^*, by induction on the height h of π, as follows.

If $h = 1$, then the last rule of π is an axiom with conclusions $X, X^\perp$; π^* is an axiom link with conclusions $X, X^\perp$. Otherwise:

- If the last rule r of π is a $\parr$-rule, having as premise the subproof π', then π^* is obtained by adding to π'^* the link corresponding to r.
- If the last rule of r is a $\otimes$- or a *cut* rule with premises the subproofs π_1 and π_2, then π^* is obtained by connecting π_1^* and π_2^* by means of the link corresponding to r.
- If the last rule of r is a Mix rule with premises the subproofs π_1 and π_2, then π^* is obtained by taking the union of π_1^* and π_2^*.

Definition 1 (*Sequentialisable*). A proof structure R is sequentializable if there exists a proof π such that $\pi^* = R$.

Proof nets. Let R be a proof structure; a *switching path* of R is a path which does not use any two edges entering on the same $\parr$ link (such edges are called *switching edges*); a *switching cycle* is a switching path which is a cycle.

Definition 2 (*Proof Net*). A *proof net* is a proof structure which does not contain switching cycles.

The following theorem states that the purely geometrical condition of being a proof net characterizes exactly all the proof structures with a logical meaning, that is proof structures which come from a sequent calculus proof.

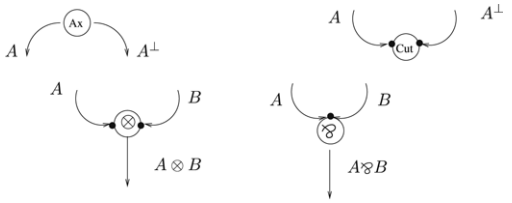
Theorem 1 (*Sequentialisation*). *A proof structure R is sequentializable if and only if is a proof net.*

The right to left direction is trivial; the aim of this paper is to provide a simple proof of the left to right one. To this purpose, we enrich the definition of proof structure by associating ports to each link.

Definition 3 (*Ports Associated to A Link*). To each link of a proof structure R, we associate a partition of *all* entering edges into ports, in the following way:

- a $\parr$ link of conclusion $A \parr B$ and premises A, B has *only one port*, containing all its premises;
- a $\otimes$ (resp. *cut*) link of conclusion $A \otimes B$ and premises A, B (resp. A, $A^\perp$), has *two ports,* one containing the premise typed by A, the other containing the premise typed by B (resp. $A^\perp$) .

From now on, we represent MLL links in the following way (where each black spot identifies a port):

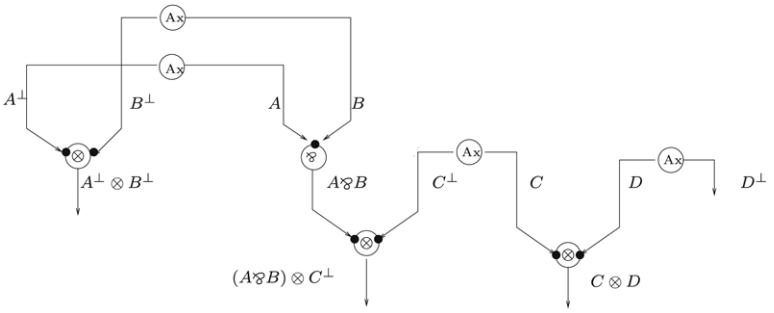


The definition of switching path can now be reformulated as follows:

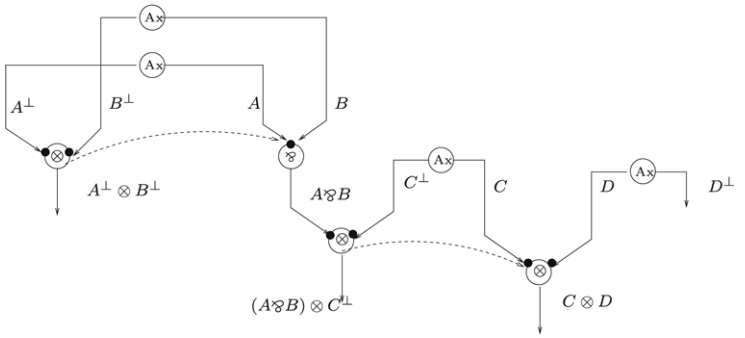
Definition 4 (*Switching Path*). Let *R* be a proof structure ; a *switching path* of *R* is a path which does not use any two edges entering trough the same port of a link.

2. Proof net, order and sequent calculus

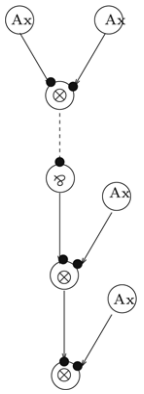
In this section, we give an intuition of our approach to sequentialisation. Consider the proof net below:



Now we add one untyped edge between the ⅋ link and the leftmost ⊗ link and one untyped edge between the middle ⊗ and the rightmost one:



Now we consider the partial order induced by such a directed graph, which yields the following tree:



Such a tree directly corresponds to the following sequent calculus proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^{\perp}}\,ax \quad \dfrac{}{\vdash B, B^{\perp}}\,ax}{\vdash A^{\perp}\otimes B^{\perp}, A, B}\,\otimes}{\vdash A^{\perp}\otimes B^{\perp}, A\parr B}\,\parr \quad \dfrac{}{\vdash C, C^{\perp}}\,ax}{\vdash A^{\perp}\otimes B^{\perp}, (A\parr B)\otimes C^{\perp}, C}\,\otimes \quad \dfrac{}{\vdash D, D^{\perp}}\,ax}{\vdash A^{\perp}\otimes B^{\perp}, (A\parr B)\otimes C^{\perp}, C\otimes D, D^{\perp},}\,\otimes$$

We will sequentialize a proof net by adding to it enough untyped edges (that we call *sequential edges*) to retrieve a sequent calculus proof.

In order to do that, we must extend our definition of proof structure to take into account proof structure with sequential edges; we will call such proof structures *constrainted structures*.

As we pointed out above, sequential edges can be considered as a generalization of jumps. We introduce ports for the links in order to make *all* the edges of a proof structure switching edges.

Remark 1. In this paper we deal only with cut-free proof net, since for sequentialisation a cut link of premises $A, A^{\perp}$ has the same geometrical properties of a $\otimes$ link with the same premises and conclusion $A \otimes A^{\perp}$.

3. Constrainted nets

Definition 5 (*Constrainted Structure*). A *constrainted structure* (or *C-structure*) is a d.a.g. obtained from a proof structure R (whose links have been given ports as in Definition 3), by adding untyped edges, called *sequential edges*, in such a way that each node n has the same label as in R, and each sequential edge entering n is added to one of the ports of n.

Definition 6 (*Constrainted Nets*). A C-structure R is called a *constrainted-net*, or *C-net*, if it has no switching cycles (in the sense of Definition 4).

Definition 7 (*Sub-nets*). Given a C-structure R, a sub-structure is any sub graph S of R which is a C-structure and such that for any link n if $n \in S$ then all edges entering on n belong to S; a sub-net of R is a sub-structure which is a C-net.

A proof net is a special case of C-net (without sequential edges). Below we show that a sequent calculus proof of MLL (or MLL + Mix) can also be seen as a special case of C-net (where there are enough sequential edges to recover the tree-like structure of the proof).

Skeleton of a C-net. We adapt the standard definition of *skeleton* of a d.a.g. to C-nets. The *skeleton* of a C-net R (denoted $Sk(R)$) is the d.a.g. (whose nodes have the same label as in R) obtained from R by erasing all the edges which are transitive (as defined in Section 1.2).

Remark 2. In the skeleton $Sk(R)$ of a C-net R, there is at least one edge for each port of a node m. If it was not the case, given two ports m^x, m^y of m there would be in R a node n, an edge $n \to m^y$ and a sequence of nodes $b_1, \ldots, b_n$ such that $a \to b_1 \to b_2 \to \ldots \to b_n \to m^x$; but then there would be a switching cycle in R.

It is immediately obvious that if the skeleton of a C-net is a forest,it (uniquely) corresponds to a sequent calculus derivation (which has the same tree structure.)

Proposition 1. *Let R be a C-net of conclusions $A_1, \ldots, A_n$ and such that $Sk(R)$ is a forest.*
We can associate to R a unique sequent calculus proof π^R of conclusion $\vdash A_1, \ldots, A_n$ in MLL + Mix.
Moreover, if $Sk(R)$ is a tree where each port has exactly one entering edge, π^R is a sequent calculus proof in MLL (without Mix).

Proof. The proof is by induction on the number ρ of nodes in $Sk(R)$.

1. $\rho = 1$. The only node in R is an Axiom link with conclusions X, $X^{\perp}$, to which we associate $\dfrac{}{\vdash X, X^{\perp}}$.
2. $\rho > 1$ and $Sk(R)$ has a single root. We reason by cases, checking the type of the root m of $Sk(R)$. The type can be:
   - $m = A\parr B$. Let $Sk(R)'$ be the forest obtained by erasing the root; to this forest corresponds a subnet R' of R with conclusion Γ, A, B. By induction we associate a proof $\pi^{R'}$ of conclusion Γ, A, B to $Sk(R)'$. π^R is $\dfrac{\pi^{R'}}{\vdash \Gamma, A\parr B}$, whose last rule is a $\parr$ rule on A, B ;
   - $m = A \otimes B$. By erasing m we obtain two forests $Sk(R)_1$, $Sk(R)_2$, one for each port of m. To each forest correspond two different subnets R_1, R_2 of R (because $Sk(R)$ is obtained just by erasing transitive edges) of conclusion Γ_1, A (resp Γ_2, B); by induction we associate a proof π^{R_i} to each $Sk(R)_i$.
     π^R is $\dfrac{\pi^{R_1}}{\vdash \Gamma_1, A} \quad \dfrac{\pi^{R_2}}{\vdash \Gamma_2, B}$ whose last rule is a $\otimes$ rule on A, B.
3. $\rho > 1$ and $Sk(R)$ has more than one root. We apply to each tree the induction hypothesis, obtaining n proofs , which we compose by using a mix rule. □

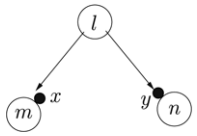


Fig. 1.

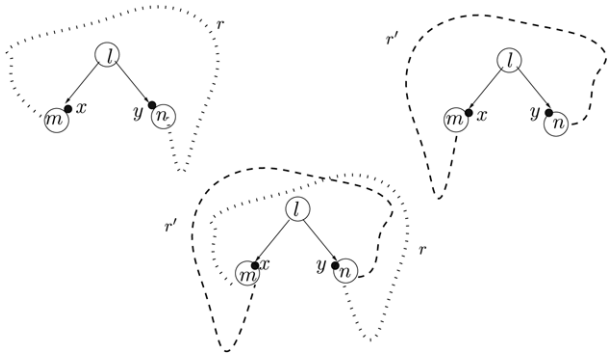


Fig. 2.

Given a C-net R, since it is a d.a.g., we can associate to it in the standard way a strict partial order $\prec_R$ on its nodes. Moreover, since $Sk(R)$ is obtained from R just by erasing transitive edges, the order associated to $Sk(R)$ and the order associated to R as d.a.g are equal.

We recall that a strict order r on a set A is *arborescent* when each element has a unique predecessor. If the order $\prec_R$ is arborescent, the skeleton of R is a forest, which (by Proposition 1) corresponds to a sequent calculus derivation π^R.

4. Arborisation

We say that a C-net is saturated when there is no way to add sequential edges to increase the order, and still have a C-net: any additional edge either does not increase the order, or violates the correctness criterion.

Definition 8 (*Saturated C-net*). A C-net R is *saturated* if for each pair of links m and n, adding a sequential edge between m and n either creates a switching cycle or doesn't increase the order $\prec_R$.

Given a proof net R, a saturation $Sat(R)$ of R is a saturated C-net obtained from R by adding sequential edges.

Our sequentialisation argument is as follows:

- Any C-net can be saturated.
- The order associated to a saturated C-net is arborescent.
- If the order $\prec_R$ associated to a C-net R is arborescent, then $Sk(R)$ is a forest and we can associate to R a proof π^R in the sequent calculus.

The following lemma is the key result of this paper.

Lemma 1 (*Arborisation*). *Let R be a C-net. If R is saturated then $\prec_R$ is arborescent.*

Proof. We prove that if $\prec_R$ is not arborescent, then there exist two links m and n s.t. adding a sequential edge between m and n (or viceversa) doesn't create switching cycles and makes the order increase.

If $\prec_R$ is not arborescent, then in R there exists a link l with two immediate predecessors m and n (they are incomparable). Observe that m and n are immediately below l so that there is an edge $l \to m^x$ entering m trough a port m^x (resp. an edge $l \to n^y$ entering n trough a port n^y) in R (see Fig. 1).

Now suppose that both the addition of a sequential edge from m to n^y and the addition of a sequential edge from n to m^x create cycles.

(i) first we observe that there is no path in R from n to m which does not exit from n trough n^y and does not enter into m trough the port m^x, by correctness of R;

(ii) there is a switching path r from n to m which does not exit from n trough the port n^y. Similarly there is a switching path r' from m to n, which does not exit from m trough the port m^y (see Fig. 2).

(iii) Following the path r, we take the first node c different from n shared by r, r'; we observe that such a node exists, since in the worst case it is m.

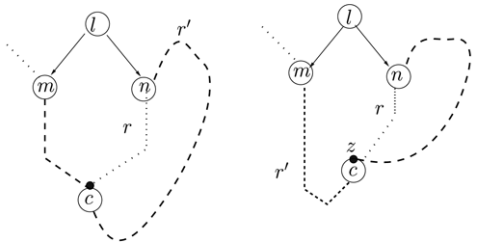


Fig. 3.

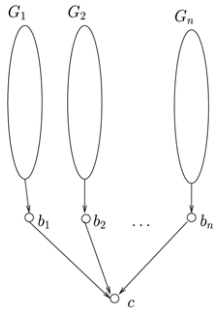


Fig. 4. The node c is splitting.

Now we have the following cases (see Fig. 3):

- either we can compose the subpath of r from n to c with the subpath of r' from c to n^y, and we get a switching cycle, contradiction;
- otherwise, r enters into c trough the port c^z of c and r' leaves c trough the same port c^z. Then we compose the subpath of r from n to c^z with the reversed subpath of r' from c to m, providing a path from n to m which does not exit n fron n^y and does not enter into m trough m^x, contradicting (i). □

5. Properties

In this section we deal with three standard results one usually has on proof nets. The novelty here is the argument. When adding sequential edges to a C-net R, we gradually transform the skeleton of R into a forest. We observe that *some properties are invariant under the transformations* we consider: adding sequential edges and removing transitive edges. Our argument is always reduced to simple observations on the final tree (the skeleton of $Sat(R)$), and on the fact that each elementary graph transformation preserves some properties of the nodes.

In 5.1 we give an immediate proof of a standard result, the splitting $\otimes$ lemma, and in 5.2 we prove that the sequentialization we have defined is correct w.r.t. Definition 1. In 5.3 we restrict our attention to MLL proof nets by getting rid of the Mix rule.

5.1. Splitting

A graph G is connected if for any pair of nodes a, b of G there exists a path (see 1.2) from a to b. We observe that given a d.a.g., adding edges, or deleting transitive edges, preserves connectedness. The following properties are all immediate consequences of this remark.

Lemma 2. (i) *Two nodes in a d.a.g. G are connected (i.e. there exists a path between the two nodes) iff they are connected in the skeleton of G.*

(ii) *Given a proof net R, if two nodes are connected in R, then they are connected in any C-net R^c obtained from R by adding sequential edges.*

(iii) *If R is connected as a graph so are R^c and $Sk(R^c)$.*

This lemma allows us to give a simple proof of a standard result, the Splitting Lemma, which we state below.

Definition 9 (*Splitting*). Let G be a d.a.g., c a root, and $b_1, \ldots, b_n$ the nodes which are immediately above c. We say that the root c is *splitting* for G if, when removing c, the nodes b_i are pairwise not connected; that is, by removing c, the graph splits into n connected components, each one containing at most one among the nodes $b_1, \ldots, b_n$ as in Fig. 4.

Remark 3. It immediately follows that if R is a C-net and a root c is splitting, the removal of c splits R into n disjoint connected components $R_1, \ldots, R_n$, and each component is a C-net.

Lemma 3 (*Splitting* $\otimes$). *Let R be a C-net whose roots are all of type $\otimes$, and let $Sat(R)$ be a saturation (hence, its skeleton is a forest); each node which is a root in $Sk(Sat(R))$ is splitting for R.*

Proof. Observe that c is obviously splitting in the skeleton of $Sat(R)$, because c is the root of a tree. Hence it is splitting in $Sat(R)$, as a consequence of Lemma 2, (i). Similarly, c must be splitting in R, as a consequence of Lemma 2, (ii). □

5.2. Sequentialisation is correct

In this section we show that sequentialization (Proposition 1) is correct. This also provide a proof that any proof net is sequentializable.

Proposition 2. *Let R be a proof net. For each saturation $Sat(R)$ of R, if $\pi = \pi^{Sat(R)}$ then $(\pi)^* = R$.*

Proof. The proof is by induction on the number ρ of nodes of R.

1. $\rho = 1$: then R consists of a single Axiom link of conclusions $A, A^\perp$, and π is the corresponding Axiom rule $\frac{}{\vdash A,\ A^\perp}$.
2. $\rho > 1$ and $Sk(Sat(R))$ is a tree. We consider its root k. By Proposition 1, to k corresponds the last rule of π. Let $\pi_1, \ldots, \pi_j$ be its premises (there is a premise for each port of k). By removing k from $Sat(R)$ we obtain a set of subnets ${R^c}_1, \ldots, R^c_j$ (one for each port of k). It immediately follows that each R^c_i is saturated. By definition, each π_i is the proof associated to an R^c_i.
   - Assume k is a $\parr$ node of conclusion $A \parr B$. We remove k from R, obtaining R_1. It immediately follows that R^c_1 is a saturation of R_1. Both have conclusions Γ, A, B. By induction hypothesis $\pi^*_1 = R_1$. We then apply a $\parr$ rule of conclusion $\vdash \Gamma, A \parr B$ to the proof $\frac{\pi_1}{\vdash \Gamma, A, B}$, so to obtain a proof which is equal to π and such that $R = \pi^*$.
   - Assume k is a $\otimes$ node of conclusion $A \otimes B$. By the splitting lemma, k is splitting in R. Let R_1, R_2 be the two sub nets (one for each port of k) of conclusions respectively Γ_1, A and Γ_2, B, obtained by removing k from R. It immediately follows that R^c_1, R^c_2 (with conclusions respectively Γ_1, A and Γ_2, B) are a saturation of, resp. R_1, R_2. By induction hypothesis $\pi^*_1 = R_1$ (resp., $\pi^*_2 = R_2$). We apply a $\otimes$ rule of conclusion $\vdash \Gamma_1, \Gamma_2, A \otimes B$ to the proofs $\frac{\pi_1}{\vdash \Gamma_1, A}$ and $\frac{\pi_2}{\vdash \Gamma_2, B}$, to obtain a proof which is equal to π and such that that $R = \pi^*$.
3. Otherwise, $Sk(R)$ is a forest with more than one root. By Lemma 2 each tree corresponds to a connected component of R. We apply the above procedure to each tree, and conclude with a number of Mix rules. □

5.3. MLL proof nets (without Mix rule)

In order to characterize the proof nets which correspond to MLL proofs, i.e. to get rid of the Mix rule, we now deal with a notion of connectedness, namely s-connectedness, which is specific to the theory of proof nets.

Definition 10 (*Correction Graph*). Given a d.a.g. R where, for each node, all entering edges are partitioned into ports, we call *switching* a function s which associates to each port of R one of its entering edges (the selected edge may or may not be typed); a *correction graph* $s(R)$ is the graph obtained by erasing for each port of R the edges not chosen by s.

Definition 11 (*s-Connected*). A C-net R is s-connected if given a switching of R, its correction graph is connected.

Remark 4. We only need to check a single switching. The condition we have given is equivalent to the condition that *all* correction graphs are connected.

The Euler–Poincaré invariant shows that assuming that all correction graphs are acyclic, if for a switching s the correction graph $s(R)$ is connected, then for all other switching s' we have that $s'(R)$ is connected.

Proposition 3. *If a proof net R is s-connected, then $Sat(R)$ is also s-connected, and $Sk(Sat(R))$ is a tree where each port has exactly one entering edge.*

Proof. First we observe that:

- any switching of R is also a switching of $Sat(R)$, producing the same correction graph. Hence if R is s-connected, $Sat(R)$ is s-connected.
- Given a C-net G, any switching of its skeleton is also a switching of G, because the skeleton is obtained by erasing some edges (those which are transitive).

As a consequence, any switching of $Sk(Sat(R))$ induces a correction graph which is a correction graph also for $Sat(R)$ and hence is connected. Moreover, we observe that there is only one possible switching. In fact, since $Sk(Sat(R))$ is a tree, we cannot erase any edge and still obtain a graph which is connected.

We conclude that in $Sk(Sat(R))$ each port has exactly one entering edge. □

From Proposition 1, it follows that

Proposition 4. *If R is s-connected, and $Sat(R)$ a saturation, we can associate to it a proof $\pi^{Sat(R)}$ which does not use the Mix rule.*

6. Jumps and splitting

In this final section, we compare the main tool in our proof of sequentialisation, that is sequential edges, with the standard techniques usually employed to prove the same result.

To analyze such a relation, we will restrict our focus on a particular class of C-nets, called C^J*-nets*; in such C-nets, all switching edges are premises only of $\parr$-links, so that sequential edges correspond to standard jumps.

Empires are a class of subnets which has been introduced by Girard in [7] (and further studied by Bellin and Van De Wiele in [2]), to prove sequentialisation; in Section 6.1 we will study the relation between sequential edges and *empires*.

The splitting $\parr$ lemma is a result, alternative to the splitting $\otimes$ lemma, introduced by Vincent Danos in [5] to prove sequentialisation; in Section 6.2 we give an immediate proof of the splitting $\parr$ lemma using sequential edges.

Definition 12. Given a C-net R, a *jump* is a sequential edge from a link b to a link c of R, such that c is a $\parr$-link; in this case we say that c jumps on b. A C^J*-net* is a C-net such that every sequential edge is a jump.

Definition 13 (*Jump-saturation*). A C^J-net R is *jump-saturated* if given a $\parr$-link p for each link q of R, jumping p on q yields either a cycle either a transitive edge; a jump-saturation of R is a jump-saturated C^J-net obtained from R by adding jumps.

6.1. Jumps and geography of subnets

In this section, we will consider only s-connected C-nets.

Given a correction graph $s(R)$ of a C-net R, a path $r\ \langle m, \ldots, n\rangle$ which leaves a link m of conclusion a and reaches a link n is said to *go up* from m, when it does not use neither a neither any sequential edge which exits from m; otherwise r is said to *go down* from m.

In the following Definition 14, we revise the standard definition of empire, taking into account sequential edges.

Definition 14 (*Empire*). Let a be conclusion of a link n in a C^J-net R: the *empire* of a in R (denoted $emp_R(a)$) is the smallest set of links closed under the following conditions:

- n belong to $emp_R(a)$;
- if m is a link of R connected with n with a path that goes up from n in all correction graphs of R, then $m \in emp_R(a)$.

We call *border* of $emp_R(a)$ the set of links $m_1, \ldots, m_n$ such that $m_i \in emp_R(a)$ and the conclusion of m_i is either a conclusion of R or a premise of another link which does not belong to $emp_R(a)$.

Lemma 4. *Let R be a C^J-net and p a $\parr$-link with conclusion a: then the C-structure R' obtained from R by jumping p to another link q is a C-net iff $q \in emp_R(a)$.*

Proof. We first prove the right to left direction: if $q \in emp_R(a)$ this means that for every correction graph of R (which is a correction graph of R' too), there is a path going up from p to q; if in R' there were a cycle, this means that in some correction graph of R' there would be a path from p to q which doesn't use any switching edge of p, so it's also in a correction graph of R, but then we have a cycle in some correction graph of R, contradiction. To prove the other direction, we simply observe that if q doesn't belong to $emp_R(a)$ this means that by s-connectedness there is at least one correction graph in R such that there is a path starting down from p to q; but then jumping p on q we get a cycle. □

Now we will prove that is possible to characterize jump saturated C^J-net by the shape of the empires of their $\parr$-links:

Definition 15 (*Cone*). Given a C-net R and a link m of conclusion a, we call *cone* of a in R (denoted $C_R(a)$) the set of all the links which are hereditary above m.

Proposition 5. *A C^J-net R is jump-saturated, iff for any $\parr$ link p of conclusion a, $emp_R(a) = C_R(a)$.*

Proof. To prove the left to right direction, let us assume R jump-saturated, and suppose $emp_R(a) \neq C_R(a)$; obviously $C_R(a) \subset emp_R(a)$. Now consider a link b of $emp_R(a)$ which isn't in $C_R(a)$: if there isn't such an element, then $emp_R(a) = C_R(a)$; otherwise we make p jump on b , and by Lemma 4 this doesn't create cycles so R is not jump-saturated.

To prove the other direction, we observe that jumping p on a link which is in $emp_R(a)$ yields a transitive jump by definition of $C_R(a)$; jumping p on a link which is outside $emp_R(a)$ creates a cycle by Lemma 4. □

6.2. Jumps and $\parr$-splitting lemma

Definition 16 (*Splitting $\parr$-Link*). Given a C-net R and a $\parr$-link p, we say that p is *splitting* for R if there exist two subgraphs G_1, G_2 of R, such that G_1 does not contain the conclusion of p, which is contained in G_2, and the only edge of R connecting a node in G_1 with a node in G_2 is the conclusion of p.

Lemma 5. *Given a jump-saturated C^j-net R and two $\parr$ links b, c either $C_R(b) \cap C_R(c) = \emptyset$, either one among $C_R(b)$, $C_R(c)$ is strictly included into the other.*

Proof. Assume that $C_R(b) \cap C_R(c) \neq \emptyset$, $b \notin C_R(c)$ and $c \notin C_R(b)$; now consider a node $a \in C_R(b) \cap C_R(c)$.

Every node in $C_R(b) \cap C_R(c)$ is hereditary above both b and c, so there is a a directed path r' (resp. r'') from a to b (resp. from a to c)

Since R is jump-saturated , there is a switching path $\langle c, \ldots, b\rangle$ starting down from the conclusion of c to b (otherwise we could make c jump on b); now this path cannot intersect r'', (otherwise there would be a cycle), and if it meets a node d of r', it follows r' from d to b: we call this path p'. In the same way we can build a switching path p'' $\langle b, \ldots, c\rangle$ starting down from the conclusion of b to c.

The rest of the proof is the same as the proof of the arborisation lemma ; p' and p'' either do not meet, either they do; in any way, we get a cycle. □

Lemma 6. *Let R be a saturated C^j-net R and a a $\parr$-link of R. If a conclusion of a node $b \in C_R(a)$ is a premise of a link $c \notin C_R(a)$, then c is a $\parr$-link.*

Proof. Suppose c is a $\otimes$-link; then by saturation, jumping a on c would create a cycle: but then there is a switching path r from a to c which does not use any switching edge of a. Since b is hereditary above a and $b \to c$ it is straightforward that the existence of r would induce a switching cycle in R, contradiction. □

Lemma 7 (*Splitting $\parr$-Lemma*). *Given a proof net R with at least one $\parr$-link, there exists a splitting $\parr$-link.*

Proof. Starting from R, we retrieve a jump saturated C^j-net R' by adding jumps on R.

Now, if there is a splitting $\parr$-link p in R' by Lemma 2 the same link p will be splitting also for R; then by proving the existence of a splitting $\parr$-link for R', we provide also a splitting $\parr$-link for R.

Let us consider a $\parr$-link a such that $C_{R'}(a)$ is maximal with respect to inclusion among all the cones of the $\parr$-links in R'; we prove that a is splitting for R'.

Suppose that a conclusion of a link in $C_{R'}(a)$ is the premise of a link (different from a) which does not belong to $C_{R'}(a)$ so it must be a premise of another $\parr$-link b by Lemma 6; now, $C_{R'}(a) \cap C_{R'}(b) \neq \emptyset$, and $b \notin C_{R'}(a)$ so by Lemma 5 $C_{R'}(a) \subset C_{R'}(b)$, contradicting the maximality of $C_{R'}(a)$.

We observe also that if a $\parr$-link c different from a jumps on a link d which belongs to $C_{R'}(a)$, then $c \in C_{R'}(a)$; otherwise (that is if $c \notin C_{R'}(a)$), $C_{R'}(a) \cap C_{R'}(c) \neq \emptyset$ and $c \notin C_{R'}(a)$ and again by Lemma 5 $C_{R'}(a) \subset C_{R'}(c)$, contradicting the maximality of $C_{R'}(a)$.

So, each conclusion of a link in $C_{R'}(a)$ is a premise of a link in $C_{R'}(a)$, or a premise of a, or a conclusion of R'.

Now, if we consider the subgraph G of R' which corresponds to $C_{R'}(a)$, by the above observations, all the paths connecting a node in G with a node in $R' \setminus G$ must pass trough the conclusion of a; but then a is splitting for R'. □

7. Conclusions and future work

By introducing sequential edges, we provided a purely geometrical way to turn a proof net into a sequent calculus proof, and we provided alternative proofs for the standard splitting lemmas of MLL proof nets. As a future research direction we aim to extend this approach to sequentialisation to proof nets of Multiplicative Additive Linear Logic, as defined by Hughes and Van Glabbeek in [10].

Acknowledgements

The authors wish to thank J. Y. Girard, for having first suggested the idea of sequentialising with jumps. Thanks also to Lorenzo Tortora de Falco, Michele Pagani, Damiano Mazza and Olivier Laurent for useful and fruitful discussions.

References

[1] R. Banach, Sequent reconstruction in *MLL*- A sweepline proof, Annals of Pure and Applied Logic 73 (1995) 277–295.
[2] G. Bellin, J. Van De Wiele, Subnets of proof-nets in MLL, in: J.-Y. Girard, Y. Lafont, L. Regnier (Eds.), Advances in Linear Logic, 1995.
[3] P.-L. Curien, C. Faggian, An approach to innocent strategies as graphs (submitted for publication).
[4] P.-L. Curien, C. Faggian, L-nets, strategies and proof-nets, in: CSL 05 (Computer Science Logic), in: LNCS, Springer, 2005.
[5] V. Danos, La Logique Linéaire appliquée l'étude de divers processus de normalisation (principalement du λ-calcul). Ph.D. Thesis, Université Paris 7, 1990.
[6] P. Di Giamberardino, C. Faggian, Jump from parallel to sequential proof: Multiplicatives, in: Proc. of CSL'06 (Computer Science Logic), Lecture Notes in Computer Science, 2006.
[7] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1987) 1–102.
[8] J.-Y. Girard, Proof-nets: The parallel syntax for proof-theory, in: Ursini, Agliano (Eds.), Logic and Algebra, 1996.
[9] J.-Y. Girard, Quantifiers in linear logic II, in: Nuovi problemi della Logica e della Filosofia della scienza, 1991.
[10] D. Hughes, R. Van Glabbeek, Proof nets for unit-free multiplicative-additive linear logic, in: Proc. of LICS'03, IEEE, 2003.