Strong Normalization of Proof Nets Modulo Structural Congruences*

Roberto Di Cosmo[1] and Stefano Guerrini[2,**]

[1] DMI-LIENS (CNRS URA 1347) Ecole Normale Supérieure
45, Rue d'Ulm - 75230 Paris France. Email:dicosmo@ens.fr
[2] Dipartimento di Scienze dell'Informazione - Università di Roma I, 'La Sapienza'
Via Salaria, 113 - I-00198, Roma, Italy - Email: guerrini@dsi.uniroma1.it

Abstract. This paper proposes a notion of reduction for the *proof nets* of Linear Logic modulo an equivalence relation on the *contraction links*, that essentially amounts to consider the contraction as an associative commutative binary operator that can float freely in and out of proof net *boxes*. The need for such a system comes, on one side, from the desire to make proof nets an even more parallel syntax for Linear Logic, and on the other side from the application of proof nets to λ-calculus with or without explicit substitutions, which needs a notion of reduction more flexible than those present in the literature. The main result of the paper is that this relaxed notion of rewriting is still strongly normalizing.

Keywords: Proof Nets. Linear Logic. Strong Normalization.

1 Introduction

In his seminal paper [6], Girard proposed proof nets as a *parallel syntax* for Linear Logic, where uninteresting permutations in the order of application of logical rules are de-sequentialised and collapsed. Nevertheless, in the presence of exponentials, that are necessary to translate λ-terms into proof nets, the traditional presentation of proof nets turns out to be inadequate: too many inessential details concerning the order of application of independent structural rules (*e.g.*, contraction) are still present.

When using proof nets to simulate λ-calculus, this redundancy already gets in the way, so that it is necessary to consider an extended notion of reduction, or a special version of proof nets with an *n*ary structural link and a brute force normalization procedure. But if one tries to simulate the behavior of explicit substitutions, then one is really forced to consider contraction links as a sort of associative-commutative operator.

Looking carefully at these difficulties, one can see that what is really needed is an extension of the notion of reduction on proof nets where the order of application of the contraction rules, and the relative order of contraction rules and box formation rules is

* Partially supported by: Italian MURST project *Tecniche Formali per la Specifica, l'Analisi, la Verifica, la Sintesi e la Trasformazione di Sistemi Software*; UE TMR Network *LINEAR: Linear Logic in Computer Science*.

** Written while at the Department of Computer Science of Queen Mary and Westfield College, London, UK supported by an EPSRC grant.

abstracted away. This can be done by defining an equivalence relation over regular proof nets that essentially amounts to consider the contraction as an associative-commutative binary operator that can float freely in and out of proof net *boxes*, and define a notion of reduction on the corresponding equivalence classes. Both λ-calculus and systems of explicit substitution can be very easily simulated in such a system. Also, this system allows to abstract away all the uninteresting permutations in the order of application of *structural* rules, which are de-sequentialised and collapsed into the same equivalence class. Yet, up to now, it was unknown whether such an extension would enjoy the same good properties as proof nets, and first of all, strong normalization. The main result of the paper is that this relaxed notion of rewriting is still strongly normalizing.

In the following, we shall first recall the traditional definition of proof nets and of their reduction, as well as the systems proposed by Danos and Regnier [4] to simulate λ-calculus, and by Di Cosmo and Kesner [5] to simulate a calculus with explicit substitution. Then, we shall define our equivalence relation and prove our main theorem.

1.1 Linear Logic and Proof Nets

Let us recall some classical notions from Linear Logic. We shall consider Multiplicative Exponential Linear Logic (MELL) without constants, *i.e.*, the fragment of Linear Logic whose formulas are: $\mathcal{F} ::= a \mid \mathcal{F} \otimes \mathcal{F} \mid \mathcal{F} \invamp \mathcal{F} \mid\, !\mathcal{F} \mid\, ?\mathcal{F}$, where a ranges over a non-empty set of *atoms* $\mathcal{A}$ that is the sum of two disjoint subsets $\mathcal{P}$ and $\mathcal{P}^\perp$, corresponding to the *positive* atoms p and to the *negative* atoms $p^\perp$ respectively. In particular, $p^\perp$ is named the *linear negation* of p, and vice versa. Linear negation extends to every formula A by means of the following De Morgan equations: $(A \otimes B)^\perp = A^\perp \invamp B^\perp$, $(?A)^\perp = \,!A^\perp$, $A^{\perp\perp} = A$. The connectives $\otimes$ (tensor) and $\invamp$ (par) are the *multiplicatives*; the connectives ! (of-course) and ? (why-not) are the *exponentials*. For the definition of the sequent calculus of Linear Logic, we refer the reader to [6].

One of the advantages of MELL is the availability of a graph-like representation of proofs that is highly non-sequential, that is, which is often able to forget the order in which some rules are used in a sequent calculus derivation, when this order is irrelevant. This representation is known as Proof Nets.

A (MELL) proof net is a finite (hyper)graph whose vertices are occurrences of MELL formulas (in the following, we shall often write ‘formula’ for ‘occurrence of formula’) and whose (hyper)edges, named *links*, correspond to connections between the active formulas of some rule of the sequent calculus of MELL. The formulas below a link are the *conclusions* of the link; the formulas above a link are its *premises*.

Fig. 1 gives the inductive rules for the construction of proof nets. As usual Γ, $?\Gamma$ and Δ stand for sets of formulas—in this case, sets of conclusions of the net above them—in particular, $?\Gamma$ denotes a set of ?-formulas. The rule *axiom* is the base case: a proof net formed of a unique link of type ax. The rules *par*, *contraction*, *dereliction* and *weakening* add a new link of the corresponding type to a previously constructed proof net. The rules *tensor* and *cut* add a new link and merge two (distinct) proof nets. Finally, the *promotion* rule promotes a formula A to $!A$. In order to apply that rule, we need a proof net M whose conclusions but A are of type ?. As a result, promotion encloses M into a *box* whose conclusions are the promoted formula $!A$ and a copy of each ?-

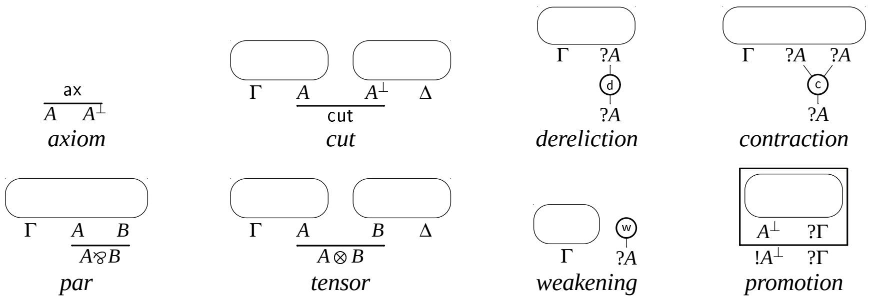

Fig. 1. Proof Nets.

conclusion of M. The conclusion $!A$ is the *principal port* of the box; the conclusions in $?\Gamma$ are its *auxiliary ports*.

Boxes force a strong constraint on the *sequentialization* of a proof net (*i.e.*, on the construction of a proof net by application of rules in Fig. 1): in any possible sequentialization of a proof net that contains a box B, no rule corresponding to a link below a conclusion of B can be applied before the complete sequentialization of B. However, the notion of box is crucial for the definition of proof net cut-elimination. In fact, because of the side condition on promotion (recall that all the auxiliary premises of a box must be of type ?), we have to keep track of the context that allowed the promotion of A (again, for a more detailed analysis, refer to [6]).

Remark 1. A proof net M is a (hyper)graph, so it does not contain any explicit information on the ways in which it can be sequentialized (*e.g.*, think at the strings of some context free language; the strings do not contain any information on their derivations in the context free grammar of the language). Therefore, let us assume to have a (hyper)graph M formed of formulas and links—such (hyper)graphs are known as *proof structures*. The problem 'is the proof structure M a proof net?' is clearly decidable, *e.g.*, take the brute force approach that tries ordering links in all the possible ways. The so called *correctness criteria* characterize proof nets with no explicit reference to the rules in Fig. 1. For instance, the Danos-Regnier criterion states that M is a proof net when all the *switches* of M are trees (a switch is a graph obtained by collapsing some boxes and by removing some edges). For a detailed discussion of correctness criteria and of their complexity, see [3, 7].

The rewriting rules in Fig. 2 define the cut-elimination procedure for proof nets. In fact, each cut-elimination rule in Fig. 2 transforms a proof net into a proof net (see [6]). In Fig. 2, a link between instances of the same set of formulas means that there is a link between each pair/triple of corresponding formulas in that sets.

Definition 1 (PN). *Proof Nets is the smallest set of (hyper)graphs closed by the rules in Fig. 1.* PN *is the rewriting system defined on Proof Nets by the rules in Fig. 2.*

In the following, $M \in$ PN will denote that M is a proof net. Moreover, since we shall consider several variants of proof net reduction, this will also mean that M reduces according to the rules of PN.

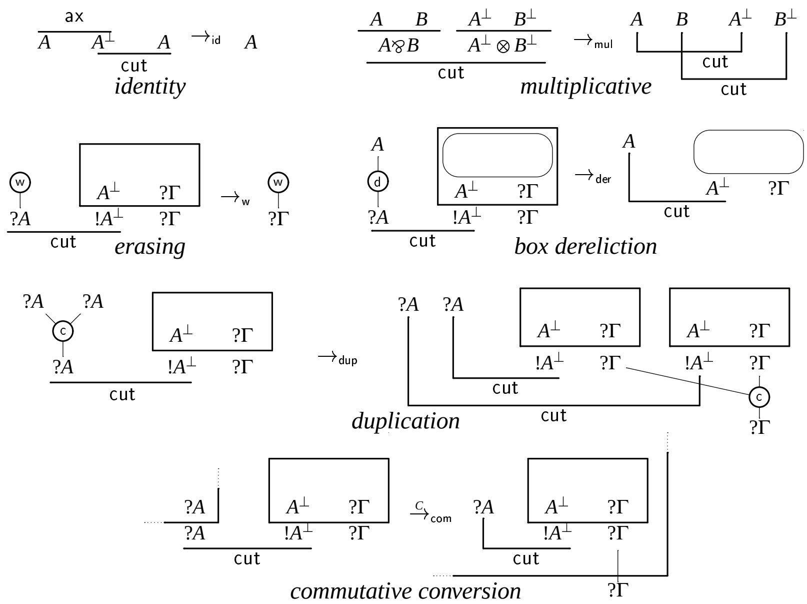

Fig. 2. Proof net cut-elimination.

Theorem 1. PN *is strongly normalizing and confluent (Church-Rosser). As a consequence,* PN *has the unique normal form property.*

Strong normalization (SN) was proved by Girard in [6] (Girard's proof of SN uses the *candidats de réductibilité*; a completely syntactical proof of SN can be found in Joinet's thesis [8]); the Church-Rosser property (CR) was proved by Danos in [2].

Henceforth, let us write $\mathsf{nf}(N)$, for the normal form of $N \in \mathsf{PN}$. More generally, since all the reduction systems that we shall analyze will be derived from PN and will be named by sub/superscripted variants of PN, $N \in \mathsf{PN}_x^y$ will denote that N reduces according to the rules of PN_x^y and $\mathsf{nf}_x^y(N)$ will denote its normal form (if any).

2 Survey and our proposal

2.1 Simulating the λ-calculus: collapsed structural links

When simulating the β-reduction of λ-calculus in PN, the rigidity of the exponential links makes things difficult: the net translation of a term t does not always reduce exactly to the translation of the reduct term s, due to the different shape of the contraction trees in the translation. This is quite annoying, to the point that the first really satisfactory proof of simulation can be found in [4], where Danos and Regnier introduce a system where all exponential links are collapsed into one single *n*ary link.

Usual proof nets are mapped into those proposed in [4] by a transformation μ that pushes contraction and dereliction out of all boxes and contracts them together. Fig. 3

describes μ by applying it to an example; see the mapping on the left. The root of the exponential tree in the example is not the premise of a contraction and is not above the auxiliary port of a box. The collapsed link of type ? that replaces the tree preserves the branches of the tree and the number of boxes that they cross. Every weakening link is replaced by a new link of type $\times$ that introduces a special (crossed) occurrence $A^\times$ of the formula A. Every formula $A^\times$ marks a *weakening branch* of the ?-link. A ?-*weakening tree* is a ?-link connected to weakening branches only; it is the translation of an exponential tree whose leaves are all weakening links. A ?-*weakening* is a ?-weakening tree formed of one weakening branch only; it corresponds to the translation of an exponential tree formed of a weakening link only (*e.g.*, see the mapping on the right in Fig. 3). The introduction of the weakening branches is due to technical reasons; the rationale is that we want to keep track of all the erasing rules required by the reduction. The $\times$-link is not present in [4], where weakening branches are simply erased.

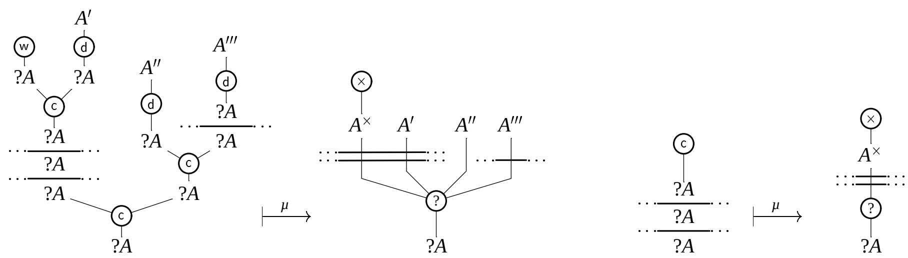

Fig. 3. Collapsing an exponential tree into a ?-link.

Definition 2 **(PN_C).** *Let* PN_C *be the set of the proof nets where contractions and exponential crossings at the auxiliary doors of boxes collapse into a unique nary link of type* ?*, and all the exponential reductions but erasing are collapsed into a unique* exponential *reduction step that performs unboxing, duplication and box inclusion, as shown by the example in Fig. 4.*

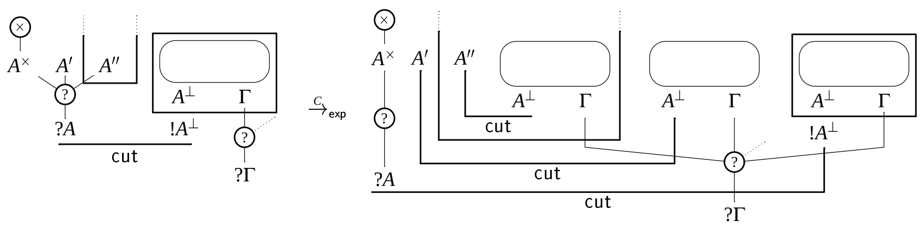

Fig. 4. The exponential rule of PN_C.

The exponential rule of PN_C introduces a ?-weakening cut for every weakening branch of the ?-link in the redex. In order to erase the corresponding boxes, that cuts must be explicitly eliminated by means of an erasing rule. The erasing rule of PN_C is the obvious translation of the erasing rule of PN: on the left-hand side, replace the weakening link by a ?-weakening and the auxiliary port crossings by ?-link branches; on the right-hand side, transform each branch into a weakening branch by putting a $\times$-link above its leaf. When the ?-link in the redex is a ?-weakening tree with n branches, the exponential rule degenerates into a *weakening duplication* that creates n copies of the box in the redex and splits the cut into n ?-weakening cuts. In particular, when the tree is a ?-weakening (*i.e.*, $n = 1$), the left-hand side and right-hand side would coincide; therefore, in order to not introduce trivial reduction loops, the exponential rule does not apply to a ?-weakening cut; the only rule that applies to that cuts is erasing. In [4], the absence of weakening branches corresponds to an exponential rule in which the ?-weakening cuts introduced by our version of the rule are automatically eliminated.

Remark 2 (No exponential axioms). The transformation μ is not defined for the proof nets that contain exponential axioms (*i.e.*, $!A, ?A^\perp$ axioms). From the point of view of provability, this is not a problem, for it is well-known that each proof net can be η-expanded into another one with the same conclusions that contains atomic axioms only (*i.e.*, $p, p^\perp$ axioms only). But, for a detailed analysis of proof net reduction and of its relations with λ-calculus, that unrestricted η-expansion is unacceptable. Therefore, let us constrain η-expansion to exponential axioms. Namely, the η_e-expansion replaces each $!A, ?A^\perp$ axiom with a box containing the axiom $A, A^\perp$ and a dereliction link from $A^\perp$ to $?A^\perp$. Every reduction of $M \in \mathsf{PN}$ is simulated by a reduction of its η_e-expansion, and similarly for $M \in \mathsf{PN}_C$. Therefore and w.l.o.g., in the following, we shall restrict PN to the case without exponential axioms. In this way, $\mu : \mathsf{PN} \to \mathsf{PN}_C$ is total.

Proposition 1. *Let $M \in \mathsf{PN}$. For every $r : \mu(M) \xrightarrow{C} P$, there is a non-empty $\rho : M \to^* N$ s.t. $P = \mu(N)$. Therefore, PN_C is SN and CR, and $\mathsf{nf}_C(\mu(M)) = \mu(\mathsf{nf}(M))$.*

The obvious limitation of this approach is that its reduction is too coarse grained: it really performs in one single step all the duplication, erasure and unboxing operations involved in a β-reduction step for the λ-calculus. For this reason, if one wants to study finer reductions on the λ-terms, like the ones involved in handling explicit substitutions, this system turns out to be inadequate: it throws out the baby with the bath water.

2.2 Simulating explicit substitutions: fusion and splitting of contraction links

In [5], the limitations of both PN and PN_C are recognised, and another system is proposed, where it is possible to fuse two *n*ary contraction links together (see the *fusion* rule in Fig. 5) and where the irrelevance of the order of contraction and box formation is taken into account via a reduction rule that allows to push some contractions inside a box (see the push rule in Fig. 5).

This approach is less coarse grained, and it was the first solution for interpreting explicit substitutions in PN, but it still suffers from a certain rigidity of the extended reductions, that makes the translation from λ-calculus with explicit substitutions into PN cumbersome (while the propagation of the substitutions is faithfully mirrored, the translation of a cut forces all the duplications to be performed at once).

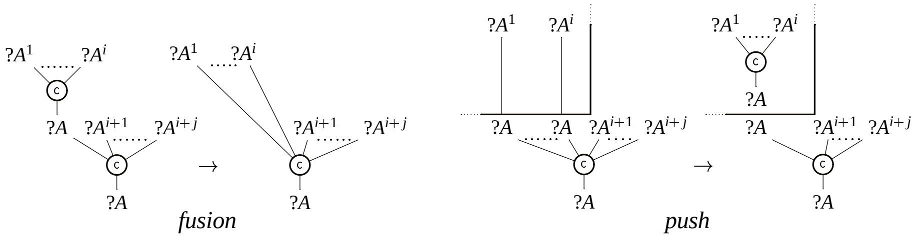

Fig. 5. Fusion and push.

2.3 Our approach: rewriting modulo an equivalence relation

If one looks carefully at the previous approaches, one really finds out that they are both trying to handle contraction links as associative-commutative operators freely floating in and out of boxes: Danos and Regnier work on a representative of the *AC* (associative-commutative) equivalence class which is obtained by collapsing all the trees of exponential links and pushing them outside of all boxes; Di Cosmo and Kesner allow a finer control on how to collapse and push in or out of boxes the contraction links.

The limitations of the previous approaches clearly point out the need of a more flexible system, which accepts explicitly the associative-commutative nature of the contraction operator, allowing a finer control of duplication and propagation of substitutions in the nets. For this reason, we introduce an equivalence relation $\sim$ on Proof Nets and define reduction on the corresponding equivalence classes.

Definition 3 **(PN_{AC}).** *The equivalence relation $\sim$, named AC, is the context closure of the graph equivalences in Fig. 6. Let us extend the reduction of* PN *to the equivalence classes of Proof Nets as $M \xrightarrow{AC} N$ iff $\exists M', N' : M \sim M' \to N' \sim N$. We shall write* PN_{AC} *for Proof Nets equipped with this new reduction.*

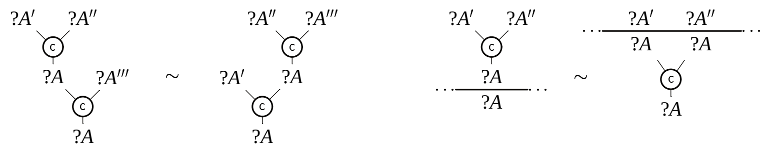

Fig. 6. *AC* congruence.

That extension of PN preserves the normal forms, as shown by the next proposition, which proves indeed that PN_{AC} is a fine analysis of PN_C.

Proposition 2. *For every $M, N \in \mathsf{PN}_{AC}$, $M \sim N$ iff $\mu(M) = \mu(N)$. Then, let $M \xrightarrow{AC}{}^* N$.*

1. *There are $\mu(M) \xrightarrow{C}{}^* P$ and $\mu(N) \xrightarrow{C}{}^* P$.*
2. $\mathsf{nf}_{\mathsf{C}}(\mu(M)) = \mathsf{nf}_{\mathsf{C}}(\mu(N))$ *and* $\mathsf{nf}(M) \sim \mathsf{nf}(N)$.

3 Main results

The main result of the paper is that PN_{AC} is strongly normalizing and has the unique normal form property (modulo *AC*).

Theorem 2. *Let $M \in \mathsf{PN}_C$.*

1. *Let $M \stackrel{C}{\to}^* N$ with N cut-free. Then $N \sim \mathsf{nf}(M)$.*
2. *Every reduction of M is finite.*

The first item is a trivial consequence of Proposition 2 (a particular case of it). The proof of strong normalization is by reduction to termination of PN_C.

3.1 Overview of the proof technique

The key point in relating PN_{AC} to PN_C is the study of the so-called *persistent paths*, an invariant introduced by Geometry of Interaction. Persistent paths capture the intuitive idea that every connection (path) between the nodes of a reduct N of M is the deformation of some connection (path) between the nodes of M (see [4]). In fact, along the reduction of M certain connections are broken (*e.g.*, take the path between A and $B^\perp$ in the multiplicative rule), while others *persist*; in particular, the paths that persist after every reduction yield the normal form. Geometry of Interaction is an algebraic formulation of the previous notion of path deformation, even if the idea 'reduction as path composition' was already implicit in Lévy labelled λ-calculus. For a survey on the relations between persistent paths, Lévy's labels and Geometry of Interaction *regular paths* see [1].

Persistent paths will be defined and studied in section 4. There, we shall assign a norm to every $M \in \mathsf{PN}_{AC}$ in terms of the persistent paths of $\mu(M)$ (actually, in terms of the persistent paths that do not collapse). That norm is decreased by the reductions of PN_{AC} with a correspondence in PN_C, while it is left unchanged by duplication and commutative conversion. In section 5, we shall analyze the transformations that simulate duplication and commutative conversion in PN_C. That analysis will lead us to define a second norm (section 5.4) that is decreased by every one-step reduction.

Unfortunately, the previous proof schema does not work if directly applied to PN_C and PN_{AC}. In fact, in order to fully exploit it, we must tackle two technical difficulties.

The first problem is connected with duplication: we need a way to count the number of box duplications in a reduction. For that purpose, instead of resorting to some measure defined on the whole reduction, we exploit the presence of weakening. Namely, using weakening, we define a proof structure $T^\checkmark$, a *tick* (see section 5.3), that reduces to the empty net and s.t. the proof structure $M^\checkmark$ obtained by inserting a tick into each box of M is a proof net. Since each box duplication duplicates a tick, the number of boxes duplicated in a reduction is equal to the number of new ticks in the result.

The second problem is that ticks might disappear along the reduction because of an erasing rule. Thus, in order to preserve our counting device, we have to delay garbage collection until the end of the computation (indeed, this approach simplifies other technical parts also). Namely, let us denote by $M \xrightarrow[\neg\mathsf{w}]{AC}{}^* N$ a reduction that does not contain erasing rules and by $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ the restriction of PN_{AC} to that non-erasing reduction.

Lemma 1. *For every $M \in \mathsf{PN}_{AC}$, if $M \xrightarrow{AC}{}^{*} N$ then $M \xrightarrow[\neg\mathsf{w}]{AC}{}^{*} P \xrightarrow{AC}{}^{*}_{\mathsf{w}} N$. Therefore, PN_{AC} is terminating iff $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ is terminating.*

Henceforth, we shall restrict to the study of $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ and of the corresponding system $\mathsf{PN}_{C}^{\neg\mathsf{w}}$, *i.e.*, PN_{C} restricted to the non-erasing reduction $\xrightarrow[\neg\mathsf{w}]{AC}$. That analysis will conclude with the proof of strong normalization of $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ (Lemma 13) that, by Lemma 1, proves the strong normalization of PN_{AC} as well.

4 Paths in $\mathsf{PN}_{C}^{\neg\mathsf{w}}$

A *path* in a proof net M is an undirected path in the graph of M that, crossing any link but axiom and cut, moves from a premise to the conclusion of the link and that, crossing an axiom/cut, moves from one conclusion/premise of the axiom/cut to the other conclusion/premise.

Let M be a proof net. We shall denote by $\Phi(M)$ the set of its paths and we shall write $\psi \sqsubseteq \phi$ to denote that ψ is a subpath of ϕ. Remarkably, when M is in normal form, $\Phi(M)$ is finite and is the set of the *elementary paths* of M (a path is elementary when it does not cross any cut); instead, when M contains cuts, the paths of M may loop and $\Phi(M)$ may be infinite.

4.1 Persistent and permanent paths

After a reduction step, paths deform or even vanish, so there is a natural notion of *residual* of a path along a proof net reduction: as in [4], this notion can be captured by associating to every $r : M \xrightarrow[\neg\mathsf{w}]{C} N$, a function $\overline{r} : \Phi(N) \xrightarrow[\neg\mathsf{w}]{C} \Phi(M)$ that maps a path of N to its *ancestor* in M. The notion of residual extends to a reduction $\rho = r_0 r_1 \ldots r_k$ by function composition, *i.e.*, $\overline{\rho} = \overline{r_0} \cdot \overline{r_1} \cdot \ldots \cdot \overline{r_k}$.

We remark that $\overline{\rho}$ is total; that is, for $\rho : M \xrightarrow[\neg\mathsf{w}]{C}{}^{*} N$, every path $\phi \in \Phi(N)$ is the deformation of some path in M. Moreover, every deformed path ϕ results from the contraction to a node of some subpath of $\rho(\phi)$; therefore, either ϕ is essentially the same as $\overline{\rho}(\phi)$, or $|\phi| < |\overline{\rho}(\phi)|$. However, $\overline{\rho}$ is not onto. In fact, a path of M disappears in the following cases:

1. The path contracts to a connection between the premises of a cut that is then reduced along ρ (*e.g.*, the path between $A \parr B$ and $A^{\perp} \otimes B^{\perp}$ in Fig. 7).
2. The execution of a multiplicative or exponential cut disconnects the path. For instance, take the dashed path in the right-hand side of Fig. 7.

The two cases above correspond to two completely different phenomena. In the first case, the path disappears enclosed into a longer path that eventually contracts to a formula. In the second case, the reduction *splits* the path. Thus, in the first case, we can say that the path *persists* along the reduction, as a trace of it is still present in the resulting proof net; in the second case, the path has no image in the result.

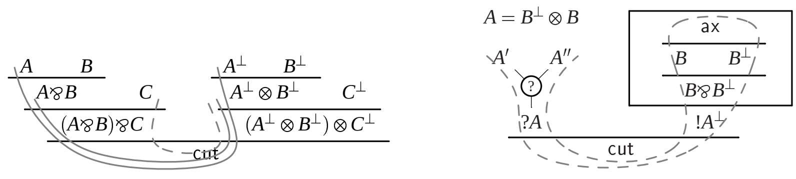

Fig. 7. Paths.

Definition 4 (persistent paths). *Let* $\rho : M \xrightarrow[\neg w]{C}{}^{*} N$. *A path* $\phi \in \Phi(M)$ *is* ρ-persistent *when there is* $\psi \in \Phi(N)$ *s.t.* $\phi \sqsubseteq \overline{\rho}(\psi)$. *The* ρ*-persistent path* ϕ *is said* ρ-permanent *when* $\phi = \overline{\rho}(\psi)$ *for some* $\psi \in \Phi(N)$. *A path of M is* persistent, *or* permanent, *when it is* ρ*-persistent, or* ρ*-permanent, for every reduction* ρ *of M.*

Henceforth, $\Psi(M)$ will denote the set of the permanent paths of M and $\Psi_{\sqsubseteq}(M)$ will denote the set of its persistent paths. By definition, $\Psi_{\sqsubseteq}(M)$ is a superset of the closure by subpaths of $\Psi(M)$; further, we shall prove that $\Psi_{\sqsubseteq}(M)$ is that closure, see Lemma 4.

Lemma 2. *Let* $M \in \mathsf{PN}_C^{\neg w}$. *Every occurrence of formula in M is persistent.*

Therefore, the set of the persistent paths is not empty. Indeed, it is readily seen that every path corresponding to a redex (*i.e.*, every cut pair $A, A^{\perp}$) is persistent. Moreover, every *virtual redex*, *i.e.*, every path that along some reduction will eventually reduce to a cut pair, is persistent, see [4] and [1].

4.2 Folding and unfolding of permanent paths

The permanent paths of a proof net M are the connections of M that are invariant under any reduction. So we expect that $\Psi(M)$ be an image of $\mathsf{nf}_C^{\neg w}(M)$; that is, we expect $\Psi(M) = \overline{\rho}(\Phi(\mathsf{nf}_C^{\neg w}(M)))$, for any normalizing reduction ρ. However, that equivalence is not immediate. In fact, though $\mathsf{PN}_C^{\neg w}$ has the unique normal form property, two distinct reductions might build the same path of $\mathsf{nf}_C^{\neg w}(M)$ by combining different paths of M.

Lemma 3. *Let* $M \in \mathsf{PN}_C^{\neg w}$. *For every* $r_1 : M \xrightarrow[\neg w]{C} M_1$ *and* $r_2 : M \xrightarrow[\neg w]{C} M_1$, *there exist* $\rho_1 : M_1 \xrightarrow[\neg w]{C}{}^{*} N$ *and* $\rho_2 : M_2 \xrightarrow[\neg w]{C}{}^{*} N$, *s.t.* $\overline{r_1\rho_1} = \overline{r_2\rho_2}$.

Proposition 3. *Let* $N = \mathsf{nf}_C^{\neg w}(M)$. *There is a canonical map* $\mathsf{fold}_M : \Phi(N) \to \Phi(M)$ *s.t.* $\mathsf{fold}_M = \rho$, *for every* $\rho : M \xrightarrow[\neg w]{C}{}^{*} N$. *Moreover,* $\Psi(M) = \mathsf{fold}_M(\Phi(N))$.

The previous proposition proves the soundness of the definition of permanent paths. Moreover, let $\rho : M \xrightarrow[\neg w]{C}{}^{*} N$; it proves that the restriction of $\overline{\rho}$ to permanent paths is an onto map $\widehat{\rho} : \Psi(N) \xrightarrow[\neg w]{C}{}^{*} \Psi(M)$ (this is a consequence of $\Psi(M) = \mathsf{fold}_M(\Phi(P)) = \overline{\rho} \cdot \mathsf{fold}_N(\Phi(P))$, where $P = \mathsf{nf}_C^{\neg w}(M) = \mathsf{nf}_C^{\neg w}(N)$). We stress that $\overline{\rho}(\Psi(N)) = \Psi(M)$ is not a trivial consequence of the definition of permanent paths, as that definition trivially implies $\overline{\rho}(\Psi(N)) \supseteq \Psi(M)$ only. Finally, as a corollary of Proposition 3, we get that every persistent path can be prolongated to a permanent path.

Lemma 4. *For every* $\phi \in \Psi_{\sqsubseteq}(M)$*, there is* $\psi \in \Psi(M)$ *s.t.* $\phi \sqsubseteq \psi$*.*

The *unfolding* of $\phi \in \Phi(M)$ is the set of its residuals in the normal form, *i.e.*,

$$\mathsf{unfold}_M(\phi) = \{\psi \in \Phi(\mathsf{nf}_{\mathsf{C}}^{\neg\mathsf{w}}(M)) \mid \mathsf{fold}_M(\psi) = \phi\} = \mathsf{fold}_M^{-1}(\phi)$$

The *cardinality* of a path is the cardinality of its unfolding, *i.e.*,

$$\#(\phi) = |\mathsf{unfold}_M(\phi)|$$

By definition, $\#(\phi) > 0$ iff $\phi \in \Psi(M)$. Thus, $\sum\{\#(\phi) \mid \phi \in \Phi(M)\} = \sum\{\#(\phi) \mid \phi \in \Psi(M)\} = |\Phi(\mathsf{nf}_{\mathsf{C}}^{\neg\mathsf{w}}(M)|$; that is another way to express the combinatorial fact that no finite reduction creates an infinite number of residuals (*i.e.*, $\#(\phi)$ is always finite).

4.3 The norm of $\mathsf{PN}_C^{\neg\mathsf{w}}$

In the reduction of $\mathsf{PN}_C^{\neg\mathsf{w}}$ we have two distinct phenomena. On one side, exponential reductions tend to unfold permanent paths, increasing their number; on the other side, every reduction reduces the length of some permanent path. The previous considerations summarize in the following lemma (as usual, $|\phi|$ denotes the length of the path ϕ, while $\overline{\rho}^{-1}(\phi) = \{\psi \mid \overline{\rho}(\psi) = \phi\}$).

Lemma 5. *Let* $\rho : M \xrightarrow[\neg\mathsf{w}]{C}{}^* N$*. For every* $\phi \in \Psi(M)$*,*

1. $\#(\phi) = \sum\{\#(\psi) \mid \psi \in \overline{\rho}^{-1}(\phi)\}$*;*
2. $|\phi| \geqslant |\psi|$*, for every* $\psi \in \overline{\rho}^{-1}(\phi)$*.*
3. *Moreover, if* ρ *is not empty and is not a sequence of weakening duplications, then* $|\phi| > |\psi|$ *for some* $\phi \in \Psi(M)$*.*

Let us equip $\mathsf{PN}_C^{\neg\mathsf{w}}$ with the following norm:

$$\|M\|_C^{\phi} = \sum\{\#(\phi) \cdot |\phi| \mid \phi \in \Phi(M)\} = \sum\{\#(\phi) \cdot |\phi| \mid \phi \in \Psi(M)\}$$

We remark that, since $\Psi(M)$ is finite, $\|M\|_C^{\phi}$ is well-defined (*i.e.*, it is finite).

Lemma 6. *For every* $\rho : M \xrightarrow[\neg\mathsf{w}]{C}{}^* N$*,* $\|N\|_C^{\phi} \leqslant \|M\|_C^{\phi}$*. Moreover, when* ρ *is not empty and is not a sequence of weakening duplications,* $\|N\|_C^{\phi} < \|M\|_C^{\phi}$*.*

5 Relating $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ to $\mathsf{PN}_C^{\neg\mathsf{w}}$

The grain of the reduction in $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ is finer than in $\mathsf{PN}_C^{\neg\mathsf{w}}$. In particular, the commutative conversion and the duplication rule have no correspondence in $\mathsf{PN}_C^{\neg\mathsf{w}}$; moreover, in $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ we reduce modulo *AC*. For the part of $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$ with a direct correspondence in PN_C the situation is clear: since $M \xrightarrow[\neg\mathsf{w}]{AC} N$ implies $\mu(M) \xrightarrow[\neg\mathsf{w}]{C} \mu(N)$, this part of the system is strongly normalizing and

$$\|M\|_{AC}^{\phi} = \|\mu(M)\|_C^{\phi} \qquad \text{for } M \in \mathsf{PN}_{AC}^{\neg\mathsf{w}}$$

seems the natural candidate for expressing that property. For the remaining part of $\mathsf{PN}_{AC}^{\neg\mathsf{w}}$, let us analyze each rule separately.

5.1 Commutative conversion

When $r : M \xrightarrow{AC}_{\mathsf{com}} N$, $\mu(N)$ and $\mu(M)$ are equal but for some boxes of $\mu(M)$ that have been moved inside some other box of $\mu(N)$, see Fig. 8.

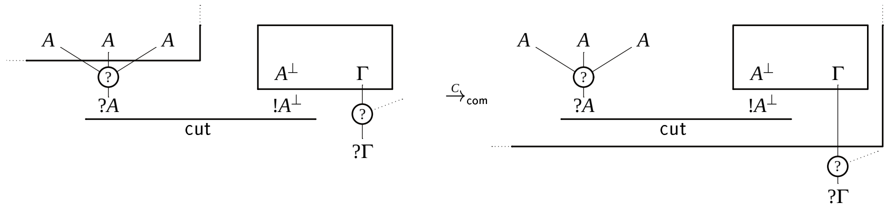

Fig. 8. Commutative conversion in PN_C.

Lemma 7. *Let $r : M \xrightarrow{AC}_{\mathsf{com}} N$.*

1. $\mathsf{nf}_{\mathsf{C}}^{\neg\mathsf{w}}(\mu(M)) = \mathsf{nf}_{\mathsf{C}}^{\neg\mathsf{w}}(\mu(N))$;
2. $\mathsf{fold}_{\mu(M)} = \mathsf{fold}_{\mu(N)}$ *and* $\Psi(\mu(M)) = \Psi(\mu(N))$;
3. $\|M\|_{AC}^{\phi} = \|N\|_{AC}^{\phi}$.

Therefore, the commutative conversion induced on PN_C preserves normal forms and persistent paths. Moreover, though it does not decrease the norm on paths, it is readily seen that we cannot have an infinite sequence of commutative conversions.

Definition 5 (depth). *The depth of an !-link, and then of the corresponding box, is the number of boxes that encapsulate it. The* depth $\partial(M)$ *of a proof net M is the sum of the depths of its !-links.*

Let $\mathsf{n}^!(M)$ be the number of !-links in M. We define

$$\|M\|^! = \mathsf{n}^!(M)^2 - \partial(M)$$

Lemma 8. *For any $M \in \mathsf{PN}_{AC}^{\neg\mathsf{w}}$.*

1. $\|M\|^! \geqslant 0$.
2. *If $r : M \xrightarrow{AC}_{\mathsf{com}} N$, then $\|N\|^! < \|M\|^!$.*

5.2 Duplication

This is the trickiest case. Fig. 9 illustrates by means of an example the transformation $\delta_r : \mu(M) \xrightarrow{AC}_{\mathsf{dup}} \mu(N)$ corresponding to $r : M \xrightarrow{C}_{\mathsf{dup}} N$. In that example, we assume that the contraction c in the redex r join two exponential subtrees whose leaves are A' and A'', A''', respectively; that two sets of leaves are the premises of the two new instances of c in $\mu(N)$. As every rule in PN_C, δ_r defines a map $\overline{\delta_r} : \Phi(\mu(N)) \to \Phi(\mu(M))$.

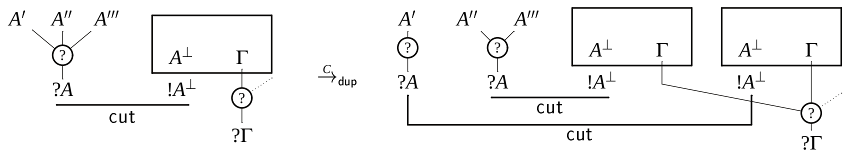

Fig. 9. Duplication in PN_C.

Lemma 9. *Let* $r : M \xrightarrow{AC}_{\mathsf{dup}} N$.

1. $\mathsf{nf}_{\mathrm{C}}^{\neg\mathsf{w}}(\mu(M)) = \mathsf{nf}_{\mathrm{C}}^{\neg\mathsf{w}}(\mu(N))$;
2. $\mathsf{fold}_{\mu(M)} = \overline{\delta_r} \cdot \mathsf{fold}_{\mu(N)}$ *and* $\Psi(\mu(M)) = \overline{\delta_r}(\Psi(\mu(N)))$;
3. $\|M\|^{\phi}_{AC} = \|N\|^{\phi}_{AC}$.

5.3 Ticked proof nets

Usually, the proof that duplication is terminating exploits the fact that, in a sequence of duplications, no box is duplicated twice by the same contraction link—this is the intuitive idea; formally, we should reason in terms of residuals. However, since we assume to know that $\mathsf{PN}_C^{\neg\mathsf{w}}$ is strongly normalizing, we can resort to a technical trick.

Duplication does not decrease the length of any permanent path. So, in order to prove that it is terminating, we need a measure of the unfolding that it causes. The remark that duplication tends to increase the number of persistent paths seems unfruitful: unfortunately, there are $M \xrightarrow{AC}_{\mathsf{dup}} N$ for which $|\Psi(\mu(M))| = |\Psi(\mu(N))|$. For instance, the proof net M in Fig. 10 reduces to an axiom; so the path ϕ drawn in the figure is the only non-empty permanent path of M. The path ϕ contains two occurrences of the path ψ (rooted at $!(A \parr A^{\perp})$) that loops inside the box, *i.e.*, $\phi = \phi_0\psi\phi_1\psi\phi_2$. After $M \xrightarrow{C}_{\mathsf{com}} N$, the residual of ϕ is $\phi' = \phi_0'\psi'\phi_1'\psi''\phi_2'$, where ψ' and ψ'' are residuals of ψ that loop inside two distinct boxes of N. In other words, instead of duplicating some permanent path, the duplication in M unfolds the loop described by the unique permanent path in the proof net. The situation would be different if the box B in M would contain a permanent path: that path would be duplicated by the duplication of B.

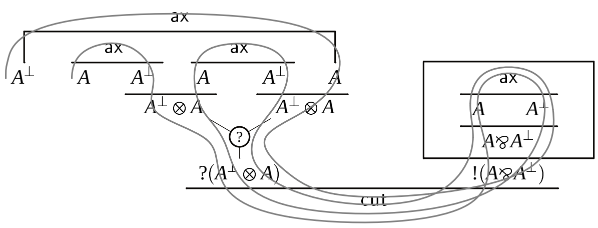

Fig. 10. Unfolding the loop of a permanent path.

Lep p be any atomic formula. A *tick* of PN_C is a proof structure $T^\checkmark$ as that in Fig. 11; and $\mu(T^\checkmark)$ is a tick of PN_{AC}. A tick is not a proof net but, for every $N \in \mathsf{PN}_C$, the proof structure $M = N \cup T^\checkmark$ obtained by attaching the tick $T^\checkmark$ to N is a proof net (*i.e.*, $M \in \mathsf{PN}_C$); moreover, $M \xrightarrow{AC}_{\mathsf{w}} N$, by contraction of the weakening cut in $T^\checkmark$. Therefore, let N be the interior of the most external box of some proof net; by replacing M for N, we get a ticked box $B \in \mathsf{PN}_C$. Then, by recursive application of this ticking procedure to the boxes in B, we eventually get a proof net whose boxes are all ticked.

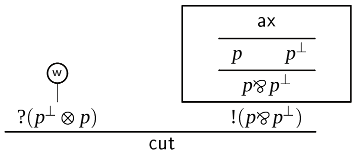

Fig. 11. A tick.

Definition 6 ($\mathsf{PN}_{AC}^\checkmark$). *A box contains (at least) a tick when its interior is a proof net $B \cup T^\checkmark$ and $T^\checkmark$ is a tick. A proof net of PN_{AC} is* ticked *when each of its boxes contains a tick. Let us denote by $\mathsf{PN}_{AC}^\checkmark$ the set of the ticked proof nets of PN_{AC}. We say that $M^\checkmark \in \mathsf{PN}_{AC}^\checkmark$ is a* ticking *of $M \in \mathsf{PN}_{AC}$ when M can be obtained from $M^\checkmark$ by erasing some of its ticks.*

The set of the ticked proof nets $\mathsf{PN}_{AC}^\checkmark$ is closed by reduction, *i.e.*, for any $M^\checkmark \in \mathsf{PN}_{AC}^\checkmark$ and any $\rho : M^\checkmark \xrightarrow[\neg\mathsf{w}]{AC}{}^{*} N^\checkmark$, $N^\checkmark \in \mathsf{PN}_{AC}^\checkmark$. In the following, $M^\checkmark$ will always denote some ticking of $M \in \mathsf{PN}_C$ (by the way, there exists at least one $M^\checkmark$ for every M). By definition, $M^\checkmark \xrightarrow{AC}{}^{*}_{\mathsf{w}} M$, for any $M^\checkmark$.

Lemma 10. *The $\xrightarrow[\neg\mathsf{w}]{AC}$-reduction of $M \in \mathsf{PN}_{AC}$ is terminating iff the $\xrightarrow[\neg\mathsf{w}]{AC}$-reduction of any $M^\checkmark$ is terminating.*

By Lemma 10, strong normalization of PN_{AC} reduces to that of $\mathsf{PN}_{AC}^\checkmark$. Moreover, as the ticks of $M^\checkmark$ are permanent, duplication is not a problem in $\mathsf{PN}_{AC}^\checkmark$. In fact, let $\mathsf{n}^\checkmark(M)$ be the number of ticks in M. For any $M \in \mathsf{PN}_{AC}^{\neg\mathsf{w}}$, we define

$$\|M\|_{AC}^\checkmark = \|\mu(M)\|_C^\checkmark \quad \text{where} \quad \|P\|_C^\checkmark = \mathsf{n}^\checkmark(\mathsf{nf}_C^{\neg\mathsf{w}}(P)) - \mathsf{n}^\checkmark(P) \quad \text{for } P \in \mathsf{PN}_C^{\neg\mathsf{w}}$$

Lemma 11. *For any $M^\checkmark \in \mathsf{PN}_{AC}^\checkmark$.*

1. $\|M^\checkmark\|_{AC}^\checkmark \geqslant 0$.
2. *If $r : M^\checkmark \xrightarrow[\neg\mathsf{w}]{AC} N^\checkmark$, then $\|N^\checkmark\|_{AC}^\checkmark \leqslant \|M^\checkmark\|_{AC}^\checkmark$; moreover, $\|N^\checkmark\|_{AC}^\checkmark < \|M^\checkmark\|_{AC}^\checkmark$, when r is a duplication.*

5.4 The norm of $\mathsf{PN}_{AC}^{\checkmark}$

Let $M \in \mathsf{PN}_{AC}^{\neg w}$, we take

$$\langle\!\langle M \rangle\!\rangle_{AC} = \langle\ \|M\|_{AC}^{\phi} + \|M\|_{AC}^{\checkmark},\ \|M\|^{!}\ \rangle$$

with the lexicographic ordering, *i.e.*, $\preccurlyeq$ is the reflexive closure of $\langle a_1, b_1 \rangle \prec \langle a_2, b_2, \rangle$ iff $(a_1 < a_2)$ or $(a_1 = a_2 \wedge b_1 < b_2)$. By definition, $\langle 0, 0 \rangle \preccurlyeq \langle\!\langle M \rangle\!\rangle_{AC}$.

We remark that, for any $M, N \in \mathsf{PN}_{AC}^{\neg w}$, $M \sim N$ implies $\langle\!\langle M \rangle\!\rangle_{AC} = \langle\!\langle N \rangle\!\rangle_{AC}$.

Lemma 12. *Let $M^{\checkmark} \in \mathsf{PN}_{AC}^{\checkmark}$. For every $r : M^{\checkmark} \xrightarrow[\neg w]{AC} N^{\checkmark}$, $\langle\!\langle N^{\checkmark} \rangle\!\rangle_{AC} \prec \langle\!\langle M^{\checkmark} \rangle\!\rangle_{AC}$.*

Lemma 13. $\mathsf{PN}_{AC}^{\neg w}$ *is strongly normalizing.*

6 Conclusions and future work

We have presented here for the first time a proof of strong normalization for Multiplicative Exponential Linear Logic's Proof Nets with an associative-commutative contraction free to float in and out of proof boxes. This is interesting for several reasons.

First, this is another significative application of the *normalization by persistent paths* slogan which can be found in Girard's Geometry of Interaction. But also, now that we know that we can rearrange contraction trees as we like during a reduction of a proof net, and still have the strong normalization property, we can go back to analyse how the classical β-reduction of the lambda calculus, or the more refined reductions of calculi with explicit substitutions are simulated in our system. We expect not only to be able to provide a much simpler simulation than the ones in the literature, but also to extract from PN_{AC} a calculus of explicit substitutions with good properties.

References

[1] A. Asperti, V. Danos, C. Laneve, and L. Regnier. Paths in the lambda-calculus: three years of communications without understanding. In *Proceedings, Ninth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 426–436, Paris, France, 1994.

[2] V. Danos. *Une Application de la Logique Linéaire à l'Ètude des Processus de Normalisation (principalement du λ-calcul)*. PhD Thesis, Université Paris 7, Paris, June 1990.

[3] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[4] V. Danos and L. Regnier. Proof-nets and the Hilbert space. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 307–328. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[5] R. Di Cosmo and D. Kesner. Strong normalization of explicit substitutions via cut elimination in proof nets. In *Proceedings, Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 35–46, Warsaw, Poland, 1997. Full Paper available as `http://www.dmi.ens.fr/~dicosmo/Pub/esll.ps.gz`.

[6] J.-Y. Girard. Linear logic. *Theoretical Comput. Sci.*, 50(1):1–102, 1987.

[7] S. Guerrini. Correctness of multiplicative proof nets is linear. In *Proceedings, Fourteenth Annual IEEE Symposium on Logic in Computer Science (LICS)*, Trento, Italy, 1999.

[8] J.-B. Joinet. *Étude de la normalisation du calcul des séquents classique à travers la logique linéaire*. PhD Thesis, Université Paris 7, Paris, 1993.