

Linear Logic and Strong Normalization

Beniamino Accattoli

Carnegie Mellon University - Pittsburgh, PA, USA

Abstract

Strong normalization for linear logic requires elaborated rewriting techniques. In this paper we give a new presentation of MELL proof nets, without any commutative cut-elimination rule. We show how this feature induces a compact and simple proof of strong normalization, via reducibility candidates. It is the first proof of strong normalization for MELL which does not rely on any form of confluence, and so it smoothly scales up to full linear logic. Moreover, it is an *axiomatic* proof, as more generally it holds for every set of rewriting rules satisfying three very natural requirements with respect to substitution, *commutation with promotion*, *full composition*, and *Kesner's IE property*. The insight indeed comes from the theory of explicit substitutions, and from looking at the exponentials as a substitution device.

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.3.2 Semantics of Programming Languages

Keywords and phrases Linear logic, proof nets, strong normalization, explicit substitutions.

Introduction

Normalization is probably the most studied topic in proof theory and in the theory of functional programming languages, not to speak of rewriting theory. It comes in two flavors, *weak* or *strong* normalization. Weak normalization (WN) holds when there is an evaluation sequence reaching a normal form. This is the relevant notion of normalization, since for instance in proof theory it suffices to establish the completeness of the cut-free sub-system. Strong normalization (SN) is the variant where *all* evaluation sequences terminate, and it is mandatory when one is interested in exploring and comparing different evaluation strategies.

Sometimes, the gap between proving weak or strong normalization is minimal; for instance for the simply typed λ -calculus there is an extremely short argument for SN, due to van Daalen [34]. Some other times the gap is huge; *e.g.* for λ -calculi with explicit substitution (ES) [20] or for linear logic proof nets [28, 29]. The occasional increment of difficulty for SN is apparently related to the combinatorics of the rewriting system, independently of the logic/type system. This point of view seems to be justified by two facts. First, the proofs of SN for ES-calculi are usually much harder than for λ -calculus, even if the underlying type system is kept unchanged¹. Second, the only proof method for proving SN for full linear logic starts by proving WN and then obtains SN using the conservation theorem², whose proof relies on a special form of local confluence and that holds even in untyped proof nets/calculi [28, 29]; this proof technique is sometimes called Gandy's method [12, 15] or Nederpelt's method [25, 33].

¹ Most of the time one shows preservation of SN (PSN), *i.e.* that if t is SN with respect to β then it is SN with respect to the ES-calculus X under analysis, rather than SN for a fixed type system. The reason is that PSN is an untyped property which implies SN for X with respect to any typing discipline for which the λ -calculus is SN. Proving PSN or SN for a type system requires the same combinatorial reasoning, but PSN is a more general formulation, independent from the type system.

² The conservation theorem says that any term/net which is weakly normalizing for non-erasing evaluation steps is strongly normalizing for non-erasing evaluation steps.



© Beniamino Accattoli;

licensed under Creative Commons License CC-BY

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\begin{array}{c}
\text{a)} \quad \frac{\frac{\frac{\pi}{\vdash ?\Gamma, A} \quad \frac{\theta}{\vdash \Delta, A^\perp}}{\vdash ?\Gamma, !A} \quad \frac{\quad}{\vdash \Delta, ?A^\perp} \text{d}}{\vdash ?\Gamma, \Delta} \text{cut} \quad \rightarrow \quad \frac{\frac{\pi}{\vdash ?\Gamma, A} \quad \frac{\theta}{\vdash \Delta, A^\perp}}{\vdash \Gamma, \Delta} \text{cut} \\
\text{b)} \quad \frac{\frac{\pi}{\vdash ?\Gamma, !A} \quad \frac{\frac{\theta}{\vdash ?A^\perp, ?\Delta, B}}{\vdash ?A^\perp, ?\Delta, !B}}{\vdash ?\Gamma, ?\Delta, !B} \text{cut} \quad \rightarrow \quad \frac{\frac{\pi}{\vdash ?\Gamma, !A} \quad \frac{\theta}{\vdash ?A^\perp, ?\Delta, B}}{\vdash ?\Gamma, ?\Delta, B} \text{cut}}{\vdash ?A^\perp, ?\Delta, !B} !
\end{array}$$

■ **Figure 1** a) Principal case of cut-elimination; b) Commutative case of cut-elimination.

Proof nets and explicit substitutions both decompose evaluation in small steps. It is then generally believed that the gap in technical efforts between WN and SN is due to the granularity of the rules; the more evaluation is decomposed the harder is the proof of SN. In particular, these rewriting systems lack *orthogonality*, which is sometimes recognized as the reason behind the difficulty [29]. Here we show that this is a misleading point of view. The feature inducing many complications is rather the presence of commutative rewriting rules. We prove this fact by exhibiting a small-step and non-orthogonal presentation of MELL proof nets without any commutative cut-elimination rule, and enjoying a compact, modular, and informative proof of SN.

Commutations and boxes. Any proof of normalization in sequent calculus deals with two kinds of cases, the principal (or key) cases and the commutative cases. Principal cases arise when the last rules of the two cut proofs are those introducing (or contracting) the cut formulas, for instance as in Figure 1.a. The pattern is then replaced by a simpler one where the two last rules have been removed; these are the cases where something logical decreases. A commutative case instead arises when the last rule of one of the two cut proofs is not the one introducing the cut formula, as in Figure 1.b. In these cases no rule is removed. The rewriting consists only in re-arranging the structure of the proof, *i.e.* in *commuting* the cut upwards, in order to get closer to a principal case.

In the study of WN, commutative cases are certainly annoying—because they take most of the proof³—but they can be handled without too many efforts. When studying SN, instead, they become a serious obstacle. In sequent calculus the definition of a strongly normalizing cut-elimination is not evident, because some commutations (for instance of cut with itself) have to be taken as equivalences, otherwise the system has silly diverging reductions. This requires to switch to rewriting modulo, which is quite more technical than plain rewriting.

In his seminal paper on linear logic [14], Girard introduced proof nets, a graphical syntax alternative to sequent calculus. In proof nets deductive rules are disposed on the plane, in parallel, and connected only by their causal relation. There is no *last rule*, and so most commutative cut-elimination cases simply disappear. Unfortunately, to handle the exponentials Girard was forced to introduce *boxes*. They come with the *black-box principle*: "boxes are treated in a perfectly modular way: we can use the box B without knowing its contents, *i.e.*, another box B' with exactly the same doors would do as well" [14].

According to this principle, boxes forbid interaction between their content and their outer

³ Usually, in a system with n rules, each rule gives rise to at most 3 principal cases and at least $n - 4$ commutative cases. Then among the $O(n^2)$ cases to consider, there are only $O(n)$ principal case but $O(n^2)$ commutative cases.

environment. Proof nets have a dedicated commutative rule which brings a box inside another one (rule \rightarrow_{\square} in Figure 4, corresponding to Figure 1.b). Despite having brought a whole bunch of new perspectives and results on cut-elimination, for instance with respect to optimal reductions [5] or implicit computational complexity [16], proof nets have somehow failed in the original intent of simplifying the study of SN: it took 23 years to obtain a complete proof for full linear logic, and the outcome—due to Pagani and Tortora de Falco—requires sophisticated rewriting techniques [28], which are out of scope for most people without a special background in rewriting and proof nets. Moreover, despite the merit of filling an embarrassing hole in the literature (and of the impressive efforts it required), that proof is technically unsatisfactory as it relies on a special form of local confluence, while termination and confluence are independent properties⁴.

In this paper we present an alternative approach, based on the removal of the black-box principle. Boxes are seen as decorations which do not prevent rules to interact through their borders. In some sense this is not a novelty, as a similar approach is taken in classic references on proof nets as [32, 8], where there is an exponential cut-elimination rule mimicking substitution in the λ -calculus. However, what is original here is that we pair this *box-crossing principle* with small-step rules. Following the new principle, our system has no commutative rule. At first sight it may seem to be only a minor variation, but we show that this change has surprising and impressive consequences on the proof of SN.

The rewriting technique. We prove SN using Girard’s reducibility candidates (in the biorthogonal form, as first introduced in [14]), which are the only known technique for SN in a second order setting⁵. We abstract the proof with respect to any set of rewriting rules enjoying three natural properties, two of which are borrowed from the theory of explicit substitutions. We define a notion of (implicit) substitution for proof nets (noted $P\{x/Q\}$), and a notion of explicit substitution (noted $P[x/Q]$), which are essentially given by a big-step exponential rule and by an ordinary exponential cut, respectively. Then, we prove SN for every rewriting relation \rightarrow enjoying:

1. *Commutation of substitution and promotion via \rightarrow* : namely $!(P\{x/Q\}) \rightarrow^* (!P)\{x/Q\}$ must hold. This property is specific to proof nets, and essentially corresponds to the permutation of contractions and weakenings with boxes; these permutations are necessary for the representation of any term calculus and so they are extremely natural.
2. *Full composition*: every explicit substitution can be evaluated fully, and independently by any other one, *i.e.* $P[x/Q] \rightarrow^* P\{x/Q\}$. This property is borrowed from the theory of explicit substitutions, and expresses a form of *context-freeness* for the evaluation of explicit substitution.
3. *Kesner’s IE property [21]*: strong normalization is preserved by expanding Implicit substitutions into Explicit substitutions, namely if $P\{x/Q\} \in SN_{\rightarrow}$ and $Q \in SN_{\rightarrow}$ then $P[x/Q] \in SN_{\rightarrow}$. For ES-calculi this property holds without any typing assumption (proof nets are typed, but the proof of the IE property does not rely on types), and encapsulates the combinatorial content of the strong normalization argument.

Delia Kesner has shown that preservation of strong normalization for ES-calculi can be reduced to the IE property. Such a property is the adaptation to ES of a classic

⁴ Sequent calculus LK for classical logic is strongly normalizing without being confluent, and λ -calculus is confluent without being even weakly normalizing.

⁵ Second order quantifiers are here omitted. The reason is that the difficulties for SN in linear logic are related to the exponentials and not to the quantifiers; in fact—once one embraces reducibility candidates—the treatment of the quantifiers is smooth, and actually forces to deal with some details which make the proof less readable.

expansion lemma in λ -calculus, called *the fundamental lemma of perpetuality* in [37]: if $t\{x/s\}u_1 \dots u_n \in SN_\beta$ and $s \in SN_\beta$ then $(\lambda x.t)su_1 \dots u_n \in SN_\beta$. This lemma holds in the untyped case, and it is the crucial clause in the inductive definition of strongly normalizing λ -terms found independently by van Raamsdonk and Severi [36] and Loader [23]. It appears in most SN arguments, in particular in Girard’s proof for System F ([17], p. 44) or in van Daalen’s short proof cited above. Essentially, IE reduces SN to an inverse preservation property with respect to substitution. The complexity of a proof of SN, then, depends on how nicely the rewriting rules interact with substitution.

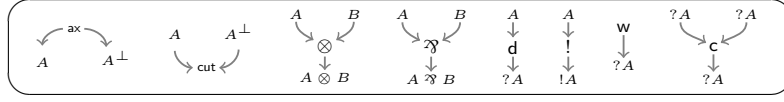
Summing up, we generalize Kesner’s technique to linear logic proof nets, a setting quite richer than λ -calculus, and isolate the role it plays in the reducibility candidates technique. In our commutation-free proof nets the IE property enjoys a simple and direct proof by induction on a triple. In contrast, in presence of commutations the induction does not go through, because commutative rules interfere with substitution, breaking the inductive invariant (see the last paragraph of Section 2).

Our proof can be seen as the equivalent for proof nets of the short proof of SN by van Daalen for the simply typed λ -calculus, but powered to higher-order linear logic via reducibility candidates. It is the first proof not relying on any form of confluence, and this feature—in contrast to all other known proofs of SN for MELL—let it smoothly scale up to full linear logic: the main difficulty posed by the additives is their stubborn reticence to be confluent in presence of the exponentials (case in which Hughes and van Glabbeek’s confluent approach to the additives [18] does not work).

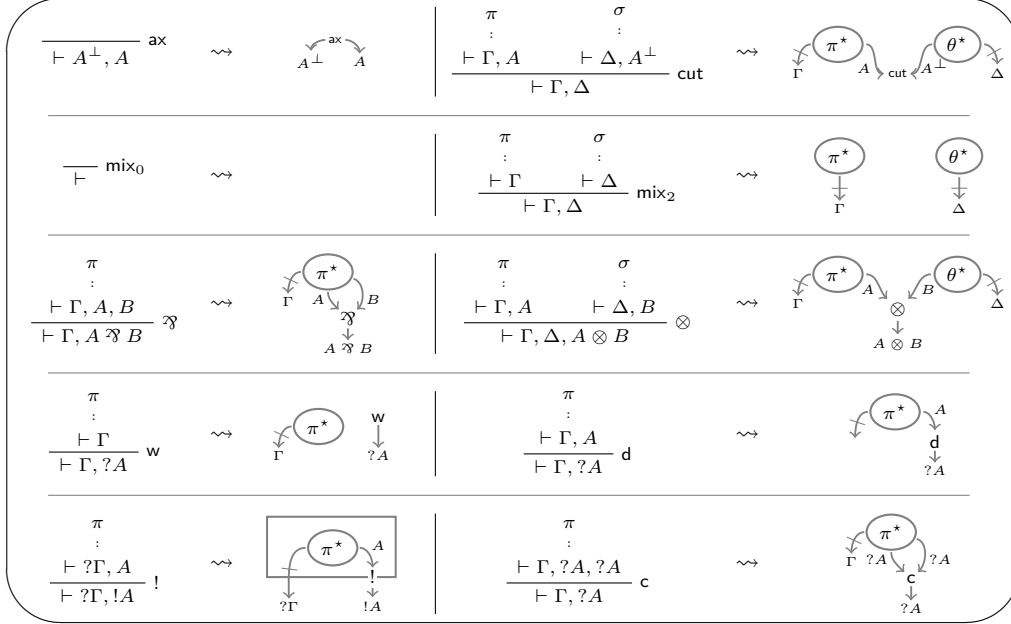
Sometimes, short proofs are not necessarily clear or informative. On the contrary, we believe that the main contribution of this work is the isolation of general and clear properties responsible for SN, making the proof understandable to anyone with a minimum background on proof nets and reducibility candidates. Furthermore, it sheds a new light on the role of commutative rules, and thus its interest goes well beyond the particular case of linear logic.

Related work. The original proof of Girard in [14] depends on a crucial lemma (which plays the role of the IE property here and yet is different) which was not proved in that paper. Later, Danos proved the lemma for second order MELL [7], but the proof is much less direct than the one we give here for the IE property. For MELL there is a proof of SN by Joinet [19], later refined by van Raamsdonk [35], but it does not scale up to second order. The first proof handling the additives is by Tortora de Falco and Pagani, in [28], and uses a conservation theorem. All these results rely on some form of confluence. For WN there also are a semantical proof by Okada [26] (see also the comments at page 53 in [28]) and an elegant formalized proof by Pfenning [31, 30]. A conservation theorem has also been used to prove SN for differential linear logic, by combining the results of Pagani and Tranquilli [29] with those on WN by Pagani [27] (only propositional) or by Gimenez [13] (propositional, but using reducibility candidates). The proof of SN presented here builds on our previous joint work with Kesner [3], and the new presentation of proof nets is a reformulation with explicit boxes of what naturally arises with implicit boxes [2, 1] and corresponds to rewriting rules *at a distance* on term calculi [3]. Finally, in [6] Bonelli studies reducibility candidates and normalization for a system F with explicit substitutions.

Road map. Section 1 defines MELL proof nets. Section 2 introduces implicit and explicit substitutions, and discusses full composition and the IE property. Section 3 introduces the terminology for reducibility candidates and proves SN. Section 4 explains how the result extends to full linear logic with algebraic rules for weakening and contraction.



■ **Figure 2** MELL links.



■ **Figure 3** The translation of MELL sequent proofs to nets.

1 MELL Proof Nets

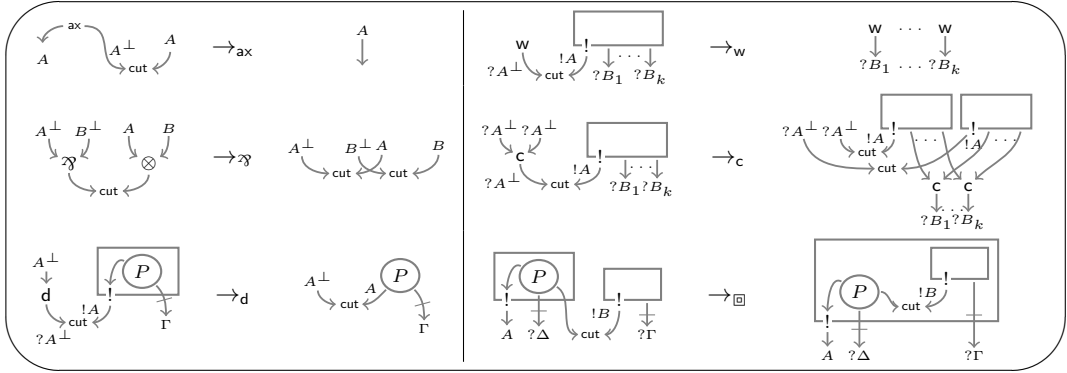
MELL. Multiplicative and Exponential Linear Logic (MELL) formulas are given by:

$$A, B, C ::= X \mid X^\perp \mid A \otimes B \mid A \wp B \mid !A \mid ?A$$

where X and X^\perp are atomic formulas. For the sake of conciseness, the multiplicative units are not considered here, because their cut-elimination is trivial. The sequents of MELL are monolateral, *i.e.* they have the shape $\vdash \Gamma$, where Γ is a multiset of formulas. The rules of MELL are in Figure 3 (left side of every \rightsquigarrow); note the presence of the binary and nullary mix rules.

Nets. Nets are labelled directed graphs with *pending edges*, *i.e.* some edges may not have a source or a target, but not both. Nodes, called **links**, represent deductive rules and are labeled with an element of $\{\mathbf{ax}, \mathbf{cut}, \mathbf{1}, \perp, \otimes, \wp, !, \mathbf{d}, \mathbf{w}, \mathbf{c}\}$. Edges are labeled with a MELL formula. The **conclusions** (**resp.** **premises**) of a link are those represented below (**resp.** above) the link symbol, *e.g.* the \otimes -link has two premises (labeled A and B), and one conclusion ($A \otimes B$). A **?-formula** is a formula $?A$ for some A , and a **?-link** is a link with an edge labeled with a **?-formula**.

► **Definition 1** (MELL net with boxes). A (**MELL**) **net** P is a finite set of links from those in Fig. 2 s.t. every edge is the conclusion of some link. The conclusion edges of P are its pending edges and the **conclusion links** of P are its links with pending edges.



■ **Figure 4** Black-box rewriting rules for MELL proof nets.

A net with boxes P is a net plus for any $!$ -link l a subset $box(l)$ of the links of P , called the **box** of l , s.t. $l \in box(l)$ and:

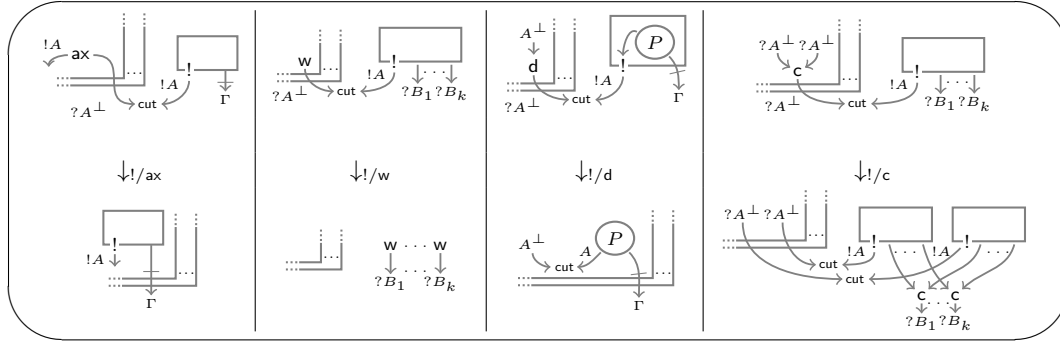
- **Subnet**: its interior $int(l) := box(l) \setminus \{l\}$ is a net with boxes, whose boxes are inherited from P .
- **Border**: the premise of l is a conclusion of $int(l)$, and all other conclusions of $int(l)$ are labeled with $?$ -formulas;
- **Nesting**: For any two $!$ -links l and i if $box(l) \cap box(i) \neq \emptyset$ then $box(l) \subseteq box(i)$ or $box(i) \subseteq box(l)$.

The translation from MELL proofs to nets with boxes is in Fig. 3, where the bar on some edges denotes a (multi)set of conclusions. The original version of the cut-elimination rules for MELL is in Figure 4. The rule \rightarrow_{\boxplus} is the commutative rule. The union of all these rules is noted \rightarrow_{com} (for containing the commutative rule).

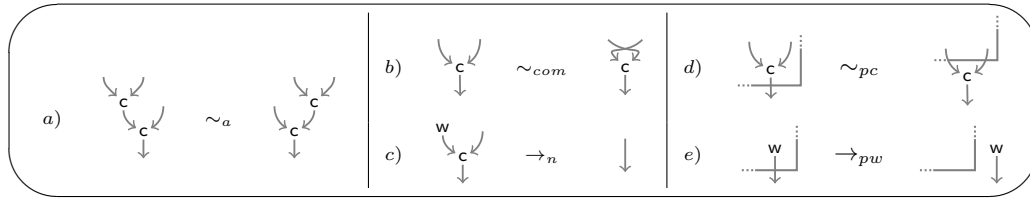
The **level** of a link is the number of boxes in which it is contained (a $!$ -link is not contained in its own box), and the level of a net P is the maximum level of a link of P . The **box address** of a link is the (possibly empty) sequence of $!$ -boxes in which it is contained, starting from the outside. Note that boxes can be represented as labels if every link is decorated with its box address.

Proof nets. A net with boxes P is a **proof net** if it is the translation of a sequent calculus proof π . The proof of strong normalization will be by induction on an inductive decomposition of P , provided by π , and it will use the fact that proof nets reduce to proof nets. One of the motivations for considering MELL extended with the mix rules is that in this case there exist correctness criteria (easy adaptations of those in [11, 4]), while without them no criteria are known. The existence of a correctness criterion implies that the result can be made independent from sequent calculus (and that proof nets are stable by reduction). However, we will not discuss this issue any further.

The box-crossing principle. A crucial point of our approach is the removal of the black-box principle explained in the introduction. Note that we first defined nets, and only afterwards nets with boxes. For us, boxes are a sort of additional layer, a decoration. In the traditional presentation boxes have an *auxiliary port* for every $?$ -conclusion of their interior, which blocks the interaction between the inner and the outer world. In our presentation there are no auxiliary ports, and boxes do not interfere with links, in particular they shall not prevent links to interact through box borders. This box-crossing principle induces a new set of exponential cut-elimination rules, shown in Figure 5. Note that there is no commutative rule, whose task is accomplished by the new axiom, dereliction, and weakening rules. Essentially,



■ **Figure 5** Box-crossing exponential rules for MELL proof nets.

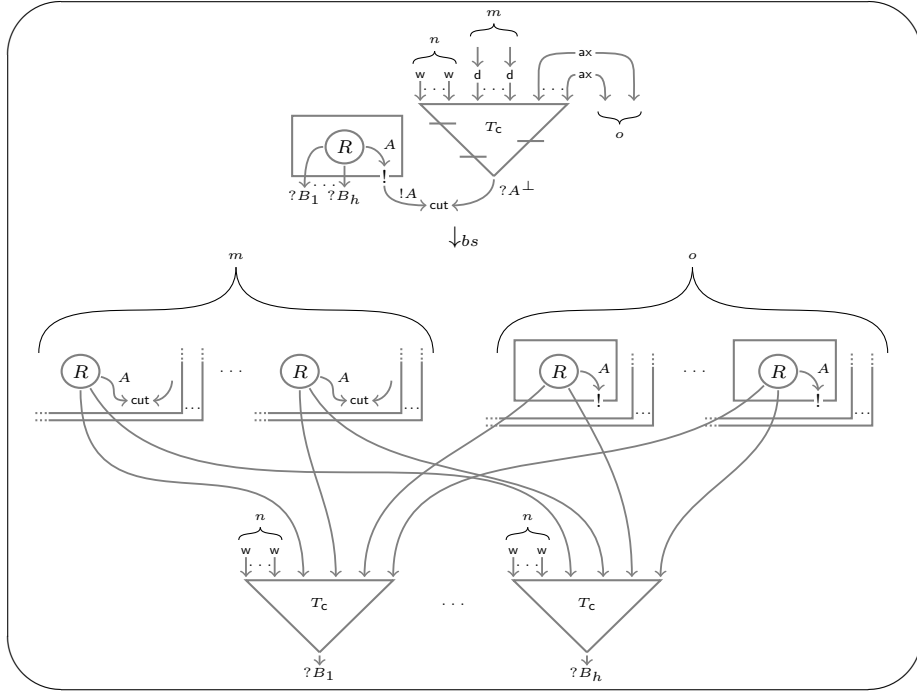


■ **Figure 6** a-c) Co-associativity, co-commutativity, and co-neutrality for contractions; d-e) Permutations with box borders.

commutations are delayed as much as possible and then performed in a single big step together with the axiom/weakening/dereliction cut-elimination. Note that the axiom rule in Figure 4 provides an additional exponential case, when the cut axiom formula is $!A$ for some A . The union of \rightarrow_{ax} , $\rightarrow_{\mathcal{A}}$, $\rightarrow_{!/ax}$, $\rightarrow_{!/w}$, $\rightarrow_{!/d}$, $\rightarrow_{!/c}$ is denoted by \rightarrow_{key} (for having only key—or principal—cut-elimination cases). This dynamics arises naturally when boxes are represented implicitly using *jumps* [2, 1], as in these cases the border of the box is not represented explicitly.

These two sets of rules are complemented by a set of equations and by two additional rules for contraction and weakening, presented in Figure 6. The equation and the rule in Figure 6.d-e permutes the structural rules with box borders, and will have an important role in the proof of strong normalization. The relations in Figure 6.a-c, instead, make contraction and weakening the operations of a co-commutative co-monoid on every type $?A$, *i.e.* they express co-associativity and co-commutativity of contractions and co-neutrality of weakening with respect to contraction. Beyond being algebraically natural and also semantically sound, these complementary operations are necessary in order simulate the synthetic $?A$ -links of [32, 8] and obtain proper representations of λ -calculi or systems of explicit substitutions [10]. Lifting strong normalization to these enriched representations is usually tricky. In [9] Di Cosmo and Guerrini employ non-trivial arguments based on the so-called *geometry of interaction*. In [29] Pagani and Tranquilli have to use sophisticated techniques for rewriting modulo equivalence relations. In our case these additional equations and rules are necessary (see next section) but they will not require any heavy machinery.

We use \equiv for the equivalence generated by the equations in Figure 6.a,b,d; the union of the rules in Figure 6.c,e considered modulo \equiv is denoted by \Rightarrow (*i.e.* $\Rightarrow := \equiv (\rightarrow_n \cup \rightarrow_{pw}) \equiv$); we use \Rightarrow_{key} for the union of \rightarrow_{key} , \rightarrow_n , and \rightarrow_{pw} modulo \equiv (*i.e.* $\Rightarrow_{key} := \equiv (\rightarrow_{key} \cup \rightarrow_n \cup \rightarrow_{pw}) \equiv$).



■ **Figure 7** The big-step exponential cut-elimination rule.

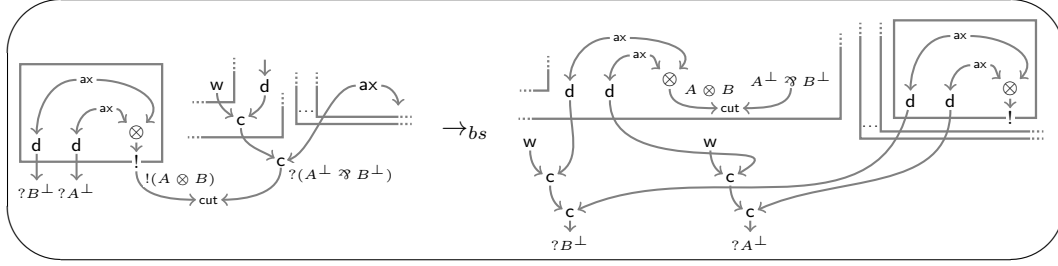
2 Implicit Substitution and its Properties

In the literature the exponential cut-elimination rules sometimes appear in a big-step variant, where a box interacts with a whole tree of $?$ -links in just one shot. This rule was first designed to match substitution in λ -calculus [32, 8], and will be a crucial ingredient of our proof. The rule is usually presented collapsing whole trees of $?$ -formulas in just one node. The additional rules in Figure 6 are an alternative way of realizing such a collapse. We prefer them because the collapsed syntax is slightly *ad-hoc* with respect to box borders.

?-trees. Given an edge e of type $?A$ there is a unique maximal tree of $?$ -links rooted in e (this fact is a consequence of being a proof net, and would be easily seen if we were considering correctness). The internal nodes of such $?$ -trees are c -links and the leaves are $\{ax, w, d\}$ -links. According to the *box-crossing principle* these $?$ -trees may cross box borders. Graphically, this is represented by some horizontal lines crossing the tree, as in the triangle over the cut in the lhs of the rule in Figure 7 (the figure simplifies slightly the shape by grouping the leaves as axioms, weakenings, and derelictions, which is not necessarily the case). If the tree does not cross any box then the horizontal lines are omitted (as in the rhs of Figure 7). The intuition is that a $?$ -tree is the graphical analogous of a variable in a λ -term, whose occurrences are the leaves of the tree.

Big-step exponential rule. Figure 7 shows the big-step exponential rule. The idea is that the rule commutes the box and the $?$ -tree (as it is *natural* from a categorical point of view, see [24]) replacing every axiom leaf with a copy of the box, every dereliction leaf with a copy of the *interior* of the box, contracting the conclusions of the copies with copies of the cut $?$ -tree (now not crossing any box), and adding to these new trees a weakening for every weakening leaf in the old tree. Namely (an example is shown in Figure 8):

1. *Derelictions:* for $r \in \{1, \dots, m\}$ every dereliction l_r of the $?$ -tree is replaced by a copy R^r



■ **Figure 8** Example of substitution.

- of R which is cut with the premise of l_r ;
2. *Axioms*: for $s \in \{1, \dots, o\}$ every axiom i_s of the $?$ -tree is replaced by a copy B^s of the cut box;
 3. *Contractions and weakenings*: for $q \in \{1, \dots, h\}$ the q -th conclusion of every R^r and that one of every B^s are contracted together via a copy of the cut $?$ -tree T_c , and the other n leaves of these copies of T_c are weakenings.
 4. *Box addresses*: the replacements gives to R^r (resp. B^s) the exact box address of l_r (resp. i_s), and each copy of the $?$ -tree T_c and every copied weakening are put out of all the boxes crossed by T_c .

The reader who finds too vague the definition of substitution may prefer to think of boxes as labels on links indicating the box address: then the rule commutes the box and the $?$ -tree replacing for each copy (resp. opened copy) of the box the address prefix corresponding to the cut box with the address of the corresponding leaf of the $?$ -tree, and copies weakenings and contractions in T_x giving them the address of the cut box.

Explicit/implicit box substitution. To employ a handy notation for substitutions, we sometimes associate variable names to some $?$ -conclusions of a proof net, and denote with T_x a $?$ -tree on a conclusion x . Let then P be a proof net of conclusions $\vdash x : ?A, \Gamma$, and Q be a $!$ -box of conclusions $\vdash ?B_1, \dots, ?B_n, !A^\perp$ of interior R . The **explicit substitution** of Q to x in P , noted $P[x/Q]$ is obtained by simply cutting P and Q on x , as in the lhs of Figure 4.b. The **(implicit) substitution** $P\{x/Q\}$, is instead given by the rhs of the same figure. Because of the rule and the equation which bring contractions and weakenings out of boxes, we have to extend the definition of explicit and implicit substitutions to **extended boxes**, *i.e.* to the case where the proof net Q is a box B plus some $?$ -trees eventually having the auxiliary conclusions of B as leaves. In such a case the implicit (resp. explicit) substitution is simply obtained by substituting the box (resp. cutting the box) and adding to the result the same additional $?$ -trees.

Properties. Now, we study the properties of substitution that will be used in the proof of strong normalization. *None of the proofs in this section relies on types.*

The first property is a sort of commutation between promotion and substitution, which is assured by the rules permuting weakening and contractions with box borders. It is simple but important, and it is peculiar of proof nets as it concerns the structure of $?$ -trees. We need some definitions.

A **pointed net** is a net with a distinguished conclusion and $nets_A$ is the set of nets pointed on a conclusion of type A . We denote with $nets_A^?$ the subset of $nets_A$ composed by the proof nets which can be the interior of a box, *i.e.* whose non-distinguished conclusions are $?$ -formulas. For $P \in nets_A^?$ we write $!P$ for the proof net obtained by boxing P on A .

► **Lemma 2** (commutation). *Let $P \in \text{nets}_A^?$, x one of its auxiliary conclusions, and Q a boxed proof net. Then, $!(P\{x/Q\}) \Rightarrow^* (!P)\{x/Q\}$.*

Proof. It is enough to push out of the outer box the contractions and weakenings of the copies of the $?$ -tree produced by substitution (if any). ◀

The second property is full composition, *i.e.* the fact that explicit substitutions can be executed and turned into implicit substitutions.

► **Lemma 3** (full composition). *Let P be a proof net and Q an extended box so that $P\{x/Q\}$ is well-defined. Then, $P\{x/Q\} \rightarrow_{key}^* P\{x/Q\}$, and so $P\{x/Q\} \Rightarrow_{key}^* P\{x/Q\}$.*

Proof. The proof is by induction on the number of links in the $?$ -tree T_x on x (it goes essentially as Lemma 3.1, page 7, in [3]), which is also the exact number of steps in the reduction. If T_x is simply a leaf, *i.e.* a $\{\text{ax}, \text{d}, \text{w}\}$ -link, then the one-step reduct of the explicit substitution is exactly the implicit substitution. Otherwise, the extended box is cut with a contraction and the reduct has two cuts on smaller trees. The *i.h.* concludes the proof. ◀

The third and last property is the IE property. It requires a fine analysis of the commutation between \Rightarrow_{key} and the implicit substitution $P\{x/Q\}$, expressed by the following lemma. Essentially, it states that if $P \Rightarrow_{key} P'$ then $P\{x/Q\} \Rightarrow_{key} P'\{x/Q\}$ (and similarly for Q), but it points out some special cases and some additional information. Its only use is in the proof of the IE property, which follows.

► **Lemma 4** (Substitutivity). *Consider a proof net P and an extended box Q so that $P\{x/Q\}$ is defined.*

1. $P \equiv P'$ implies $P\{x/Q\} \equiv P'\{x/Q\}$, and $Q \equiv Q'$ implies $P\{x/Q\} \equiv P\{x/Q'\}$.
2. $P \rightarrow_{key} P'$ implies $P\{x/Q\} \rightarrow_{key} P'\{x/Q\}$.
3. Let $P \rightarrow_n P'$. If the step acts on T_x and Q has no auxiliary conclusion then $P\{x/Q\} = P'\{x/Q\}$, otherwise $P\{x/Q\} \rightarrow_n^+ P'\{x/Q\}$.
4. Let $P \rightarrow_{pw} P'$. If the step acts on T_x then $P\{x/Q\} = P'\{x/Q\}$, otherwise $P\{x/Q\} \rightarrow_{pw} P'\{x/Q\}$.
5. Let $Q \Rightarrow_{key} Q'$. If the step is strictly contained in the box of Q and every leaf of T_x is a weakening then $P\{x/Q\} \equiv P\{x/Q'\}$, otherwise $P\{x/Q\} \Rightarrow_{key}^+ P\{x/Q'\}$.

Proof. 1. It is enough to prove the statement for the generators of \equiv —whose proof is given by easy verifications—as the result then follows by a straightforward induction.

2-4. The cut c reduced in $P \rightarrow_{key} P'$ clearly exists in $P\{x/Q\}$. For each point the only interesting case is when the reduction of c affects T_x , otherwise substitution and reduction simply commute. 2) A case analysis shows that if $P\{x/Q\} \rightarrow_{key} R$ by reducing the cut corresponding to c then R is equal to $P'\{x/Q\}$ up to the the rules and equations in Figure 6; 3) It is enough to repeat the \rightarrow_n -step on every copy of T_x in $P\{x/Q\}$; if Q has no auxiliary conclusion then the implicit substitution does not copy T_x , that explains the equality; 4) Substitution pushes weakenings and contractions out of boxes, so $P\{x/Q\} = P'\{x/Q\}$; if instead the step does not act on T_x then it commutes with substitution.

5. We show the property using only the rewriting rules, the statement then follows by point 1. Any reduction inside the box of Q has to be repeated for the $m + o$ copies of the interior of the cut box in $P\{x/Q\}$; this means 0 times if T_x has only weakening leaves. In the case of a \rightarrow_{pw} step on the border of the box of Q , it is enough to do some \rightarrow_{pw} steps and then some \rightarrow_n steps, *i.e.* $P\{x/Q\} \rightarrow_{pw}^* \rightarrow_n^* P\{x/Q'\}$. The only remaining case is of a \rightarrow_n step on one of the $?$ -trees extending the box of Q : the step simply commutes with substitution. ◀

We can now prove the IE property for \rightarrow_{key} . The contraction cut-elimination rule forces to prove a generalized n -ary formulation with respect to the one presented in the introduction.

Notations. SN_{key} denotes the set of proof nets which are strongly normalizing with respect to \Rightarrow_{key} . For $R \in SN_{key}$ let $\eta(R)$ be the sum of the lengths of all \Rightarrow_{key} -reductions from R , for $i \leq j$ let $\{\cdot\}_i^j := \{x_i/Q_i\} \dots \{x_j/Q_j\}$ and $[\cdot]_i^j := [x_i/Q_i] \dots [x_j/Q_j]$, and let T_{x_i} be the ? -tree on x_i for $i \in \{1, \dots, n\}$. The **size** $|T|$ of a ? -tree T is the number of $\{\text{ax}, \text{w}, \text{d}, \text{c}\}$ -links in T plus for every weakening l in T the number of boxes crossed by the path from l to the root of T (added to take into account rule \rightarrow_{pw}).

► **Lemma 5** (IE property). *Let P be a net of conclusions $\vdash \Gamma, x_1:?A_1, \dots, x_n:?A_n$ and $Q_1, \dots, Q_n \in SN_{key}$ be extended boxes of type $!A_1^\perp, \dots, !A_n^\perp$, respectively. Then $P\{\cdot\}_1^n \in SN_{key}$ implies $P[\cdot]_1^n \in SN_{key}$.*

Proof. The proof is by induction on the triple, lexicographically ordered (and it is similar to the one for Theorem 4.3, page 13, in [3]):

$$(\eta(P\{\cdot\}_1^n), \sum_{i=1}^n |T_{x_i}|, \sum_{i=1}^n \eta(Q_i))$$

The proof consists in showing that whenever $P[\cdot]_1^n \Rightarrow_{key} R$ then $R \in SN_{key}$. For the reduction cases the measure decreases (so that we can apply the *i.h.*) and for the equivalence cases the measure is invariant, and so it properly lifts to equivalence classes. Cases:

1. *Equivalence in P :* $P[\cdot]_1^n \equiv P'[\cdot]_1^n$ because $P \equiv P'$. Then Lemma 4.1 gives $P\{\cdot\}_1^n \equiv P'\{\cdot\}_1^n$, and so the first component of the measure does not change. The second cannot be altered by \equiv and the third one is not affected.
2. *Equivalence in Q_i :* analogous to the previous case.
3. *\rightarrow_{key} -reduction in P :* if $P[\cdot]_1^n \rightarrow_{key} P'[\cdot]_1^n$ then Lemma 4.2 gives $P\{\cdot\}_1^n \rightarrow_{key} \Rightarrow_{key}^* P'\{\cdot\}_1^n$ and so $\eta(P'\{\cdot\}_1^n) < \eta(P\{\cdot\}_1^n)$. Then, the *i.h.* allows to conclude with $P'[\cdot]_1^n \in SN_{key}$.
4. *\rightarrow_n -reduction in P :* if the step acts on T_{x_i} (for some i) and Q_i has no auxiliary conclusions then by Lemma 4.3 $P\{\cdot\}_1^n = P'\{\cdot\}_1^n$, and so the first component of the measure does not change, but the size of T_{x_i} strictly decreases and so we conclude by the *i.h.*. Otherwise, Lemma 4.3 gives $P\{\cdot\}_1^n \rightarrow_n^+ P'\{\cdot\}_1^n$, and the first component decreases.
5. *\rightarrow_{pw} -reduction in P :* If the step acts on T_{x_i} for some i then by Lemma 4.4 $P\{\cdot\}_1^n = P'\{\cdot\}_1^n$, but the second component decreases. Otherwise, Lemma 4.4 gives $P\{\cdot\}_1^n \rightarrow_{pw} P'\{\cdot\}_1^n$ and the first component decreases.
6. *Reduction in Q_i :* by Lemma 4.5 there are two sub-cases. If every leaf of T_{x_i} is a weakening and the step is contained in the box of Q_i then the lemma gives $P\{\cdot\}_1^{i-1}\{x_i/Q_i\}\{\cdot\}_{i+1}^n = P\{\cdot\}_1^{i-1}\{x_i/Q'_i\}\{\cdot\}_{i+1}^n$, because the implicit substitution erases both Q_i and Q'_i . The second component of the measure does not change either. However, the third component decreases, because $\eta(Q'_i) < \eta(Q_i)$, and we conclude using the *i.h.*. Otherwise, $P\{\cdot\}_1^{i-1}\{x_i/Q_i\}\{\cdot\}_{i+1}^n \rightarrow_{key}^+ P\{\cdot\}_1^{i-1}\{x_i/Q'_i\}\{\cdot\}_{i+1}^n$ and the first component decreases.
7. *Reduction of $[x_i/Q_i]$:* there are two sub-cases:
 - a. *Cut with a contraction:* $P[\cdot]_1^{i-1}[x_i/Q_i][\cdot]_{i+1}^n \rightarrow_{key} P'[\cdot]_1^{i-1}[x'_i/Q_i][x''_i/Q'_i][\cdot]_{i+1}^n$, where P' is the net obtained from P by removing the contraction on x_i , and x'_i and x''_i are the names of the obtained two ? -trees. We can apply the *i.h.*, because $P'\{\cdot\}_1^{i-1}\{x'_i/Q_i\}\{x''_i/Q'_i\}\{\cdot\}_{i+1}^n = P\{\cdot\}_1^{i-1}\{x_i/Q_i\}\{\cdot\}_{i+1}^n$ and the second component decreases, since $|T_{x'_i}| + |T_{x''_i}| < |T_{x_i}|$. Note that the third element of the measure potentially increases, so that the lexicographic order on the measure is necessary (in the previous case of a reduction in P , similarly, the second element may increase).

- b. *Cut with an axiom/dereliction/weakening*: the reduct and the implicit substitution coincide, *i.e.* $P[\cdot]_1^{i-1}[x_i/Q_i][\cdot]_{i+1}^n \rightarrow_{key} P[\cdot]_1^{i-1}\{x_i/Q_i\}[\cdot]_{i+1}^n = P\{x_i/Q_i\}[\cdot]_1^{i-1}[\cdot]_{i+1}^n$. As in the previous sub-case the first component of the measure does not change, while the second decreases (one \cdot -tree less, and they always have positive size), and so we conclude by the *i.h.*. ◀

The IE property also holds for \rightarrow_{com} , because it is known that the system is SN, but the simple proof technique presented here for \Rightarrow_{key} cannot be applied. The problem is that the box commutation rule breaks the inductive invariant. In fact, in the reduction of $[x_i/Q_i]$ one has to consider the case of a commutative step, which would bring Q_i inside P , getting a proof net P' for which $\eta(P'\{\cdot\}_1^{i-1}\{\cdot\}_{i+1}^n)$ is bigger than $\eta(P\{\cdot\}_1^n)$. Then for \rightarrow_{com} a much more involved proof strategy has to be employed, like the labeling technique in [21].

3 Strong Normalization via Reducibility Candidates

Reducibility candidates are a standard construction, that we use following the schema in the literature [14, 13]. What is original here is the proof that every proof net is reducible. The method requires many definitions and notations.

A rewriting relation \rightarrow for proof nets is **substitutive** if 1) multiplicative cuts are reduced only according to the usual cut-elimination rules; 2) it is defined for all cuts; 3) it makes promotion commute with the implicit substitution, *i.e.* $!(P\{x/Q\}) \rightarrow^* (!P)\{x/Q\}$; 4) it satisfies *full composition* and the *IE property*; 5) reduction is stable by subnets and by context closure. In the following \rightarrow denotes a substitutive rewriting relation and every notion is parametrized by \rightarrow . However, to simplify the terminology and the notation *we keep the parametrization implicit as much as possible*.

Notations. If $P \in nets_A$ and $Q \in nets_{A^\perp}$ then $cut(P|Q)$ is the net obtained by cutting P and Q on their distinguished conclusions. We use SN_{\rightarrow} for the set of \rightarrow -strongly normalizing proof nets and we define $SN_A := nets_A \cap SN_{\rightarrow}$. Moreover, $\eta(P)$ here denotes the sum of the length of all \rightarrow -reductions from P , for $P \in SN_{\rightarrow}$.

Duality. Given $S \subseteq nets_A$ the dual set $S^\perp \subseteq nets_{A^\perp}$ contains the nets Q s.t. $cut(Q|P) \in SN_{\rightarrow}$ for every $P \in S$. *Properties of duality:* if $S \neq \emptyset$ then $S^\perp \subseteq SN_{\rightarrow}$; $S \subseteq S^{\perp\perp}$; $S^{\perp\perp\perp} = S^\perp$; if $S \subseteq R$ then $R^\perp \subseteq S^\perp$.

► **Lemma 6 (non-emptiness).** *If $S \subseteq SN_A$ then S^\perp contains the axiom on A^\perp (and A). Consequently, $S^\perp \neq \emptyset$.*

Proof. Let Q be the axiom on A^\perp (and A). We show that $cut(P|Q) \in SN_{\rightarrow}$ for all $P \in S$; it then follows that $Q \in S^\perp$. By induction on $\eta(P)$, showing that any reduct of $cut(P|Q)$ is in SN_{\rightarrow} . Two cases. 1) *Reduction of the introduced cut*: we get $cut(P|Q) \rightarrow P$ and we conclude, since $P \in SN_{\rightarrow}$ by hypothesis. 2) *Reduction of a cut of P* : then $cut(P|Q) \rightarrow cut(P'|Q)$ and we conclude by the *i.h.*. ◀

A **reducibility candidate** is a set of pointed nets $S \subseteq nets_A$ s.t. $S = S^{\perp\perp}$, $S \neq \emptyset$ and $S \subseteq SN_A$. The general property $S^{\perp\perp\perp} = S^\perp$ provides the typical way of building reducibility candidates, consisting in taking the dual of a non-empty set of strongly normalizing nets.

We associate to every type A a reducibility candidate $\llbracket A \rrbracket \subseteq nets_A$, by induction on A :

- *Atomic formula:* $\llbracket X \rrbracket := SN_X$.
- *Tensor:* $\llbracket A \otimes B \rrbracket := \{P \otimes Q \mid P \in \llbracket A \rrbracket, Q \in \llbracket B \rrbracket\}^{\perp\perp}$, where $P \otimes Q$ is the net obtained by adding a tensor link on the distinguished conclusions of P and Q .
- *Par:* $\llbracket A \wp B \rrbracket := \llbracket A^\perp \otimes B^\perp \rrbracket^\perp$.

- *Bang*: $\llbracket !A \rrbracket := \{!P \mid P \in \llbracket A \rrbracket \cap \text{nets}_A^?\}^{\perp\perp}$ (these notations are defined in the paragraph before Lemma 2).
- *Why not*: $\llbracket ?A \rrbracket := \llbracket !A^\perp \rrbracket^\perp$.

► **Lemma 7.** $\llbracket A \rrbracket$ is a reducibility candidate for every MELL formula A .

Proof. By induction on the definition of $\llbracket A \rrbracket$. The base case follows from the more general fact that for whatever A , SN_A is the reducibility candidate given by $\{P\}^\perp$, where P is the axiom on A^\perp . The inductive cases follow easily by the *i.h.* and the properties of duality. ◀

More notations. Given a multiset of formulas $\Gamma = A_1, \dots, A_k$ we use Q^Γ for a multiset of pointed proof nets Q_1, \dots, Q_k s.t. $Q_i \in \text{nets}_{A_i}$ for $i \in \{1, \dots, k\}$, and we use $Q^\Gamma \in \llbracket \Gamma \rrbracket$ if moreover $Q_i \in \llbracket A_i \rrbracket$. If P has conclusions Γ we also write $\text{cut}(P|Q_1, \dots, Q_k)$ or $\text{cut}(P|Q^\Gamma)$ for the net obtained by cutting the conclusion A_i of P with Q_i for every $i \in \{1, \dots, k\}$.

A proof net P of conclusions $\vdash \Gamma$ is **(\rightarrow)reducible** if $\text{cut}(P|Q^\Gamma) \in SN_{\rightarrow}$ for every $Q^\Gamma \in \llbracket \Gamma \rrbracket$. Similarly, a proof net P pointed on A and of conclusions $\vdash A, \Gamma$ is reducible when $\text{cut}(P|Q^\Gamma) \in \llbracket A \rrbracket$ for every $Q^\Gamma \in \llbracket \Gamma \rrbracket$. *1st property of reducibility:* the two formulations of reducibility are related as follows. Let P be a proof net of conclusions $\vdash A, \Gamma$, and let P^A be P but pointed on A . Then, P is reducible iff P^A is reducible. *2nd property of reducibility:* Let P be a proof net of conclusions $\vdash ?A, \Gamma$. By the previous property P is reducible iff $P^{?A} \in \llbracket ?A \rrbracket$. By definition, to show $P^{?A} \in \llbracket ?A \rrbracket$ we have to cut P on $?A$ with a proof net in $\{!P \mid P \in \llbracket A \rrbracket\}$ (note the absence of the double dual), *i.e.* only with respect to nets contained in boxes. Similarly, for \wp -formulas we only have to consider the case in which they are cut with a tensor.

The two properties of reducibility are repeatedly used in the following proof, and applied to many proof nets simultaneously. The novelties of the proof are the treatment of the exponential cases, in particular the case of $!$, and the parametrization with respect to a substitutive relation.

► **Theorem 8.** Let \rightarrow be a substitutive rewriting relation. Every proof net is \rightarrow -reducible.

Proof. Let P be a proof net and π a sequent calculus proof mapping to P . The proof is by induction on π . The base cases:

- **Zeroary mix:** trivial, because the net is empty.
- **Axiom:** P is an axiom of conclusions $\vdash A, A^\perp$. Then we need to show that $\text{cut}(P|Q, R) \in SN_{\rightarrow}$ for $Q \in \llbracket A \rrbracket$ and $R \in \llbracket A^\perp \rrbracket$. By induction on $\eta(Q) + \eta(R)$, showing that whenever $\text{cut}(P|Q, R) \rightarrow S$ then $S \in SN_{\rightarrow}$. If reduction takes place in Q or in R then we conclude by the *i.h.*. Otherwise it is one of the two introduced axiom cuts which is reduced. The axiom is at level 0, so it is rule \rightarrow_{ax} which is applied. In both cases the reduct is $\text{cut}(Q|R)$, which is in SN_{\rightarrow} by the properties of reducibility candidates.

The inductive exponential cases. Suppose that the last rule of π is a:

- **Promotion.** Consider P as a net pointed on $!A$. It writes as $!Q$, for Q of conclusions $\vdash ?B_1, \dots, ?B_n, A$. We need to show that $\text{cut}(!Q|S, R_1, \dots, R_n) \in SN_{\rightarrow}$ where $S \in \llbracket ?A^\perp \rrbracket$ and (by the second property of reducibility) R_i is a box of distinguished conclusion $!B_i^\perp$. Note that $\text{cut}(!Q|S, R_1, \dots, R_n) = \text{cut}(!Q|S)[x_1/R_1] \dots [x_n/R_n]$ (shorted $\text{cut}(!Q|S)[\cdot]_1^n$), if $?B_i$ is named x_i for $i \in \{1, \dots, n\}$. Now, by *i.h.* Q is reducible, which implies $Q[\cdot]_1^n \in \llbracket A \rrbracket$ and so $!(Q[\cdot]_1^n) \in \llbracket !A \rrbracket$. Note that in particular $\text{cut}(!(Q[\cdot]_1^n)|S) \in SN_{\rightarrow}$ holds. By full composition $\text{cut}(!(Q[\cdot]_1^n)|S) \rightarrow^* \text{cut}(!(Q\{\cdot\}_1^n)|S)$ and by commutation $\text{cut}(!(Q\{\cdot\}_1^n)|S) \rightarrow^* \text{cut}(!Q)\{\cdot\}_1^n|S = \text{cut}(!Q|S)\{\cdot\}_1^n$, and so this last proof net is in SN_{\rightarrow} . For $i \in \{1, \dots, n\}$ we have $R_i \in SN_{\rightarrow}$, and so we can apply the IE property and get $(\text{cut}(!Q|S))[\cdot]_1^n \in SN_{\rightarrow}$, which concludes the proof.

- **Weakening**, *i.e.* P is given by a proof net Q of conclusions $\vdash \Gamma$ plus a weakening l of conclusion $?A$, that we name x . By *i.h.* $cut(Q|R^\Gamma) \in SN_{\rightarrow}$ for every $R^\Gamma \in \llbracket \Gamma \rrbracket$. We need to show that $cut(P|R^\Gamma) \in \llbracket ?A \rrbracket$, *i.e.* if we name x the conclusion of the weakening we need to show that $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$ for $S \in \llbracket A^\perp \rrbracket \cap nets_{A^\perp}^?$ (remark the use of the 2nd property of reducibility). We get $cut(P|R^\Gamma)[x/!S] \rightarrow^* cut(P|R^\Gamma)\{x/!S\}$ by full composition, where $cut(P|R^\Gamma)\{x/!S\}$ is $cut(Q|R^\Gamma)$ plus a weakening at level 0 for every non-pointed conclusion of S . The proof net $cut(Q|R^\Gamma)$ is in SN_{key} by *i.h.*, which implies $cut(P|R^\Gamma)\{x/!S\} \in SN_{\rightarrow}$. We also know that S , and thus $!S$, is in SN_{\rightarrow} . The IE property then gives $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$, *i.e.* $cut(P|R^\Gamma) \in \llbracket ?A \rrbracket$.
- **Dereliction**, *i.e.* P is given by a proof net Q of conclusions $\vdash A, \Gamma$ plus a dereliction on A , whose conclusion $?A$ we name x . We need to show that $cut(P|R^\Gamma) \in \llbracket ?A \rrbracket$, *i.e.* that $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$ for $S \in \llbracket A \rrbracket \cap nets_{A^\perp}^?$. By full composition $cut(P|R^\Gamma)[x/!S] \rightarrow^* cut(P|R^\Gamma)\{x/!S\} = cut(Q|R^\Gamma, S)$. By *i.h.* this last net is in SN_{\rightarrow} and by hypothesis $S \in SN_{\rightarrow}$, so we can apply the IE property and get $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$.
- **Contraction**, *i.e.* P is given by a proof net Q of conclusions $\vdash ?A, ?A, \Gamma$ plus a contraction on the two occurrences of $?A$. Let us call these occurrences y and z . The *i.h.* is $cut(Q|R_1^\Gamma)[y/!R_1][z/!R_2] \in SN_{\rightarrow}$ for every $R_1, R_2 \in \llbracket A \rrbracket \cap nets_{A^\perp}^?$ and for every $R^\Gamma \in \llbracket \Gamma \rrbracket$. We need to show that $cut(P|R^\Gamma) \in \llbracket ?A \rrbracket$, which is equivalent to $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$ for every $S \in \llbracket A^\perp \rrbracket \cap nets_{A^\perp}^?$, if we name x the conclusion of the contraction. We get $cut(P|R^\Gamma)[x/!S] \rightarrow^* cut(P|R^\Gamma)\{x/!S\}$ by full composition. Note that $cut(P|R^\Gamma)\{x/!S\}$ is equal to $cut(Q|R^\Gamma)\{y/!S\}\{z/!S\}$ and that this net is in SN_{\rightarrow} , because the proof net $cut(Q|R^\Gamma)[y/!S][z/!S]$ (which is SN by *i.h.*) reduces to it by full composition. We also know that S , and thus $!S$, is in SN_{\rightarrow} . Then we can apply the IE property, getting $cut(P|R^\Gamma)[x/!S] \in SN_{\rightarrow}$.

We only show one non-exponential case (for \otimes) as the others (\wp , cut, and binary mix) are simpler and similar to what already appeared in the literature. If the last rule of π is a tensor then P writes as $Q_1 \otimes Q_2$. Let $\vdash \Gamma, \Delta, A \otimes B, \vdash \Gamma, A$, and $\vdash \Delta, B$ be the conclusions of P , Q_1 , and Q_2 , respectively. We need to show that $cut(Q_1 \otimes Q_2|R^\Gamma, R^\Delta) \in \llbracket A \otimes B \rrbracket$. But $cut(Q_1 \otimes Q_2|R^\Gamma, R^\Delta) = cut(Q_1|R^\Gamma) \otimes cut(Q_2|R^\Delta)$, and by *i.h.* $cut(Q_1|R^\Gamma) \in \llbracket A \rrbracket$ and $cut(Q_2|R^\Delta) \in \llbracket B \rrbracket$. Then by definition of $\llbracket A \otimes B \rrbracket$ we get $cut(Q_1 \otimes Q_2|R^\Gamma, R^\Delta) \in \llbracket A \otimes B \rrbracket$ ◀

► **Corollary 9.** *Every proof net is in SN_{\rightarrow} , and so in SN_{key} .*

Proof. Let P be a proof net. By the previous theorem P is reducible. If $\vdash A_1, \dots, A_n$ are its conclusions then $P \in \llbracket A_i \rrbracket$ for $i \in \{1, \dots, n\}$. By Lemma 6 $\llbracket A_i^\perp \rrbracket$ contains the axiom on A_i^\perp . Then cutting P with axioms on $A_1^\perp, \dots, A_n^\perp$ we get a net Q s.t. $Q \in SN_{\rightarrow}$ and $Q \rightarrow^* P$. So, $P \in SN_{\rightarrow}$. Section 2 shows that \rightarrow_{key} is substitutive, and so $P \in SN_{key}$. ◀

4 Variations on a Theme

The multiplicative units are in fact already inside MELL: 1 can be coded with the proof of $!(A \wp A^\perp)$ obtained by an axiom on A , and \perp can be coded by a weakening on $?(A^\perp \otimes A)$.

Second order quantifiers. the extension of the proof to second order can be done as in [14], it is standard, not particularly interesting, and it obfuscates the structure of the proof by forcing to deal with substitution on type variables. Once the reducibility candidates technique is adopted the treatment of second order is smooth.

Additives. If they are represented using slices (as in [22], for instance) their treatment is trivial, because the additive cut-elimination rules strictly decrease the size of the proof net. Replacing slices by additive boxes may seem trickier, as it requires some commutation

rules. But in [28] it is shown that SN for the additive slices implies SN for the additive boxes (Proposition 5.6, page 56). Thus, in either cases we catch strong normalization for full linear logic. Our proof technique does not rely on confluence, which is why the extension is so simple. This is an impressive simplification of the involved proof by Tortora de Falco and Pagani in [28], because they use a conservation theorem, that requires to show a delicate form of confluence, and confluence is notoriously problematic with the additives.

Confluence. In the case without exponential axioms, confluence (and even Church-Rosser modulo \equiv) can be shown along the lines of the study in [1]. Exponential axioms introduce a new critical pair that is unusual and difficult to study. We claim that confluence still holds. Preliminary results suggest that the notion of implicit substitution can be used to obtain a proof of confluence by projection, following the usual argument for explicit substitutions.

Conclusions

We gave a presentation of MELL proof nets without any commutative rule, and showed a proof of strong normalization which is simple, informative, modular, and does not rely on confluence. The cut-elimination theorem for proof nets is now not just proved, but also understood and made accessible to a wider audience.

Acknowledgements

To Delia Kesner, who taught me the art of strong normalization proofs, and with whom I had uncountable discussions on commutative rules, normalization, and proof nets. And to the anonymous reviewers, who pointed out various inaccuracies. I also had fruitful discussions with Michele Pagani, Lorenzo Tortora de Falco, Paolo Tranquilli, Olivier Laurent, Stefano Guerrini, Stéphane Gimenez, and Frank Pfenning. This work was partially supported by the Qatar National Research Fund under grant NPRP 09-1107-1-168.

References

- 1 Beniamino Accattoli. Compressing polarized boxes. Accepted at LICS, 2013.
- 2 Beniamino Accattoli and Stefano Guerrini. Jumping boxes. In *CSL*, pages 55–70, 2009.
- 3 Beniamino Accattoli and Delia Kesner. Preservation of strong normalisation modulo permutations for the structural λ -calculus. *Logical Methods in Computer Science*, 8(1), 2012.
- 4 Andrea Asperti. Causal dependencies in multiplicative linear logic with mix. *Mathematical Structures in Computer Science*, 5(3):351–380, 1995.
- 5 Andrea Asperti and Stefano Guerrini. *The Optimal Implementation of Functional Programming Languages*. Cambridge University Press, 1998.
- 6 Eduardo Bonelli. Perpetuality in a named lambda calculus with explicit substitutions. *Mathematical Structures in Computer Science*, 11(1):47–90, 2001.
- 7 Vincent Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ -calcul)*. Phd thesis, Université Paris 7, 1990.
- 8 Vincent Danos and Laurent Regnier. Proof-nets and the Hilbert space. In *Advances in Linear Logic*, pages 307–328. Cambridge University Press, 1995.
- 9 Roberto Di Cosmo and Stefano Guerrini. Strong normalization of proof nets modulo structural congruences. In *RTA*, pages 75–89, 1999.
- 10 Roberto Di Cosmo, Delia Kesner, and Emmanuel Polonovski. Proof nets and explicit substitutions. *Math. Str. in Comput. Sci.*, 13(3):409–450, 2003.
- 11 Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2):273–285, 1994.

- 12 R. O. Gandy. Proofs of strong normalization. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 457–477. Academic Press Limited, 1980.
- 13 Stéphane Gimenez. Realizability proof for normalization of full differential linear logic. In *TLCA*, pages 107–122, 2011.
- 14 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- 15 Jean-Yves Girard. *Proof theory and logical complexity*. Studies in proof theory. Bibliopolis, 1987.
- 16 Jean-Yves Girard. Light linear logic. *Inf. Comput.*, 143(2):175–204, 1998.
- 17 Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, New York, NY, USA, 1989.
- 18 Dominic J. D. Hughes and Rob J. van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. *ACM Trans. Comput. Log.*, 6(4):784–842, 2005.
- 19 Jean-Baptiste Joinet. *Étude de la normalisation du calcul des séquents*. Phd thesis, Université Paris 7, 1977.
- 20 Delia Kesner. The theory of calculi with explicit substitutions revisited. In *CSL*, pages 238–252, 2007.
- 21 Delia Kesner. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science*, 5(3), 2009.
- 22 Olivier Laurent and Lorenzo Tortora de Falco. Slicing polarized additive normalization. In *Linear Logic in Computer Science*, volume 316 of *London Mathematical Society Lecture Note Series*, pages 247–282. Cambridge University Press, November 2004.
- 23 Ralph Loader. Normalisation by calculation. Unpublished note, available at <http://homepages.ihug.co.nz/~suckfish/papers/normal.pdf>, 1995.
- 24 Paul-André Melliès. Functorial boxes in string diagrams. In *CSL*, pages 1–30, 2006.
- 25 Robert Pieter Nederpelt. *Strong Normalization for a Typed Lambda Calculus with Lambda Structured Types*. Ph.D. thesis, Technische Hogeschool Eindhoven, 1973.
- 26 Mitsuhiro Okada. Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. *Theor. Comput. Sci.*, 227(1-2):333–396, 1999.
- 27 Michele Pagani. The cut-elimination theorem for differential nets with promotion. In *TLCA*, pages 219–233, 2009.
- 28 Michele Pagani and Lorenzo Tortora de Falco. Strong normalization property for second order linear logic. *Theor. Comput. Sci.*, 411(2):410–444, 2010.
- 29 Michele Pagani and Paolo Tranquilli. The conservation theorem for differential nets. Accepted for publication in *Math. Str. in Comput. Sci.*, 2009.
- 30 Frank Pfenning. Structural cut elimination in linear logic. Technical Report CMU-CS-94-222, Carnegie Mellon University, 1994.
- 31 Frank Pfenning. Structural cut elimination. In *LICS*, pages 156–166, 1995.
- 32 Laurent Regnier. *Lambda-calcul et réseaux*. PhD thesis, Univ. Paris VII, 1992.
- 33 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
- 34 D. T. van Daalen. *The language theory of automath*. Phd thesis, Technische Hogeschool Eindhoven, 1977.
- 35 Femke van Raamsdonk. *Confluence and Normalization for Higher-Order Rewriting*. PhD thesis, Amsterdam Univ., Netherlands, 1996.
- 36 Femke van Raamsdonk and Paula Severi. On normalisation. Technical Report CS-R9545, Centrum Wiskunde & Informatica, 1995.
- 37 Femke van Raamsdonk, Paula Severi, Morten Heine Sørensen, and Hongwei Xi. Perpetual reductions in lambda-calculus. *Inf. Comput.*, 149(2):173–225, 1999.