Under consideration for publication in Math. Struct. in Comp. Science

Deep Inference and Expansion Trees for Second Order Multiplicative Linear Logic

Lutz Straßburger

Inria

Received 7 September 2018

In this paper we introduce the notion of expansion tree for linear logic. As in Miller's original work, we have a shallow reading of an expansion tree that corresponds to the conclusion of the proof, and a deep reading which is a formula that can be proved by propositional rules. We focus our attention to MLL2, and we also present a deep inference system for that logic. This allows us to give a syntactic proof to a version of Herbrand's theorem.

1. Introduction

Expansion trees (Miller, 1987) have been introduced by Miller to generalize Herbrand's theorem to higher order logic. In principle, an expansion tree is a data structure for proofs that carries the information of two formulas. The *shallow* formula is the conclusion of the proof, and the *deep* formula is a propositional tautology for which the information about the proof has to be provided by other means.

This possible separation of the "quantifier part" and the "propositional part" in a proof is a unique property of classical logic. For intuitionistic logic, for example, only a limited form of Herbrand's theorem can be obtained (Lyaletski and Konev, 2006). The question we would like to address in this paper is whether some form of Herbrand's theorem can be achieved for linear logic.

For simplicity, we concentrate in this paper on second order multiplicative linear logic (MLL2) because the notion of proof in its propositional fragment (MLL) is thoroughly understood: on the deductive level via rule permutations in the sequent calculus (Lafont, 1995) and the calculus of structures (Straßburger, 2003), on the combinatoric level via proof nets (Girard, 1987), and on the algebraic level via star-autonomous categories (Lafont, 1988; Lamarche and Straßburger, 2006), and the first order version has a rather simple proof theory (Bellin and van de Wiele, 1995).

There are two main contributions in this paper:

— First, we will present a data structure for linear logic proofs that carries the information of two formulas: a *shallow* formula that is the conclusion of the proof, and the *deep* formula for which another proof data structure will be provided that is essentially an ordinary MLL proof net. Due to the similarities to Miller's work (Miller, 1987), we will call our data structure *expansion tree*. Since we will also consider the multiplicative units, we follow the work in (Straßburger and Lamarche, 2004; Lamarche and Straßburger, 2006) to present a notion of proof graph, that can be plugged on top of our expansion tree and that will cover the

$$\mathsf{id}\,\frac{}{\vdash a^\perp, a} \qquad \perp\,\frac{\vdash \Gamma}{\vdash \perp, \Gamma} \qquad 1\,\frac{}{\vdash 1} \qquad \mathsf{exch}\,\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}$$

$$\parr\,\frac{\vdash A, B, \Gamma}{\vdash [A \parr B], \Gamma} \qquad \otimes\,\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, (A \otimes B), \Delta} \qquad \exists\,\frac{\vdash A\langle a\backslash B\rangle, \Gamma}{\vdash \exists a.A, \Gamma} \qquad \forall\,\frac{\vdash A, \Gamma}{\vdash \forall a.A, \Gamma}\ \text{a not free in Γ}$$

Fig. 1. Sequent calculus system for MLL2

propositional part of an MLL2 proof. In order to make cut elimination work, we need to impose an equivalence relation on these proof graphs. This is a consequence of the PSPACE-completeness of proof equivalence in MLL (Heijltjes and Houston, 2014).

— Our second contribution will be a deductive proof system for MLL2 in the calculus of structures (Guglielmi and Straßburger, 2001; Guglielmi, 2007), making extensive use of deep inference features. This allows us to achieve the same decomposition of a proof into a "quantifier part" and a "propositional part", as it happens with the expansion trees and the proof graphs. This relation will be made precise via a correspondence theorem.

The paper is organized as follows: We will first recall the presentation of MLL2 in the sequent calculus (Section 2) and then give its presentation in the calculus of structures (Section 3). We also show the relation between the sequent calculus system, that we call $\mathsf{MLL2}_{\mathsf{Seq}}$ and the deep inference system that we call $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$. Then, in Section 4, we introduce our expansion trees and also show their relation to the deep inference system. This is followed by the introduction of proof graphs in Section 5. In Section 6, we explore the relation between proof graphs and the calculus of structures, i.e., we show how to translate between the two. Finally, in Section 7 we show cut elimination for our proof graphs with expansion trees.

Some of the results of this paper have already been published at the TLCA 2009 conference (Straßburger, 2009). The main additions here are (1) full proofs of all results, (2) the presentation of cut elimination, and (3) an improved presentation that clearly separates the expansion trees from the propositional part. Further technical details can be found in (Straßburger, 2017).

2. MLL2 in the sequent calculus

Let us first recall the logic MLL2 by giving its presentation in the sequent calculus, by providing a grammar for well-formed formulas and sequents, together with a set of (sequent style) inference rules. Then the theorems of the logic are defined to be those formulas that are derivable via the rules. For MLL2 the set $\mathscr{F}$ of *formulas* is generated by the grammar

$$\mathscr{F} \quad ::= \quad \perp \mid 1 \mid \mathscr{A} \mid \mathscr{A}^\perp \mid [\mathscr{F} \parr \mathscr{F}] \mid (\mathscr{F} \otimes \mathscr{F}) \mid \forall\mathscr{A}.\mathscr{F} \mid \exists\mathscr{A}.\mathscr{F}$$

where $\mathscr{A} = \{a, b, c, \ldots\}$ is a countable set of *propositional variables*. Formulas are denoted by capital Latin letters $(A, B, C, \ldots)$. Linear negation $(-)^\perp$ is defined for all formulas by the usual De Morgan laws:

$$\begin{array}{llll} \perp^\perp = 1 & a^\perp = a^\perp & [A \parr B]^\perp = (A^\perp \otimes B^\perp) & (\exists a.A)^\perp = \forall a.A^\perp \\ 1^\perp = \perp & a^{\perp\perp} = a & (A \otimes B)^\perp = [A^\perp \parr B^\perp] & (\forall a.A)^\perp = \exists a.A^\perp \end{array}$$

An *atom* is a propositional variable a or its dual $a^\perp$. *Sequents* are finite lists of formulas, separated by comma, and are denoted by capital Greek letters $(\Gamma, \Delta, \ldots)$. The notions of *free* and *bound*

$$\mathsf{ai}\!\downarrow \frac{S\{1\}}{S[a^\perp \mathbin{⅋} a]} \qquad \bot\!\downarrow \frac{S\{A\}}{S[\bot \mathbin{⅋} A]} \qquad 1\!\downarrow \frac{S\{A\}}{S(1 \otimes A)} \qquad \mathsf{e}\!\downarrow \frac{S\{1\}}{S\{\forall a.1\}}$$

$$\alpha\!\downarrow \frac{S[[A \mathbin{⅋} B] \mathbin{⅋} C]}{S[A \mathbin{⅋} [B \mathbin{⅋} C]]} \qquad \sigma\!\downarrow \frac{S[A \mathbin{⅋} B]}{S[B \mathbin{⅋} A]} \qquad \mathsf{ls} \frac{S([A \mathbin{⅋} B] \otimes C)}{S[A \mathbin{⅋} (B \otimes C)]} \qquad \mathsf{rs} \frac{S(A \otimes [B \mathbin{⅋} C])}{S[(A \otimes B) \mathbin{⅋} C]}$$

$$\mathsf{u}\!\downarrow \frac{S\{\forall a.[A \mathbin{⅋} B]\}}{S[\forall a.A \mathbin{⅋} \exists a.B]} \qquad \mathsf{n}\!\downarrow \frac{S\{A\langle a \backslash B\rangle\}}{S\{\exists a.A\}} \qquad \mathsf{f}\!\downarrow \frac{S\{\exists a.A\}}{S\{A\}} \; \text{a not free in A.}$$

Fig. 2. Deep inference system for MLL2

variable are defined in the usual way, and we can always rename bound variables. In view of the later parts of the paper, and in order to avoid changing syntax all the time, we use the following syntactic conventions:

(i) We always put parentheses around binary connectives. For better readability we use $[\ldots]$ for $⅋$ and $(\ldots)$ for $\otimes$.
(ii) We omit parentheses if they are superfluous under the assumption that $⅋$ and $\otimes$ associate to the left, e.g., we write $[A \mathbin{⅋} B \mathbin{⅋} C \mathbin{⅋} D]$ to abbreviate $[[[A \mathbin{⅋} B] \mathbin{⅋} C] \mathbin{⅋} D]$.
(iii) The scope of a quantifier ends at the earliest possible place (and not at the latest possible place as usual). This helps saving unnecessary parentheses. For example, in $[\forall a.(a \otimes b) \mathbin{⅋} \exists c.c \mathbin{⅋} a]$, the scope of $\forall a$ is $(a \otimes b)$, and the scope of $\exists c$ is just c. In particular, the a at the end is free.

The inference rules for MLL2 are shown in Figure 1. In the following, we will call this system $\mathsf{MLL2}_{\mathsf{Seq}}$. As shown in (Girard, 1987), it has the cut elimination property:

Theorem 2.1. The cut rule $\mathsf{cut}\,\dfrac{\vdash \Gamma, A \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}$ is admissible for $\mathsf{MLL2}_{\mathsf{Seq}}$.

3. MLL2 in the calculus of structures

We now present a deductive system for MLL2 based on deep inference. We use the calculus of structures, in which the distinction between formulas and sequents disappears. This is the reason for the syntactic conventions introduced above.[†]

The inference rules now work directly (as rewriting rules) on the formulas. The system for MLL2 is shown in Figure 2. There, $S\{\ \}$ stands for an arbitrary formula context.[‡] We omit the braces if the structural parentheses fill the hole. E.g., $S[A \mathbin{⅋} B]$ abbreviates $S\{[A \mathbin{⅋} B]\}$. The system in Figure 2 is called $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$. We use the down-arrow in the name to emphasize that we consider here only the so-called *down fragment* of the system, which corresponds to the cut-free

[†] In the literature on deep inference, e.g., (Brünnler and Tiu, 2001; Guglielmi, 2007), the formula $(a \otimes [b \mathbin{⅋} (a^\perp \otimes c)])$ would be written as $(a, [b, (a^\perp, c)])$, while without our convention mentioned in the previous section, it would be written as $a \otimes (b \mathbin{⅋} (a^\perp \otimes c))$. Our syntactic convention can therefore be seen as an attempt to please both communities. In particular, note that the motivation for the syntactic convention (iii) above is the collapse of the $⅋$ on the formula level and the comma on the sequent level, e.g., $[\forall a.(a \otimes b) \mathbin{⅋} \exists c.c \mathbin{⅋} a]$ is the same as $[\forall a.(a, b), \exists c.c, a]$.

[‡] More precisely, an arbitrary *positive* formula context, which means that the context-hole most not occur inside the scope of an odd number of negations. However, since we only have formulas in negation normal form, there is no need for that additional condition.

system in the sequent calculus.[§] Note that the $\forall$-rule of $\mathsf{MLL2}_{\mathsf{Seq}}$ is in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ decomposed into three pieces, namely, $\mathsf{e}\downarrow$, $\mathsf{u}\downarrow$, and $\mathsf{f}\downarrow$. In $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$, we also need an explicit rule for associativity which is in the sequent calculus "built in". The other rules are almost the same as in the sequent calculus. In particular, the relation between the $\otimes$-rule and the rules ls and rs (called *switch*) has already in detail been investigated by several authors (Retoré, 1993; Blute et al., 1996; Devarajan et al., 1999; Guglielmi, 2007). A *derivation* $\mathscr{D}$ in the system $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ is denoted by

$$\begin{array}{c} A \\ \mathsf{MLL2}_{\mathsf{DI}\downarrow} \Big\Vert \mathscr{D} \\ B \end{array}$$

and is simply a rewriting path from A to B using the inference rules in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$. We say A is the *premise* and B the *conclusion* of $\mathscr{D}$. A *proof* in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ is a derivation whose premise is 1. The following theorem ensures that $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ is indeed a deductive system for $\mathsf{MLL2}$.

Theorem 3.1. Let $A_1, \ldots, A_n$ be arbitrary $\mathsf{MLL2}$ formulas. For every proof of $\vdash A_1, \ldots, A_n$ in $\mathsf{MLL2}_{\mathsf{Seq}}$, there is a proof of $[A_1 \parr \cdots \parr A_n]$ in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$, and vice versa.

Proof. We proceed by structural induction on the sequent proof to construct the deep inference proof. The only non-trivial cases are the rules for $\otimes$ and $\forall$, for which we can produce the derivations

$$\begin{array}{c} 1 \\ \mathsf{MLL2}_{\mathsf{DI}\downarrow} \Big\Vert \mathscr{D}_2 \\ 1\downarrow \dfrac{[B \parr \Delta]}{[(1 \otimes B) \parr \Delta]} \\ \mathsf{MLL2}_{\mathsf{DI}\downarrow} \Big\Vert \mathscr{D}_1 \\ \mathsf{ls}\, \dfrac{[([\Gamma \parr A] \otimes B) \parr \Delta]}{[\Gamma \parr (A \otimes B) \parr \Delta]} \end{array} \quad \text{and} \quad \begin{array}{c} \mathsf{e}\downarrow \dfrac{1}{\forall a.1} \\ \mathsf{MLL2}_{\mathsf{DI}\downarrow} \Big\Vert \mathscr{D} \\ \mathsf{u}\downarrow \dfrac{\forall a.[A \parr \Gamma]}{[\forall a.A \parr \exists a.\Gamma]} \\ \mathsf{f}\downarrow \dfrac{}{[\forall a.A \parr \Gamma]} \end{array}$$

where $\mathscr{D}_2$, $\mathscr{D}_1$, and $\mathscr{D}$ exist by induction hypothesis. Conversely, for translating a $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ proof $\mathscr{D}$ into the sequent calculus, we proceed by induction on the length of $\mathscr{D}$. We then translate

$$\begin{array}{c} 1 \\ \mathsf{MLL2}_{\mathsf{DI}\downarrow} \Big\Vert \mathscr{D}' \\ \rho\, \dfrac{A}{B} \end{array} \quad \text{into} \quad \mathsf{cut}\, \dfrac{\begin{array}{c}\triangledown\, \mathscr{D}_1 \\ \vdash A\end{array} \qquad \begin{array}{c}\triangledown\, \mathscr{D}_2 \\ \vdash A^{\perp}, B\end{array}}{\vdash B}$$

where $\mathscr{D}_1$ exists by induction hypothesis and $\mathscr{D}_2$ exists because every rule ρ of $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ is a valid implication of $\mathsf{MLL2}$. Finally, we apply cut elimination (Theorem 2.1). $\square$

Remark 3.2. Later in this paper we will introduce methods that will allow us to translate cut-free proofs from deep inference to the sequent calculus without introducing cuts.

[§] The *up fragment* (which corresponds to the cut in the sequent calculus) is obtained by dualizing the rules in the down fragment, i.e., by negating and exchanging premise and conclusion. See, e.g., (Straßburger, 2003; Brünnler, 2003; Brünnler and Tiu, 2001; Guglielmi and Straßburger, 2001; Chaudhuri et al., 2011) for details. Note that here we do not have associativity and commutativity of $\parr$ and $\otimes$ as congruence, but as explicit inference rules. For $\parr$ they belong to the down fragment and for $\otimes$ to the up fragment.

As for $\mathsf{MLL2_{Seq}}$, we also have for $\mathsf{MLL2_{DI\downarrow}}$ the cut elimination property, which can be stated as follows:

Theorem 3.3. The cut rule $\mathsf{i}\!\uparrow \dfrac{S(A \otimes A^{\perp})}{S\{\bot\}}$ is admissible for $\mathsf{MLL2_{DI\downarrow}}$.

Proof. Given a proof in $\mathsf{MLL2_{DI\downarrow}} \cup \{\mathsf{i}\!\uparrow\}$, we translate it into $\mathsf{MLL2_{Seq}}$ as done in the proof of Theorem 3.1, eliminate the cut (Theorem 2.1), and translate the result back into $\mathsf{MLL2_{DI\downarrow}}$. □

We could also give a direct proof of Theorem 3.3, inside the calculus of structures, without referring to the sequent calculus, by using a combination of the techniques of *decomposition* and *splitting* (Straßburger, 2003; Brünnler, 2003; Guglielmi, 2007; Straßburger and Guglielmi, 2011; Guglielmi and Straßburger, 2011; Tubella, 2016). However, presenting all the details would go beyond the scope of this paper. We show here only the "one-sided" version of the *decomposition theorem* for $\mathsf{MLL2_{DI\downarrow}}$, which can be seen as a version of Herbrand's theorem for MLL2, and which has no counterpart in the sequent calculus.

Theorem 3.4.

Every derivation $\begin{array}{c} 1 \\ \mathsf{MLL2_{DI\downarrow}} \Big\Vert \mathscr{D} \\ C \end{array}$ can be transformed into $\begin{array}{c} 1 \\ \{\mathsf{ai}\!\downarrow, \bot\!\downarrow, 1\!\downarrow, \mathsf{e}\!\downarrow\} \Big\Vert \mathscr{D}_1 \\ A \\ \{\alpha\!\downarrow, \sigma\!\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\!\downarrow\} \Big\Vert \mathscr{D}_2 \\ B \\ \{\mathsf{n}\!\downarrow, \mathsf{f}\!\downarrow\} \Big\Vert \mathscr{D}_3 \\ C \end{array}$.

The proof of that theorem is essentially a series of rule permutations. In the middle of that process the following inference rule is created:

$$\mathsf{v}\!\downarrow \frac{S\{\exists a.[A \parr B]\}}{S[\exists a.A \parr \exists a.B]} \tag{1}$$

And we need to show that this rule is admissible. This is done by proving a slightly more general statement.

Lemma 3.5. (i) Whenever there are derivations $\begin{array}{c} 1 \\ \mathsf{MLL2_{DI\downarrow}} \Big\Vert \mathscr{D}_1 \\ S\{\exists a.C\} \end{array}$ and $\begin{array}{c} C \\ \mathsf{MLL2_{DI\downarrow}} \Big\Vert \mathscr{D}_2 \\ A \parr B \end{array}$ then there is a derivation $\begin{array}{c} 1 \\ \mathsf{MLL2_{DI\downarrow}} \Big\Vert \mathscr{D}_3 \\ S[\exists a.A \parr \exists a.B] \end{array}$, and (ii) the rule $\mathsf{v}\!\downarrow$ is admissible for $\mathsf{MLL2_{DI\downarrow}}$.

Proof. For proving the first statement, we proceed by induction on the length of $\mathscr{D}_1$, and make a case analysis of the bottommost rule instance in $\mathscr{D}_1$. If this rule instance acts inside C or inside the context $S\{\ \}$, then we can apply immediately the induction hypothesis. The only interesting cases are when this rule removes the $\exists$. There are two cases: $\mathsf{u}\!\downarrow$ and $\mathsf{n}\!\downarrow$. In the case of $\mathsf{u}\!\downarrow$, the derivation $\mathscr{D}_1$ has the shape on the left below, and we can construct $\mathscr{D}_3$ as shown on the right below:

$$
\begin{array}{c}
1 \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}'_1 \\
\mathsf{u}\!\downarrow \dfrac{S'\{\forall a.[D \parr C]\}}{S'[\forall a.D \parr \exists a.C]}
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
1 \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}'_1 \\
S'\{\forall a.[D \parr C]\} \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}_2 \\
\alpha\!\downarrow,\sigma\!\downarrow \dfrac{S'\{\forall a.[D \parr [A \parr B]]\}}{\mathsf{u}\!\downarrow \dfrac{S'\{\forall a.[[D \parr A] \parr B]\}}{\mathsf{u}\!\downarrow \dfrac{S'[\forall a.[D \parr A] \parr \exists a.B]}{\alpha\!\downarrow \dfrac{S'[[\forall a.D \parr \exists a.A] \parr \exists a.B]}{S'[\forall a.D \parr [\exists a.A \parr \exists a.B]]}}}}
\end{array}
$$

and in the case of n↓, the situation is as follows:

$$
\begin{array}{c}
1 \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}'_1 \\
\mathsf{n}\!\downarrow \dfrac{S\{C\langle a \backslash D\rangle\}}{S\{\exists a.C\}}
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
1 \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}'_1 \\
S\{C\langle a \backslash D\rangle\} \\
\mathsf{MLL2_{DI\downarrow}} \Big\| \mathscr{D}_2 \\
= \dfrac{S\{[A \parr B]\langle a \backslash D\rangle\}}{\mathsf{n}\!\downarrow \dfrac{S[A\langle a \backslash D\rangle \parr B\langle a \backslash D\rangle]}{\mathsf{n}\!\downarrow \dfrac{S[A\langle a \backslash D\rangle \parr \exists a.B]}{S[\exists a.A \parr \exists a.B]}}}
\end{array}
\quad .
$$

The second statement of the lemma is just a special case of the first, where $\mathscr{D}_2$ is the identity. □

Proof of Theorem 3.4 The construction is done in two phases. First, we permute all instances of ai↓, ⊥↓, 1↓, e↓ to the top of the derivation. For ai↓ and e↓ this is trivial, because all steps are similar to the following:

$$
\sigma\!\downarrow \dfrac{S[A \parr B\{1\}]}{\mathsf{e}\!\downarrow \dfrac{S[B\{1\} \parr A]}{S[B\{\forall a.1\} \parr A]}}
\quad \to \quad
\mathsf{e}\!\downarrow \dfrac{S[A \parr B\{1\}]}{\sigma\!\downarrow \dfrac{S[A \parr B\{\forall a.1\}]}{S[B\{\forall a.1\} \parr A]}}
$$

For ⊥↓ and 1↓ there are some more cases to inspect. We show here only one because all others are similar:

$$
\mathsf{u}\!\downarrow \dfrac{S\{\forall a.[A \parr B]\}}{1\!\downarrow \dfrac{S[\forall a.A \parr \exists a.B]}{S[(1 \otimes \forall a.A) \parr \exists a.B]}}
\quad \to \quad
1\!\downarrow \dfrac{S\{\forall a.[A \parr B]\}}{\mathsf{u}\!\downarrow \dfrac{S(1 \otimes \forall a.[A \parr B])}{\mathsf{rs} \dfrac{S(1 \otimes [\forall a.A \parr \exists a.B])}{S[(1 \otimes \forall a.A) \parr \exists a.B]}}}
$$

Here, in order to permute the 1↓ above the u↓, we need an additional instance of rs (and possibly two instances of σ↓). The situation is analogous if we permute the 1↓ over ls, rs, or α↓ (or ai↓ or ⊥↓, but this is not needed for this theorem). When permuting ⊥↓ up (instead of 1↓), then we need α↓ (and σ↓) instead of rs. For a detailed analysis of this kind of permutation arguments, the reader is referred to (Straßburger, 2003).

In the second phase of the decomposition, all instances of n↓ and f↓ are permuted down to the bottom of the derivation. For the rule n↓ this is trivial since no rule can interfere (except for f↓, which is also permuted down). For permuting down the rule f↓, the problematic cases are as before caused by the rules u↓, ls, rs, and α↓. They are all similar and cause the need of the rule $\mathsf{v}\!\downarrow$. Below is the case for ls:

$$\mathsf{f}\!\downarrow\frac{S(\exists a.[A \parr B] \otimes C)}{\mathsf{ls}\,\dfrac{S([A \parr B] \otimes C)}{S[A \parr (B \otimes C)]}} \quad\rightarrow\quad \mathsf{v}\!\downarrow\frac{S(\exists a.[A \parr B] \otimes C)}{\mathsf{ls}\,\dfrac{S([\exists a.A \parr \exists a.B] \otimes C)}{\mathsf{f}\!\downarrow\dfrac{S[\exists a.A \parr (\exists a.B \otimes C)]}{\mathsf{f}\!\downarrow\dfrac{S[A \parr (\exists a.B \otimes C)]}{S[A \parr (B \otimes C)]}}}}$$

We first eliminate the $\mathsf{v}\!\downarrow$-instance by applying Lemma 3.5, and then continue permuting the two new $\mathsf{f}\!\downarrow$ further down. To see that this terminates, we can use a multiset-ordering where the elements of the multiset are the sizes of the principal formulas of the $\mathsf{f}\!\downarrow$ to be permuted down. □

Observation 3.6. The attentive reader might wonder why there are two versions of the "switch" in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$, the *left switch* ls, and the *right switch* rs. For completeness (Theorem 3.1), the ls-rule would be sufficient, but for obtaining the decomposition in Theorem 3.4, we need the rs-rule as well, because we do not have associativity and commutativity of $\otimes$ in $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$.

If a derivation $\mathscr{D}$ uses only the rules $\alpha\!\downarrow, \sigma\!\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\!\downarrow$, then premise and conclusion of $\mathscr{D}$ (and every formula in between the two) must contain the same atom occurrences. Hence, the *atomic flow-graph* (Buss, 1991; Guglielmi and Gundersen, 2008) of the derivation $\mathscr{D}$ defines a bijection between the atom occurrences of premise and conclusion of $\mathscr{D}$. Here is an example of a derivation together with its flow-graph.

$$\begin{array}{rc}
 & \forall a.\forall c.([a^\perp \parr a] \otimes [c^\perp \parr c]) \\ \hline
\mathsf{ls} & \forall a.\forall c.[a^\perp \parr (a \otimes [c^\perp \parr c])] \\ \hline
\mathsf{rs} & \forall a.\forall c.[a^\perp \parr [(a \otimes c^\perp) \parr c]] \\ \hline
\mathsf{u}\!\downarrow & \forall a.[\exists c.a^\perp \parr \forall c.[(a \otimes c^\perp) \parr c]] \\ \hline
\mathsf{u}\!\downarrow & \forall a.[\exists c.a^\perp \parr [\exists c.(a \otimes c^\perp) \parr \forall c.c]] \\ \hline
\mathsf{u}\!\downarrow & [\forall a.\exists c.a^\perp \parr \exists a.[\exists c.(a \otimes c^\perp) \parr \forall c.c]]
\end{array} \tag{2}$$

To avoid crossings in the flow-graph, we left some applications of $\alpha\!\downarrow$ and $\sigma\!\downarrow$ implicit.

4. Expansion trees for MLL2

In their essence, expansion trees (Miller, 1987) are enriched formula trees that encode two formulas, called the *deep formula* and the *shallow formula*, at the same time. The shallow formula is the conclusion of the proof, and the deep formula is a propositional tautology. Miller's original work makes indirect use of the properties of classical logic, and it is an interesting question whether we can achieve a similar data structure for linear logic. In one sense, the situation is more difficult because there is no simple Boolean semantics, but on the other hand, the situation is simpler because we do not have to deal with contraction. We start with a set $\mathscr{E}$ of *expanded formulas* that are generated by

$$\mathscr{E} \quad ::= \quad \perp \mid 1 \mid \mathscr{A} \mid \mathscr{A}^\perp \mid [\mathscr{E} \parr \mathscr{E}] \mid (\mathscr{E} \otimes \mathscr{E}) \mid \forall \mathscr{A}.\mathscr{E} \mid \exists \mathscr{A}.\mathscr{E} \mid \boxminus \mathscr{A}.\mathscr{E} \mid \boldsymbol{\exists} \mathscr{A}.\mathscr{E}$$

There are only two additional syntactic primitives: the $\boxminus$, called *virtual existential quantifier*, and the $\boldsymbol{\exists}$, called *bold existential quantifier*. An *expanded sequent* is a finite list of expanded formulas, separated by commas. We denote expanded sequents by capital Greek letters (Γ, Δ,

…). For disambiguation, the formulas/sequents introduced in Section 2 (i.e., those without $\mathbb{\exists}$ and $\boldsymbol{\exists}$) will also be called *simple formulas/sequents*. In the following we will identify formulas with their syntax trees, where the leaves are decorated by elements of $\mathscr{A} \cup \mathscr{A}^\perp \cup \{1, \perp\}$. We can think of the inner nodes as decorated either with the connectives/quantifiers $\otimes$, $\parr$, $\forall a$, $\exists a$, $\boldsymbol{\exists} a$, $\mathbb{\exists} a$, or with the whole subformula rooted at that node. For this reason we will use capital Latin letters (A, B, C, …) to denote nodes in a formula tree. We write $A \preccurlyeq B$ if A is a (not necessarily proper) ancestor of B, i.e., B is a subformula occurrence in A. We write $\ell\Gamma$ (resp. ℓA) for denoting the set of leaves of a sequent Γ (resp. formula A).

Definition 4.1. A *stretching σ for a sequent* Γ consists of two binary relations $\overset{\sigma}{\underset{+}{\to}}$ and $\overset{\sigma}{\underset{-}{\to}}$ on the set of nodes of Γ (i.e., its subformula occurrences) such that $\overset{\sigma}{\underset{+}{\to}}$ and $\overset{\sigma}{\underset{-}{\to}}$ are disjoint, and whenever $A \overset{\sigma}{\underset{+}{\to}} B$ or $A \overset{\sigma}{\underset{-}{\to}} B$ then $A = \boldsymbol{\exists} a.A'$ with $A' \preccurlyeq B$ in Γ. An *expansion tree* is an expanded formula E or sequent Γ with a stretching, denoted by $E \blacktriangleleft \sigma$ or $\Gamma \blacktriangleleft \sigma$, respectively.

A stretching consists of edges connecting $\boldsymbol{\exists}$-nodes with some of its subformulas, and these edges can be positive or negative. Their purpose is to mark the places of the substitution of the atoms quantified by the $\boldsymbol{\exists}$. When writing an expansion tree $\Gamma \blacktriangleleft \sigma$, we will draw the stretching edges either inside Γ when it is written as a tree, or below Γ when it is written as string. The positive edges are dotted and the negative ones are dashed. Examples are shown in Figures 3 and 4. The next step is to define the deep and the shallow formula of an expansion tree.

Definition 4.2. For an expansion tree $E \blacktriangleleft \sigma$, we define the *deep formula*, denoted by $\lceil E \blacktriangleleft \sigma \rceil$, and the *shallow formula*, denoted by $\lfloor E \blacktriangleleft \sigma \rfloor$, inductively as follows:

$$
\begin{array}{lll}
\lceil 1 \blacktriangleleft \emptyset \rceil = 1 & \lceil a \blacktriangleleft \emptyset \rceil = a & \lceil A \parr B \blacktriangleleft \sigma \rceil = \lceil A \blacktriangleleft \sigma' \rceil \parr \lceil B \blacktriangleleft \sigma'' \rceil \\
\lceil \perp \blacktriangleleft \emptyset \rceil = \perp & \lceil a^\perp \blacktriangleleft \emptyset \rceil = a^\perp & \lceil A \otimes B \blacktriangleleft \sigma \rceil = \lceil A \blacktriangleleft \sigma' \rceil \otimes \lceil B \blacktriangleleft \sigma'' \rceil
\end{array}
$$

$$
\begin{array}{ll}
\lceil \forall a.A \blacktriangleleft \sigma \rceil = \forall a.\lceil A \blacktriangleleft \sigma \rceil & \lceil \mathbb{\exists} a.A \blacktriangleleft \sigma \rceil = \exists a.\lceil A \blacktriangleleft \sigma \rceil \\
\lceil \exists a.A \blacktriangleleft \sigma \rceil = \exists a.\lceil A \blacktriangleleft \sigma \rceil & \lceil \boldsymbol{\exists} a.A \blacktriangleleft \sigma \rceil = \lceil A \blacktriangleleft \sigma' \rceil
\end{array}
$$

$$
\begin{array}{lll}
\lfloor 1 \blacktriangleleft \emptyset \rfloor = 1 & \lfloor a \blacktriangleleft \emptyset \rfloor = a & \lfloor A \parr B \blacktriangleleft \sigma \rfloor = \lfloor A \blacktriangleleft \sigma' \rfloor \parr \lfloor B \blacktriangleleft \sigma'' \rfloor \\
\lfloor \perp \blacktriangleleft \emptyset \rfloor = \perp & \lfloor a^\perp \blacktriangleleft \emptyset \rfloor = a^\perp & \lfloor A \otimes B \blacktriangleleft \sigma \rfloor = \lfloor A \blacktriangleleft \sigma' \rfloor \otimes \lfloor B \blacktriangleleft \sigma'' \rfloor
\end{array}
$$

$$
\begin{array}{ll}
\lfloor \forall a.A \blacktriangleleft \sigma \rfloor = \forall a.\lfloor A \blacktriangleleft \sigma \rfloor & \lfloor \mathbb{\exists} a.A \blacktriangleleft \sigma \rfloor = \lfloor A \blacktriangleleft \sigma \rfloor \\
\lfloor \exists a.A \blacktriangleleft \sigma \rfloor = \exists a.\lfloor A \blacktriangleleft \sigma \rfloor & \lfloor \boldsymbol{\exists} a.A \blacktriangleleft \sigma \rfloor = \exists a.\lfloor \tilde{A} \blacktriangleleft \tilde{\sigma} \rfloor
\end{array}
$$

where σ' is the restriction of σ to A, and σ'' is the restriction of σ to B. The expanded formula $\tilde{A}$ in the last line is obtained from A as follows: For every node B with $A \preccurlyeq B$ and $\boldsymbol{\exists} a.A \overset{\sigma}{\underset{+}{\to}} B$ remove the whole subtree B and replace it by a, and for every B with $\boldsymbol{\exists} a.A \overset{\sigma}{\underset{-}{\to}} B$ replace B by $a^\perp$. The stretching $\tilde{\sigma}$ is the restriction of σ to $\tilde{A}$. For an expanded sequent Γ, we proceed analogously.

Note that the deep and and the shallow formula an expansion tree differ only on $\boldsymbol{\exists}$ and $\mathbb{\exists}$. In the deep formula, the $\mathbb{\exists}$ is treated as ordinary $\exists$, and the $\boldsymbol{\exists}$ is completely ignored. In the shallow formula, the $\mathbb{\exists}$ is ignored, and the $\boldsymbol{\exists}$ uses the information of the stretching to "undo the substitution". To provide this information on the location is the purpose of the stretching. To ensure that we really only "undo the substitution" instead of doing something weird, we need some further constraints, which are given by Definition 4.3 below.

Before, we need some additional notation. Let $\Gamma \blacktriangleleft \sigma$ be given, and let A and B be nodes in Γ

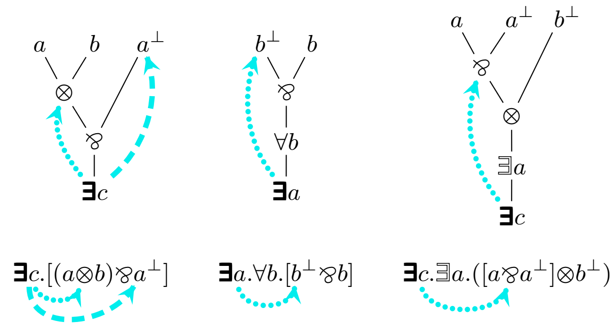


Fig. 3. Examples of expansion trees that are not appropriate

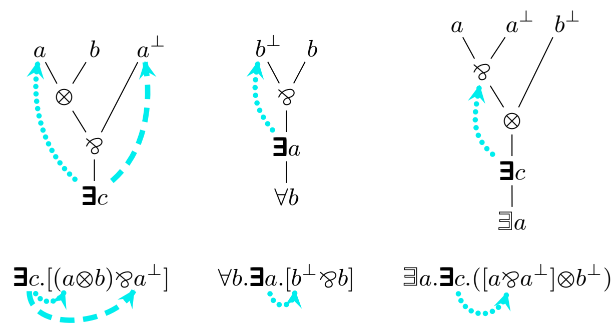


Fig. 4. Appropriate examples of expansion trees

with A being a quantifier node and $A \preccurlyeq B$. Then we write $A \rightharpoonup B$ if A is a $\boldsymbol{\exists}$-node and there is a stretching edge between A and B, or A is an ordinary quantifier node and B is the variable (or its negation) that is bound in A.

Definition 4.3. An expansion tree $\Gamma \blacktriangleleft \sigma$ is *appropriate*, if the following three conditions hold:

1 *Same-formula-condition:* For all nodes A, B_1, B_2,
if $A \overset{\sigma}{\underset{+}{\rightharpoonup}} B_1$ and $A \overset{\sigma}{\underset{+}{\rightharpoonup}} B_2$, then $\lfloor B_1 \blacktriangleleft \sigma_1 \rfloor = \lfloor B_2 \blacktriangleleft \sigma_2 \rfloor$,
if $A \overset{\sigma}{\underset{-}{\rightharpoonup}} B_1$ and $A \overset{\sigma}{\underset{-}{\rightharpoonup}} B_2$, then $\lfloor B_1 \blacktriangleleft \sigma_1 \rfloor = \lfloor B_2 \blacktriangleleft \sigma_2 \rfloor$,
if $A \overset{\sigma}{\underset{+}{\rightharpoonup}} B_1$ and $A \overset{\sigma}{\underset{-}{\rightharpoonup}} B_2$, then $\lfloor B_1 \blacktriangleleft \sigma_1 \rfloor = \lfloor B_2 \blacktriangleleft \sigma_2 \rfloor^\perp$,
where σ_1 and σ_2 are the restrictions of σ to B_1 and B_2, respectively.

2 *No-capture-condition:* For all nodes A_1, A_2, B_1, B_2, where A_1 is a $\boldsymbol{\exists}$-node,
if $A_1 \rightharpoonup B_1$ and $A_2 \rightharpoonup B_2$ and $A_1 \preccurlyeq A_2$ and $B_1 \preccurlyeq B_2$, then $B_1 \preccurlyeq A_2$.

3 *Not-free-condition:* For all subformulas $\exists a.A$, the formula $\lfloor A \blacktriangleleft \sigma' \rfloor$ does not contain a free occurrence of a, where σ' is the restriction of σ to A.

The first condition above says that in a substitution a variable is instantiated everywhere by the same formula B. The second condition ensures that there is no variable capturing in such a substitution step. The third condition is exactly the side condition of the rule $\mathsf{f}\!\downarrow$ in Figure 2. For better explaining the three conditions above, we show in Figure 3 three examples of pairs $\Gamma \blacktriangleleft \sigma$ that are not appropriate: the first fails Condition 1, the second fails Condition 2, and the third fails Condition 3. In Figure 4 all three examples are appropriate.

We can characterize expansion trees $\Gamma \blacktriangleleft \sigma$ that are appropriate very naturally in terms of deep inference.

Lemma 4.4. For every derivation $\begin{array}{c} D \\ \{\mathsf{n}{\downarrow},\mathsf{f}{\downarrow}\}\,\Big\Vert\,\mathscr{D} \\ C \end{array}$ there is an appropriate expansion tree $\Gamma \blacktriangleleft \sigma$ with $D = \lceil \Gamma \blacktriangleleft \sigma \rceil$ and $C = \lfloor \Gamma \blacktriangleleft \sigma \rfloor$. Conversely, if $\Gamma \blacktriangleleft \sigma$ is appropriate, then $\begin{array}{c} \lceil \Gamma \blacktriangleleft \sigma \rceil \\ \{\mathsf{n}{\downarrow},\mathsf{f}{\downarrow}\}\,\Big\Vert\,\mathscr{D} \\ \lfloor \Gamma \blacktriangleleft \sigma \rfloor \end{array}$ for some derivation $\mathscr{D}$.

Proof. We begin by extracting $\Gamma \blacktriangleleft \sigma$ from $\mathscr{D}$. For this, we proceed by induction on the length of $\mathscr{D}$. In the base case, let $\Gamma = D = C$ and σ be empty. In the inductive case consider the bottommost rule instance $\rho\,\dfrac{C'}{C}$ in $\mathscr{D}$, which is either $\mathsf{f}{\downarrow}\,\dfrac{S\{\exists a.A\}}{S\{A\}}$ or $\mathsf{n}{\downarrow}\,\dfrac{S\{A\langle a \backslash B\rangle\}}{S\{\exists a.A\}}$ and let $\Gamma' \blacktriangleleft \sigma'$ be obtained by induction hypothesis, in particular, $C' = \lfloor \Gamma' \blacktriangleleft \sigma' \rfloor$.

—If ρ is $\mathsf{f}{\downarrow}$, then we construct Γ from Γ' as follows: If the $\exists$ to which $\mathsf{f}{\downarrow}$ is applied appears in Γ' as ordinary $\exists$, then replace it by a $\exists\!\!\exists$-node, and let $\sigma = \sigma'$. If the $\exists$ is in fact a $\boldsymbol{\exists}$, then completely remove it, and let σ be obtained from σ' by removing all edges adjacent to that $\boldsymbol{\exists}$. In both cases the same-formula-condition and the no-capture-condition (4.3-1 and 4.3-2) are satisfied for $\Gamma \blacktriangleleft \sigma$ by induction hypothesis (because $\Gamma' \blacktriangleleft \sigma'$ is appropriate). The not-free-condition (4.3-3) holds because otherwise the $\mathsf{f}{\downarrow}$ would not be a valid rule application.

—If ρ is $\mathsf{n}{\downarrow}$, we insert an $\boldsymbol{\exists}$-node at the position where the $\mathsf{n}{\downarrow}$-rule is applied and let σ be obtained from σ' by adding a positive (resp. negative) edge from this new $\boldsymbol{\exists}$ to every occurrence of B in C' which is replaced by a (resp. $a^{\perp}$) in C. Then clearly the same-formula-condition is satisfied since it is everywhere the same B which is substituted. Let us now assume by way of contradiction, that the no-capture-condition is violated. This means we have A_1, A_2, B_1, B_2 such that $A_1 \rightharpoondown B_1$ and $A_2 \rightharpoondown B_2$ and $A_1 \preccurlyeq A_2$ and $B_1 \preccurlyeq B_2$ and $B_1 \not\preccurlyeq A_2$. Note that by the definition of stretching we have that A_1, A_2, B_1, B_2 all sit on the same branch in Γ. Therefore we must have that $A'_2 \preccurlyeq B_1$, where A'_2 is child of A_2. Since the no-capture-condition is satisfied for $\Gamma' \blacktriangleleft \sigma'$ we have that either A_1 or A_2 is the newly introduced $\boldsymbol{\exists}$. Note that it cannot be A_2 because then B_1 would not be visible in $\lfloor \Gamma' \blacktriangleleft \sigma' \rfloor$ because it has been replaced by the variable a bound in A_1. Since B_2 is inside B_1 it would also be invisible in $\lfloor \Gamma' \blacktriangleleft \sigma' \rfloor$. Hence the new $\boldsymbol{\exists}$ must be A_1. Without loss of generality, let $A_1 = \boldsymbol{\exists} a.A'_1$. Then our $\mathsf{n}{\downarrow}$-instance must look like

$$\mathsf{n}{\downarrow}\,\frac{S\{A'_1\{\mathsf{Q}b.A'_2\{B_1\{b\}\}\}\}}{S\{\exists a.\tilde{A}'_1\{\mathsf{Q}b.\tilde{A}'_2\{a\}\}\}} \tag{3}$$

where a is substituted by $B_1\{b\}$ everywhere inside $\tilde{A}'_1\{\mathsf{Q}b.\tilde{A}'_2\{a\}\}$ and Q is either $\forall$ or $\exists$. Clearly, the variable b is captured. Therefore (3) is not a valid rule application. Hence, the no-capture-condition must be satisfied. Finally, the not-free-condition could only be violated in a situation as above where A_2 is a $\exists\!\!\exists$-node. But since (3) is not valid, the not-free-condition does also hold.

Conversely, for constructing $\mathscr{D}$ from $\Gamma \blacktriangleleft \sigma$, we proceed by induction on the number of $\exists\!\!\exists$ and $\boldsymbol{\exists}$ in Γ. The base case is trivial. Now pick in Γ an $\exists\!\!\exists$ or $\boldsymbol{\exists}$ which is minimal wrt. $\preccurlyeq$, i.e., has no other $\exists\!\!\exists$ or $\boldsymbol{\exists}$ as ancestor.

—If we pick an $\exists$, say $\Gamma = S\{\exists a.A\}$, then let $\Gamma' = S\{\exists a.A\}$. By the not-free-condition, a does not appear free in $\lfloor A \blacktriangleleft \sigma \rfloor$. Hence $\mathsf{f}\!\downarrow \frac{\lfloor \Gamma' \blacktriangleleft \sigma \rfloor}{\lfloor \Gamma \blacktriangleleft \sigma \rfloor}$ is a proper application of $\mathsf{f}\!\downarrow$.

—If we pick an $\boldsymbol{\exists}$, say $\Gamma = S\{\boldsymbol{\exists} a.A\}$, then let $\Gamma' = S\{A\}$ and let σ' be obtained from σ by removing all edges coming out of the selected $\boldsymbol{\exists} a$. We now have to check that $\mathsf{n}\!\downarrow \frac{\lfloor \Gamma' \blacktriangleleft \sigma' \rfloor}{\lfloor \Gamma \blacktriangleleft \sigma \rfloor}$ is a proper application of $\mathsf{n}\!\downarrow$. Indeed, by the same-formula-condition, every occurrence of a bound by $\boldsymbol{\exists} a$ in $\lfloor \Gamma \blacktriangleleft \sigma \rfloor$ is substituted by the same formula in $\lfloor \Gamma' \blacktriangleleft \sigma' \rfloor$. The no-capture-condition ensures that no other variable is captured by this.

In both cases we have that $\lceil \Gamma' \blacktriangleleft \sigma' \rceil = \lceil \Gamma \blacktriangleleft \sigma \rceil$, and we can proceed by induction hypothesis. □

We can explain the idea of the previous lemma by considering again the examples in Figures 3 and 4. To the non-appropriate examples in Figure 3 would correspond the following **incorrect** derivations:

$$\mathsf{n}\!\downarrow \frac{[(a \otimes b) \parr a^\perp]}{\exists c.[c \parr c^\perp]} \qquad \mathsf{n}\!\downarrow \frac{\forall b.[b^\perp \parr b]}{\exists a.\forall b.[a \parr b]} \qquad \mathsf{f}\!\downarrow \frac{\exists a.([a \parr a^\perp] \otimes b^\perp)}{\mathsf{n}\!\downarrow \dfrac{([a \parr a^\perp] \otimes b^\perp)}{\exists c.(c \otimes b^\perp)}}$$

And to the appropriate examples in Figure 4 correspond the following **correct** derivations:

$$\mathsf{n}\!\downarrow \frac{[(a \otimes b) \parr a^\perp]}{\exists c.[(c \otimes b) \parr c^\perp]} \qquad \mathsf{n}\!\downarrow \frac{\forall b.[b^\perp \parr b]}{\forall b.\exists a.[a \parr b]} \qquad \mathsf{n}\!\downarrow \frac{\exists a.([a \parr a^\perp] \otimes b^\perp)}{\mathsf{f}\!\downarrow \dfrac{\exists a.\exists c.(c \otimes b^\perp)}{\exists c.(c \otimes b^\perp)}}$$

5. Proof graphs for MLL2

After studying the "quantifier part" of MLL2, we will now look into the "propositional part". Since the units are present, we have to have more structure than ordinary proof nets. Furthermore, in linear logic, we cannot fully separate the quantifier part from the propositional part, as it is the case with classical logic.¶ We follow the ideas presented in (Straßburger and Lamarche, 2004) and (Lamarche and Straßburger, 2006) where the axiom linking of multiplicative proof nets has been replaced by a *linking formula* to accommodate the units 1 and ⊥. In such a linking formula, the ordinary axiom links are replaced by ⊗-nodes, which are then connected by ⅋s. A unit can then be attached to a sublinking by another ⊗, and so on. Here we extend the syntax for the linking formula by an additional construct to accommodate the quantifiers. The set $\mathscr{L}$ of *linking formulas* is generated by the grammar

$$\mathscr{L} ::= \perp \mid (\mathscr{A} \otimes \mathscr{A}^\perp) \mid (1 \otimes \mathscr{L}) \mid [\mathscr{L} \parr \mathscr{L}] \mid \exists \mathscr{A}.\mathscr{L}$$

The basic idea of our proof graphs is to attach a linking formula to an expansion tree. This is similar to Miller's idea of attaching a *mating* (Andrews, 1976) to an expansion tree in classical logic (Miller, 1987).

¶ The reason is that we cannot transform every formula into an equivalent prenex normal form, since the two formulas $A \parr \exists a.B(a)$ and $\exists a.(A \parr B(a))$ are not equivalent in general.

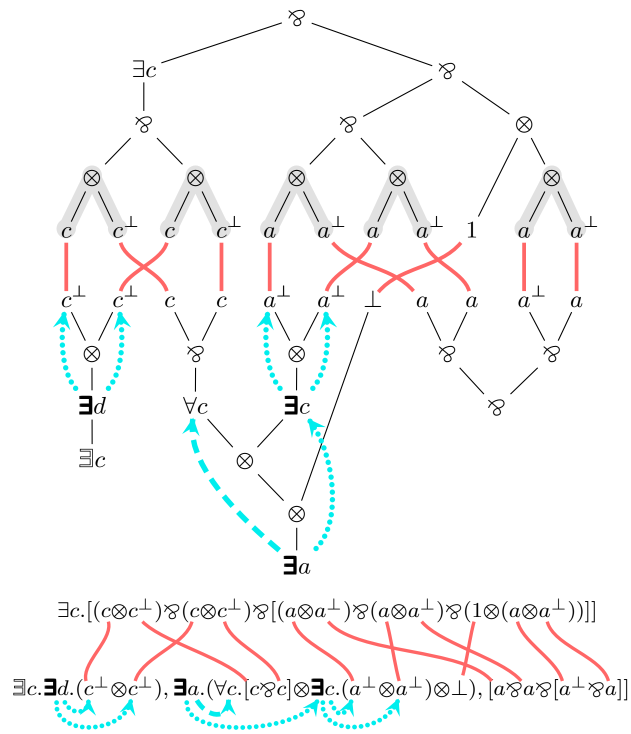


Fig. 5. Two ways of writing a proof graph

Definition 5.1. A *pre-proof graph* is a quadruple, denoted by $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$, where P a linking formula, $\Gamma \blacktriangleleft \sigma$ is an expansion tree, and ν is a bijection $\ell\Gamma \overset{\nu}{\to} \ell P$ from the leaves of Γ to the leaves of P, such that only dual atoms/units are paired up. If Γ is simple, we say that the pre-proof graph is *simple*. In this case σ is empty, and we can simply write $P \overset{\nu}{\triangleright} \Gamma$.

The "pre-" means that we do not yet know whether it really comes from an actual proof, and that we need a *correctness criterion* to distinguish the pre-proofs from the proofs.

Observe that due to the mobility of $\perp$, we need to introduce the notion of proof graph. A proof net will then be an equivalence class of proof graphs (see also (Lamarche and Straßburger, 2006) for details). If there are no units present, then the notions of (pre-)proof graph and (pre-)proof net coincide.

For $B \in \ell\Gamma$ we write B^ν for its image under ν in ℓP. When we draw a pre-proof graph $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$, then we write P above Γ (as trees or as strings) and the leaves are connected by edges according to ν. Figure 5 shows an example written in both ways. To help the reader we marked the traditional "axiom links" in the tree-like version.

Let us now turn our attention towards correctness. For this we concentrate first on simple pre-proof graphs and begin with pure multiplicative correctness, using the standard Danos-Regnier criterion (Danos and Regnier, 1989).

Definition 5.2. A *switching* s of a simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ is the graph that is obtained from $P \overset{\nu}{\triangleright} \Gamma$ by removing for each $\parr$-node one of the two edges connecting it to its children.

A simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ is *multiplicatively correct* if all its switchings are acyclic and connected (Danos and Regnier, 1989). For a pre-proof graph $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ we define *multiplicative correctness* accordingly, but we ignore the edges of the stretching when checking acyclicity and connectedness.

For multiplicative correctness the quantifiers are treated as unary connectives and are therefore completely irrelevant. The example in Figure 5 is multiplicatively correct. For involving the quantifiers into a correctness criterion, we need some more conditions.

Let s be a switching for $P \overset{\nu}{\triangleright} \Gamma$, and let A and B be two nodes in Γ. We write $A \frown\!\textcircled{s}\!\frown B$ for saying that there is a path in s from A to B which starts from A going up to one of its children and which comes into B down from one of its children, and we write $A \frown\!\textcircled{s}\!\smile B$ if the path comes into B from its parent below. Similarly, we write $A \smile\!\textcircled{s}\!\frown B$ (resp. $A \smile\!\textcircled{s}\!\smile B$) if the path starts from A going down to its parent and comes into B from above (resp. from below).

Let Γ be a simple sequent, and let A and B be nodes in Γ with $A \preccurlyeq B$. Then the *quantifier depth* of B in A, denoted by $\bigtriangledown_A B$, is defined to be the number of quantifier nodes on the path from A to B (including A if it happens to be an $\forall a$ or an $\exists a$, but not including B). Similarly we define $\bigtriangledown_\Gamma B$. Now assume we have a simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ and quantifier nodes A' in P and A in Γ. We say A and A' are *partners* if there is a leaf $B \in \ell\Gamma$ with $A \preccurlyeq B$ in Γ, and $A' \preccurlyeq B^\nu$ in P, and $\bigtriangledown_A B = \bigtriangledown_{A'} B^\nu$. We denote this by $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A$.

Definition 5.3. We say a simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ is *well-nested* if the following five conditions are satisfied:

1 *Same-depth-condition:* For every $B \in \ell\Gamma$, we have $\bigtriangledown_\Gamma B = \bigtriangledown_P B^\nu$.
2 *Same-variable-condition:* whenever $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A$, then A' and A quantify the same variable.
3 *One-$\exists$-condition:* For every quantifier node A in Γ there is exactly one $\exists$-node A' in P with $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A$.
4 *One-$\forall$-condition:* For every $\exists$-node A' in P there is exactly one $\forall$-node A in Γ with $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A$.
5 *No-down-path-condition:* If $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A_1$ and $A' \overset{P\,\Gamma}{\leftarrow\!\blacksquare\!\rightarrow} A_2$ for some A' in P and A_1, A_2 in Γ, then there is no switching s with $A_1 \smile\!\textcircled{s}\!\smile A_2$.

To understand the motivation behind this definition, observe that every quantifier node in P must be an $\exists$, and every quantifier node in Γ has exactly one of them as partner. On the other hand, an $\exists$ in P can have many partners in Γ, but exactly one of them has to be an $\forall$. Following Girard (Girard, 1987), we can call an $\exists$ in P together with its partners in Γ the *doors of an $\forall$-box* and the sub-graph induced by the nodes that have such a door as ancestor is called the *$\forall$-box* associated to the unique $\forall$-door. Even if the boxes are not really present, we can use the terminology to relate our work to Girard's. There should be no surprise here: If we claim at some point that our proof graphs capture the essential information of a proof, we must be able to recover a sequent calculus proof, which necessarily contains the Girard-boxes. Furthermore, all the properties of these boxes that are postulated in (Girard, 1987), e.g., that every box is correct in itself, follow from the global multiplicative correctness and the five conditions above. In order to help the reader to understand these five conditions, we show in Figure 6 six simple pre-proof graphs, where the first fails Condition 1, the second one fails Condition 2, and so on; only the sixth one is well-nested.

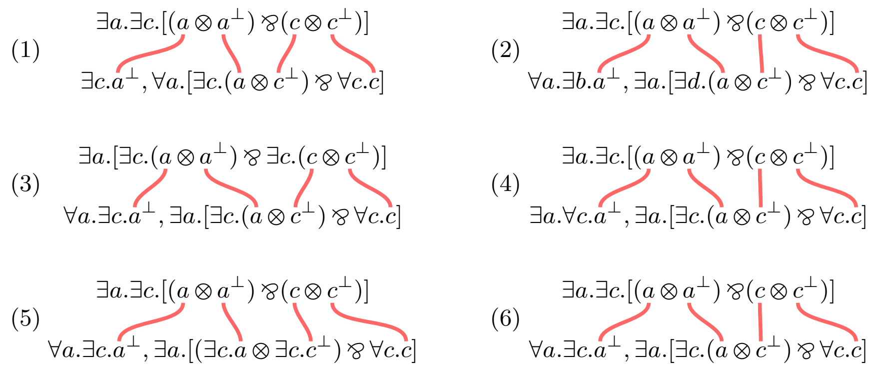


Fig. 6. Examples (1)–(5) are not well-nested, only (6) is well-nested

Definition 5.4. A simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ is *correct* if it is well-nested and multiplicatively correct. In this case we will also speak of a *simple proof graph*.

Definition 5.5. We say that a pre-proof graph $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ is *correct* if the simple pre-proof graph $P \overset{\nu}{\triangleright} \lceil \Gamma \blacktriangleleft \sigma \rceil$ is correct and the expansion tree $\Gamma \blacktriangleleft \sigma$ is appropriate. In this case we say that $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ is a *proof graph* and $\lfloor \Gamma \blacktriangleleft \sigma \rfloor$ is its *conclusion*.

The example in Figure 5 is correct. There $\lceil \Gamma \blacktriangleleft \sigma \rceil$ is

$$\vdash \exists c.(c^\perp \otimes c^\perp), (\forall c.[c \parr c] \otimes (a^\perp \otimes a^\perp) \otimes \bot), [a \parr a \parr [a^\perp \parr a]]$$

and the conclusion $\lfloor \Gamma \blacktriangleleft \sigma \rfloor$ is

$$\vdash \exists d.(d \otimes d), \exists a.(a^\perp \otimes a \otimes \bot), [a \parr a \parr [a^\perp \parr a]] \quad .$$

6. The relation between simple proof graphs and deep inference

With Lemma 4.4 we already gave a deep inference characterization of expansion trees. In this section we do something similar for simple proof graphs.

Let us begin with a characterization of linking formulas.

Lemma 6.1. An MLL2 formula P is a linking formula if and only if there is a derivation

$$\{\mathsf{ai}\downarrow, \bot\downarrow, 1\downarrow, \mathsf{e}\downarrow\} \left\Vert \begin{array}{c} 1 \\ \mathscr{D} \\ P^\perp \end{array}\right. \quad . \tag{4}$$

Proof. We can proceed by structural induction on P to construct $\mathscr{D}$. The base case is trivial.

Here are the four inductive cases:

$$\mathsf{ai}\!\downarrow \frac{\{1\}}{[a^\perp \mathbin{⅋} a]} \qquad \bot\!\downarrow \frac{\begin{array}{c}1\\ \Vert\, \mathscr{D}'\\ A\end{array}}{[\bot \mathbin{⅋} A]} \qquad \begin{array}{c}1\\ \Vert\, \mathscr{D}'\\ 1\!\downarrow \dfrac{B}{(1 \otimes B)}\\ \Vert\, \mathscr{D}''\\ (A \otimes B)\end{array} \qquad \begin{array}{c}\mathsf{e}\!\downarrow \dfrac{1}{\forall a.1}\\ \Vert\, \mathscr{D}'\\ \forall a.A\end{array}$$

where $\mathscr{D}'$ and $\mathscr{D}''$ always exist by induction hypothesis. Conversely, we proceed by induction on the length of $\mathscr{D}$ to show that P is a linking formula. We show only the case where the bottommost rule in $\mathscr{D}$ is a $\mathsf{ai}\!\downarrow$. Then by induction hypothesis $S\{1\}^\perp = P\{\bot\}$ is a linking for some context $P\{\ \}$. Hence $S[a^\perp \mathbin{⅋} a]^\perp = P(a \otimes a^\perp)$ is also a linking. The other cases are similar. □

Definition 6.2. If a linking has the shape $S_1(1 \otimes S_2(a \otimes a^\perp))$ for some contexts $S_1\{\ \}$ and $S_2\{\ \}$, then we say that the 1 *governs* the pair $(a \otimes a^\perp)$. Now let P_1 and P_2 be two linkings. We say that P_1 *is weaker than* P_2, denoted by $P_1 \lesssim P_2$, if

— $\ell P_1 = \ell P_2$,
— P_1 and P_2 contain the same set of $\exists$-nodes, and for every $\exists$-node, its set of leaves is the same in P_1 and P_2, and
— whenever a 1 governs a pair $(a \otimes a^\perp)$ in P_2, then it also governs this pair in P_1.

This $\lesssim$ relation can also be characterized by deep inference derivations. For this, we also use the following inference rules:

$$\alpha\!\uparrow \frac{S(A \otimes (B \otimes C))}{S((A \otimes B) \otimes C)} \qquad \text{and} \qquad \sigma\!\uparrow \frac{S(A \otimes B)}{S(B \otimes A)} \tag{5}$$

which are the duals for $\alpha\!\downarrow$ and $\sigma\!\downarrow$, respectively.

Lemma 6.3. Let P_1 and P_2 be two linkings. Then the following are equivalent

1 $P_1 \lesssim P_2$.
2 There is a derivation $\{\alpha\!\downarrow, \sigma\!\downarrow, \mathsf{rs}\} \begin{array}{c}P_1\\ \Vert\, \mathscr{D}\\ P_2\end{array}$.
3 Dually, there is a derivation $\{\alpha\!\uparrow, \sigma\!\uparrow, \mathsf{ls}\} \begin{array}{c}P_2^\perp\\ \Vert\, \mathscr{D}'\\ P_1^\perp\end{array}$.
4 The simple pre-proof graph $P_2 \triangleright P_1^\perp$ is correct.

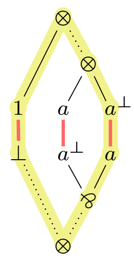
Proof. 1 ⇒ 2: The only way in which P_1 and P_2 can differ from each other are the ⅋-trees above the pairs $(a \otimes a^\perp)$ and where in these trees the 1-occurrences are attached. Therefore, the rules for associativity and commutativity of ⅋ and the rule $\mathsf{rs} \dfrac{S(1 \otimes [B \mathbin{⅋} C])}{S[(1 \otimes B) \mathbin{⅋} C]}$ are sufficient to transform P_1 into P_2.

2 ⇒ 3: The derivation $\mathscr{D}'$ is the dual of $\mathscr{D}$.

3 ⇒ 4: We proceed by induction on the length of $\mathscr{D}'$. Clearly $P_2 \triangleright P_2^\perp$ is correct. Furthermore, all three inference rules $\alpha\!\uparrow$, $\sigma\!\uparrow$, and ls preserve correctness.

4 ⇒ 1: We have $\ell P_1 = \ell P_2$ because $P_2 \triangleright P_1^\perp$ is a simple proof graph. The second condition

in Definition 6.2 follows immediately from the well-nestedness of $P_2 \triangleright P_1^\perp$ and the fact that P_1 and P_2 are both linkings, i.e., do not contain $\forall$-nodes. Therefore, we only have to check the last condition. Assume, by way of contradiction, that there is a 1-occurrence which governs a pair $(a \otimes a^\perp)$ in P_2 but not in P_1, i.e., $P_2 = S_1(1 \otimes S_2(a \otimes a^\perp))$ for some contexts $S_1\{\ \}$ and $S_2\{\ \}$, and $P_1 = S_3[S_4\{1\} \parr S_5(a \otimes a^\perp)]$ for some contexts $S_3\{\ \}$, $S_4\{\ \}$, and $S_5\{\ \}$. This means we have the following situation in $P_2 \triangleright P_1^\perp$



which clearly fails the acyclicity condition. □

The next step is to characterize correctness via deep inference.

Lemma 6.4. A simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ is correct if and only if there is a linking P' with $P' \lesssim P$ and a derivation

$$\begin{array}{c} P'^{\perp} \\ \{\alpha\downarrow, \sigma\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow\} \Big\Vert \mathscr{D} \\ \Gamma \end{array}, \qquad (6)$$

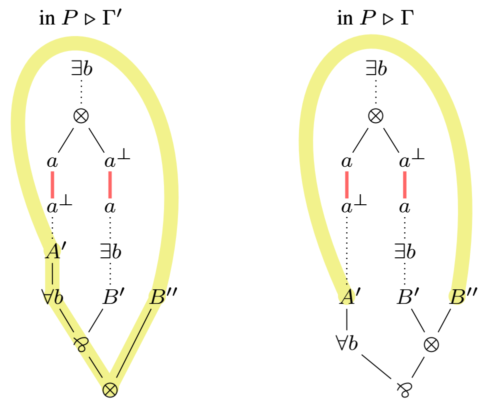
such that ν coincides with the bijection induced by the flow graph of $\mathscr{D}$.

As an example, consider the derivation in (2) which corresponds to example (6) in Figure 6.

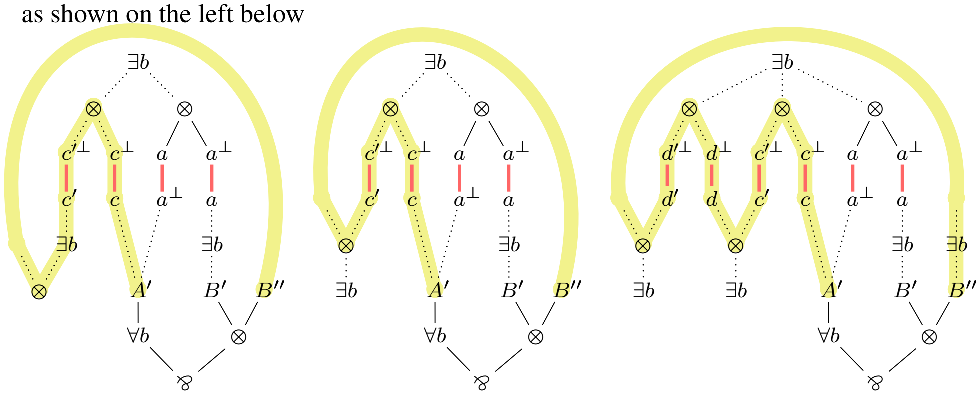
Before we can give the proof of this lemma, we need a series of additional lemmas that we have to show first. We also use the following notation: Let A and B be nodes in Γ with $A \not\lesssim B$ and $B \not\lesssim A$. Then we write $A \overset{\Gamma}{\underset{\otimes}{\frown}} B$ if the first common ancestor of A and B is a $\otimes$, and we write $A \overset{\Gamma}{\underset{\parr}{\frown}} B$ if it is a $\parr$, or if A and B appear in different formulas of Γ. We will also sometimes identify a sequent $\vdash A_1, \ldots, A_n$ with the formula $[A_1 \parr \cdots \parr A_n]$.

Lemma 6.5. Let $\pi = P(a \otimes a^\perp) \overset{\nu}{\triangleright} S[\forall b.A'\{a^\perp\} \parr (B'\{a\} \otimes B'')]$ be a simple proof graph, where $S\{\ \}$, $A'\{\ \}$, and $B'\{\ \}$ are arbitrary contexts, $P\{\ \}$ is a linking formula context, and ν pairs up the shown occurrences of a and $a^\perp$. Then the simple proof graph $\pi' = P(a \otimes a^\perp) \overset{\nu}{\triangleright} S([\forall b.A'\{a^\perp\} \parr B'\{a\}] \otimes B'')$ is also correct.

Proof. Let $\pi = P \overset{\nu}{\triangleright} \Gamma$ and $\pi' = P \overset{\nu}{\triangleright} \Gamma'$. By way of contradiction, assume that $P \overset{\nu}{\triangleright} \Gamma'$ is not correct. If it is not multiplicatively correct then there is a switching s which is either disconnected or cyclic. If it is disconnected, then we get from s immediately a disconnected switching for $P \overset{\nu}{\triangleright} \Gamma$. So, let us assume s is cyclic. The only modification from Γ to Γ' that could produce such a cycle is the change from $A'\{a^\perp\} \overset{\Gamma}{\underset{\parr}{\frown}} B''$ to $A'\{a^\perp\} \overset{\Gamma'}{\underset{\otimes}{\frown}} B''$. Hence, we must have a path $A'\{a^\perp\} \frown\!\!\textcircled{s}\!\!\frown B''$, which is also present in $P \overset{\nu}{\triangleright} \Gamma$. Note that this path cannot pass through $a^\perp$ and a because otherwise we could use $(B'\{a\} \otimes B'')$ to get a cyclic switching for $P \overset{\nu}{\triangleright} \Gamma$. Furthermore, because $P \overset{\nu}{\triangleright} \Gamma$ is well-nested, there is an $\exists b$-node inside $B'\{a\}$ below a. We can draw the following pictures to visualize the situation:



Now, let c be the leaf at which our path leaves $A'\{a^\perp\}$ and goes into P, and let c' be the leaf at which it leaves P and comes back into Γ. by well-nestedness of $P \overset{\nu}{\triangleright} \Gamma$, there must be some $\exists b$-node somewhere in Γ below c'. We also know that our path, coming into Γ at c', goes first down, and at some point goes up again. This turning point must be some $\otimes$-node below c'. Since the $\exists b$-node and the $\otimes$-node are both on the path from c' to the root of the formula, one must be an ancestor of the other. Let us first assume the $\otimes$ is below the $\exists b$. Then our path is of the shape as shown on the left below

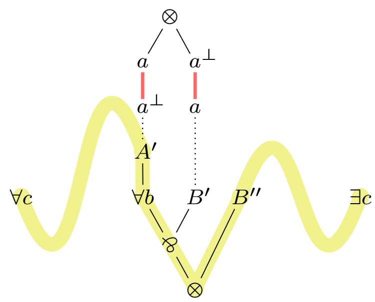


(7)

This, however, is a contradiction to the well-nestedness of $P \overset{\nu}{\triangleright} \Gamma$ because it violates the no-down-path-condition (5.3-5) because there is a path between the $\exists b$ below the c' and the $\exists b$ below the a. Therefore the $\otimes$ must be above the $\exists b$. The situation is now as shown in the middle in (7) above. From the $\otimes$, the path must go up again. Without loss of generality, assume it leaves Γ at d and reenters Γ at d'. For the same reasons as above, there must be an $\exists b$ and a $\otimes$ below d'. And so on. There are two possibilities: either at some point the $\otimes$ is below the $\exists b$, which gives us a violation of the no-down-path-condition as on the left in (7), or we reach eventually B'', as shown on the right in (7).

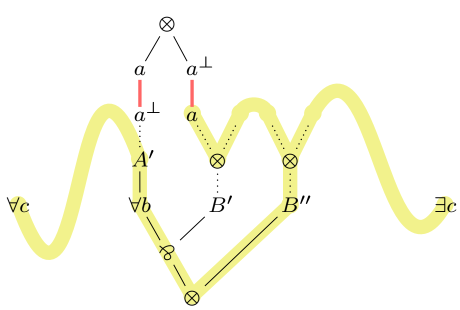
For the same reasons as above, there must be an $\exists b$ inside B'', and we get immediately a violation of the no-down-path-condition because of the short path between the two $\exists b$ above B' and B''. Consequently, $P \overset{\nu}{\triangleright} \Gamma'$ must be multiplicatively correct.

Let us therefore assume $P \overset{\nu}{\triangleright} \Gamma'$ is not well-nested. The same-depth-condition and the same-variable-condition (5.3-1 and 5.3-2) must hold in $P \overset{\nu}{\triangleright} \Gamma'$ because they hold in $P \overset{\nu}{\triangleright} \Gamma$ and the quantifier structure is identical in Γ and Γ'. For the same reasons also the one-$\exists$-condition and the one-$\forall$-condition (5.3-3 and 5.3-4) must hold in $P \stackrel{\nu}{\triangleright} \Gamma'$. Therefore, it must be the no-down-path-condition which is violated. This means we must have in Γ' two quantifier nodes, say $\forall c$ and $\exists c$, connected by a path $\forall c \overset{s}{\smile} \exists c$ in some switching s. Because this path is not present in $P \stackrel{\nu}{\triangleright} \Gamma$ it must pass through the new $\otimes$ between $\forall b.A'\{a^{\perp}\}$ and B'', as follows:

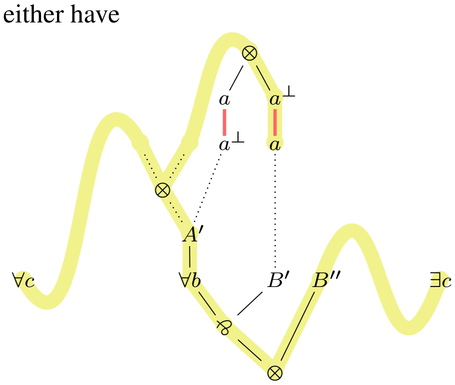


(8)

Since $P \stackrel{\nu}{\triangleright} \Gamma'$ is multiplicatively correct, the switching s must be connected. Therefore there is in s a path from the $\forall b$-node to the a inside B'. This new path must follow the path between $\forall c$ and $\exists c$ for some steps in one direction. Hence, we either have

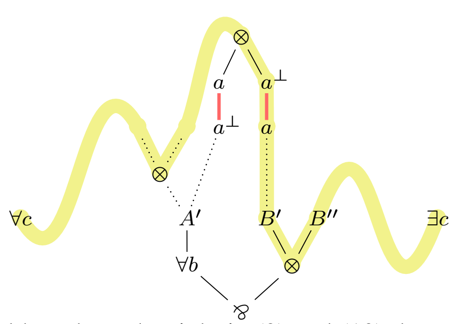


or



(9)

Clearly, the graph on the left in (9) violates the acyclicity condition for $P \stackrel{\nu}{\triangleright} \Gamma'$ as well as for $P \stackrel{\nu}{\triangleright} \Gamma$. And from the one on the right in (9), we can obtain a switching for $P \stackrel{\nu}{\triangleright} \Gamma$ with a path $\forall c \overset{s}{\smile} \exists c$ as follows:

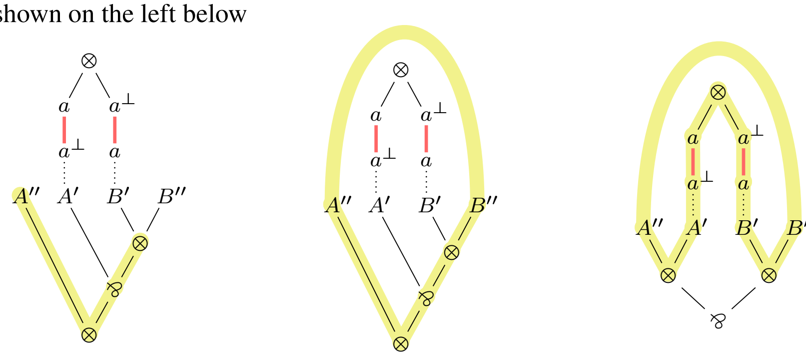


(10)

Contradiction. (Note that although on the right in (9) and (10) the path does not go through the $a^{\perp}$ inside A', this case is not excluded by the argument.) □

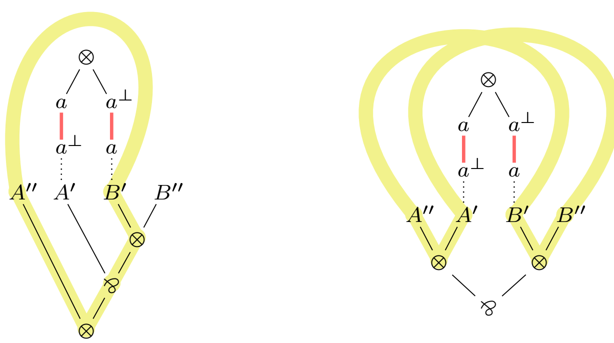
Lemma 6.6. Let $\pi = P(a \otimes a^{\perp}) \stackrel{\nu}{\triangleright} S[(A'' \otimes A'\{a^{\perp}\}) \mathbin{⅋} (B'\{a\} \otimes B'')]$ be a simple proof graph, where $S\{\ \}$, $A'\{\ \}$, $B'\{\ \}$, $P\{\ \}$, and ν are as above. Then at least one of the two proof graphs $\pi' = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S([(A'' \otimes A'\{a^{\perp}\}) \parr B'\{a\}] \otimes B'')$ and $\pi'' = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S(A'' \otimes [A'\{a^{\perp}\} \parr (B'\{a\} \otimes B'')])$ is also correct.

Proof. We start by showing that one of π' and π'' has to be multiplicatively correct. We consider here only the acyclicity condition and leave connectedness to the reader. First, assume that there is a switching s' for π' that is cyclic. Then the cycle must pass through A'', the root $\otimes$ and the $\parr$ as shown on the left below



(11)

Otherwise we could construct a switching with the same cycle in π. If our cycle continues through B'' (as shown in the middle in (11)) then we can use the path from A'' to B'' (which cannot go through A' or B') to construct a cyclic switching s in π as shown on the righ in (11). Hence, the cycle in s' goes through B', giving us a path from A'' to B' (not passing through A'), as shown on the left below:



(12)

By the same argumentation we get a switching s'' in π'' with a path from A' to B'', not going through B'. From s' and s'', we can now construct a switching s for π with a cycle as shown on the right above in (12), which contradicts the correctness of π.

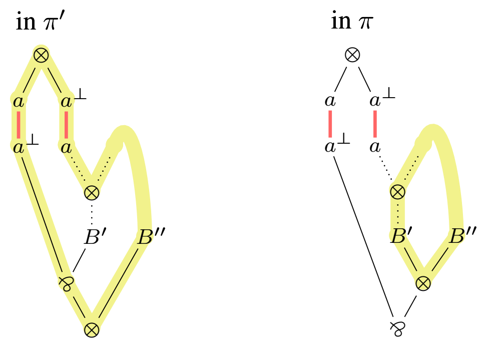
We now have to show that π' and π'' are both well-nested. This can be done in almost literally the same way as in the proof of Lemma 6.5. □

Lemma 6.7. Let $\pi = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S[\forall b.A'\{a^{\perp}\} \parr \exists b.B'\{a\}]$ be a simple proof graph, where $S\{\ \}, A'\{\ \}, B'\{\ \}, P\{\ \}$, and ν are as above. Then $\pi' = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S\{\forall b.[A'\{a^{\perp}\} \parr B'\{a\}]\}$ is also correct.

Proof. Multiplicative correctness of π' follows immediately, because the $\otimes$-$\parr$-structure is the same as in π. Furthermore, all five conditions in Definition 5.3 are obviously preserved by going from π to π'. Hence π' is correct. □

Lemma 6.8. Let $\pi = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S[a^{\perp} \mathbin{\bindnasrepma} (B'\{a\} \otimes B'')]$ be a simple proof graph. Then $\pi' = P(a \otimes a^{\perp}) \overset{\nu}{\triangleright} S([a^{\perp} \mathbin{\bindnasrepma} B'\{a\}] \otimes B'')$ is also correct.

Proof. Well-nestedness of π' follows trivially from the well-nestedness of π. By way of contradiction, assume π' is not multiplicatively correct. Since connectedness is trivial, assume there is a cyclic switching s. If the cycle does not involve the $\otimes$ between $a^{\perp}$ and B'', then we immediately have a cyclic switching for π. Since the cycle involves $a^{\perp}$, it must also involve a. Therefore it must leave $B'\{a\}$ at some other leaf, and finally enter B'' from above, as shown below left.



This means there is a cyclic switching for π, as shown on the right above. Contradiction. ☐

We can now complete our proof.

Proof of Lemma 6.4 Let a simple pre-proof graph $P \overset{\nu}{\triangleright} \Gamma$ be given, and assume we have a linking $P' \lesssim P$ and derivation $\mathscr{D}$ as in (6) whose flow-graph determines ν. By Lemma 6.3 we have a derivation $\mathscr{D}_1$ such that

$$\begin{array}{c} P^{\perp} \\ \{\alpha\uparrow, \sigma\uparrow, \mathsf{ls}\} \Big\| \mathscr{D}_1 \\ P'^{\perp} \\ \{\alpha\downarrow, \sigma\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow\} \Big\| \mathscr{D} \\ \Gamma \end{array} \quad . \tag{13}$$

Now we proceed by induction on the length of $\mathscr{D}_1$ and $\mathscr{D}$ to show that $P \overset{\nu}{\triangleright} \Gamma$ is multiplicatively correct and well-nested. In the base case it is easy to see that $P \triangleright P^{\perp}$ has the desired properties. Now it remains to show that all rules $\alpha\downarrow, \sigma\downarrow, \alpha\uparrow, \sigma\uparrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow$ preserve multiplicative correctness and well-nestedness. For multiplicative correctness it is easy: for $\mathsf{u}\downarrow$ it is trivial because it does not change the $\mathbin{\bindnasrepma}$-$\otimes$-structure of the graph, and for the other rules it is well-known. That well-nestedness is preserved is also easy to see: rules $\alpha\downarrow, \sigma\downarrow, \alpha\uparrow, \sigma\uparrow, \mathsf{ls}, \mathsf{rs}$ do not modify the $\forall$-$\exists$-structure of the graph, and therefore trivially preserve Conditions 1–4 in Definition 5.3. For the no-down-path condition it suffices to observe that it cannot happen that a $\underset{\bindnasrepma}{\smile}$ is changed into $\underset{\otimes}{\smile}$ while going down in a derivation. Finally, it is easy to see that $\mathsf{u}\downarrow$ preserves all five conditions in Definition 5.3.

Conversely, assume $P \overset{\nu}{\triangleright} \Gamma$ is well-nested and multiplicatively correct. For constructing $\mathscr{D}$, we will again need the rule $\mathsf{v}\downarrow$ that has already been used in the proof of Theorem 3.4. We proceed by induction on the distance between $P^{\perp}$ and Γ. For defining this formally, let A be a simple formula and define $\#_{\bindnasrepma} A$ to be the number of pairs $\langle a, b\rangle$ with $a, b \in \ell A$ and $a \underset{\bindnasrepma}{\overset{A}{\smile}} b$, and define $\#_{\exists}A$ to be the number of $\exists$-nodes in A. Now observe that $P^{\perp}$ and Γ have the same set of leaves. We can therefore define $\delta_{\parr}\langle P^{\perp}, \Gamma\rangle = \#_{\parr}\Gamma - \#_{\parr}P^{\perp}$ and $\delta_{\exists}\langle P^{\perp}, \Gamma\rangle = \#_{\exists}\Gamma - \#_{\exists}P^{\perp}$. Note that because of acyclicity it can never happen that for some $a, b \in \ell\Gamma$ we have $a \overset{P^{\perp}}{\underset{\parr}{\smile}} b$ and $a \overset{\Gamma}{\underset{\otimes}{\smile}} b$. Therefore $\delta_{\parr}\langle P^{\perp}, \Gamma\rangle$ is the number of pairs $a, b \in \ell\Gamma$ with $a \overset{P^{\perp}}{\underset{\otimes}{\smile}} b$ and $a \overset{\Gamma}{\underset{\parr}{\smile}} b$. Furthermore, observe that by definition there cannot be any $\exists$-node in $P^{\perp}$. Hence $\delta_{\exists}\langle P^{\perp}, \Gamma\rangle = \#_{\exists}\Gamma$. Now define the *distance between* $P^{\perp}$ *and* Γ to be the pair $\delta\langle P^{\perp}, \Gamma\rangle = \langle \delta_{\parr}\langle P^{\perp}, \Gamma\rangle, \delta_{\exists}\langle P^{\perp}, \Gamma\rangle\rangle$, where we assume the lexicographic ordering.

Let us now pick in Γ a pair of dual atoms, say $a^{\perp}$ and a, which appear in the same "axiom link" in P, i.e., P is $P(a \otimes a^{\perp})$. We now make a case analysis on the relative position of $a^{\perp}$ and a to each other in Γ. Because of acyclicity we must have $a^{\perp} \overset{\Gamma}{\underset{\parr}{\smile}} a$. This means $\Gamma = S[A\{a^{\perp}\} \parr B\{a\}]$ for some contexts $S\{\ \}$, $A\{\ \}$, and $B\{\ \}$. Without loss of generality, we assume that neither A nor B has a $\parr$ as root (otherwise apply $\alpha{\downarrow}$ and $\sigma{\downarrow}$). There are the following cases:

—$A\{\ \}$ and $B\{\ \}$ have both a quantifier as root. Then both must quantify the same variable (because of the same-depth-condition and the same-variable-condition), and at least one of them must be an $\exists$ (because of the one-$\exists$-condition and the one-$\forall$-condition). Assume, without loss of generality, that $A\{a^{\perp}\} = \forall b.A'\{a^{\perp}\}$ and $B\{a\} = \exists b.B'\{a\}$. Then by Lemma 6.7 we have that $P \overset{\nu}{\rhd} \Gamma'$ with $\Gamma' = S\{\forall b.[A'\{a^{\perp}\} \parr B'\{a\}]\}$ is also correct. We can therefore apply the $\mathsf{u}{\downarrow}$-rule and proceed by induction hypothesis because $\delta\langle P^{\perp}, \Gamma'\rangle$ is strictly smaller than $\delta\langle P^{\perp}, \Gamma\rangle$. If A and B have both an $\exists$ as root, we apply the $\mathsf{v}{\downarrow}$-rule instead of $\mathsf{u}{\downarrow}$.

—One of $A\{\ \}$ and $B\{\ \}$ has a quantifier as root and the other a $\otimes$. Without loss of generality, let $A\{\ \} = \forall b.A'\{\ \}$ and $B\{\ \} = (B'\{\ \} \otimes B'')$, i.e., $\Gamma = S[\forall b.A'\{a^{\perp}\} \parr (B'\{a\} \otimes B'')]$. By Lemma 6.5 we have that $P \overset{\nu}{\rhd} \Gamma'$ with $\Gamma' = S([\forall b.A'\{a^{\perp}\} \parr B'\{a\}] \otimes B'')$ is also correct. We can apply ls and proceed by induction hypothesis because $\delta\langle P^{\perp}, \Gamma'\rangle < \delta\langle P^{\perp}, \Gamma\rangle$.

—One of $A\{\ \}$ and $B\{\ \}$ has a quantifier as root and the other is just $\{\ \}$. This is impossible because it is a violation of the same-depth-condition.

—$A\{\ \}$ and $B\{\ \}$ have both a $\otimes$ as root. Assume $\Gamma = S[(A'' \otimes A'\{a^{\perp}\}) \parr (B'\{a\} \otimes B'')]$. By Lemma 6.6, $P \overset{\nu}{\rhd} \Gamma'$ is correct, with either $\Gamma' = S([(A'' \otimes A'\{a^{\perp}\}) \parr B'\{a\}] \otimes B'')$ or $\Gamma' = S(A'' \otimes [A'\{a^{\perp}\} \parr (B'\{a\} \otimes B'')])$. In one case we apply the rs-rule, and in the other the ls-rule. In both cases we have that $\delta\langle P^{\perp}, \Gamma'\rangle$ is strictly smaller than $\delta\langle P^{\perp}, \Gamma\rangle$. Therefore we can proceed by induction hypothesis.

—One of $A\{\ \}$ and $B\{\ \}$ has a $\otimes$ as root and the other is just $\{\ \}$. Without loss of generality, $\Gamma = S[a^{\perp} \parr (B'\{a\} \otimes B'')]$. Then, by Lemma 6.8, $P \overset{\nu}{\rhd} S([a^{\perp} \parr B'\{a\}] \otimes B'')$, is also correct. We can apply the ls-rule and proceed by induction hypothesis.

—If $A\{\ \}$ and $B\{\ \}$ are both just $\{\ \}$, i.e., $\Gamma = S[a^{\perp} \parr a]$, then do nothing and pick another pair of dual atoms.

We continue until we cannot proceed any further by applying these cases. This means, all pairs of dual atoms in $\ell\Gamma$ are in a situation as in the last case above. Now observe that a formula is the negation of a linking formula iff it is generated by the grammar

$$\mathscr{N} ::= 1 \mid [\mathscr{A}^{\perp} \parr \mathscr{A}] \mid [\bot \parr \mathscr{N}] \mid (\mathscr{N} \otimes \mathscr{N}) \mid \forall \mathscr{A}. \mathscr{N}$$

Consequently, the only thing that remains to do is to bring the all $\bot$ to the left-hand side of a $\parr$. This can be done in a similar fashion as we brought pairs $[a^{\perp} \parr a]$ together, by applying

$\alpha\downarrow, \sigma\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow$. This makes Γ the negation of a linking. (Because of well-nestedness, there can be no $\exists$-nodes left.) Let us call this linking formula P'. Now we have a proof graph $P \triangleright P'^{\perp}$. By Lemma 6.3 we have $P' \lesssim P$.

It remains to remove all instances of $\mathsf{v}\downarrow$, which is done as in the proof of Theorem 3.4. □

We can now directly translate between deep inference proofs and proof graphs. More precisely, we can translate a $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ proof into a pre-proof graph by first decomposing it via Theorem 3.4 and then applying Lemmas 6.1, 6.4, and 4.4. Let us call a pre-proof graph *DI-sequentializable* if is obtained in this way from a $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ proof.

Theorem 6.9. Every DI-sequentializable pre-proof graph is a proof graph.

Proof. Theorem 3.4 decomposes a $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ proof into three parts, which correspond via Lemmas 6.1, 6.4, and 4.4 to the linking, the simple proof graph, and the expansion tree, respectively, of a proof graph $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ with $P^{\perp} = A$ and $\lceil \Gamma \blacktriangleleft \sigma \rceil = B$ and $\lfloor \Gamma \blacktriangleleft \sigma \rfloor = C$. □

By the method presented in (Straßburger, 2011), it is also possible to translate a $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ directly into a proof graph without prior decomposition. However, the decomposition is the key for the translation from proof graphs back into $\mathsf{MLL2}_{\mathsf{DI}\downarrow}$ proofs (i.e., "sequentialization" into the calculus of structures):

Theorem 6.10. If $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ is correct, then there is a $P' \lesssim P$, such that $P' \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ is DI-sequentializable.

Proof. Lemmas 6.1, 6.4, and 4.4 give us for a $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ the derivation on the left below:

$$
\begin{array}{c}
1 \\
\{\mathsf{ai}\downarrow, \bot\downarrow, 1\downarrow, \mathsf{e}\downarrow\} \Big\| \mathscr{D}'_1 \\
P'^{\perp} \\
\{\alpha\downarrow, \sigma\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow\} \Big\| \mathscr{D}'_2 \\
\lceil \Gamma \blacktriangleleft s \rceil \\
\{\mathsf{n}\downarrow, \mathsf{f}\downarrow\} \Big\| \mathscr{D}_3 \\
\lfloor \Gamma \blacktriangleleft s \rfloor
\end{array}
\qquad\qquad
\begin{array}{c}
1 \\
\{\mathsf{ai}\downarrow, \bot\downarrow, 1\downarrow, \mathsf{e}\downarrow\} \Big\| \mathscr{D}_1 \\
P^{\perp} \\
\{\alpha\uparrow, \sigma\uparrow, \alpha\downarrow, \sigma\downarrow, \mathsf{ls}, \mathsf{rs}, \mathsf{u}\downarrow\} \Big\| \mathscr{D}_2 \\
\lceil \Gamma \blacktriangleleft s \rceil \\
\{\mathsf{n}\downarrow, \mathsf{f}\downarrow\} \Big\| \mathscr{D}_3 \\
\lfloor \Gamma \blacktriangleleft s \rfloor
\end{array}
$$

where $P' \lesssim P$. Note that by Lemma 6.3, we also have derivation on the right above. □

Remark 6.11. It is also possible to translate directly between between sequent calculus proofs and proof graphs. But for the details the reader is referred to (Straßburger, 2017).

7. Cut elimination

In proof graphs, the cut is represented by a special connective $\oplus\!\!\!|$, such that whenever we have $A \,\mathbin{\oplus\!\!\!|}\, B$ in $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$, then we must have $\lfloor A \blacktriangleleft \sigma \rfloor = \lfloor B \blacktriangleleft \sigma \rfloor^{\perp}$.[‖] Morally, a $\oplus\!\!\!|$ may occur only at the root of a formula in Γ. However, due to well-nestedness we must allow cuts to have $\exists$-nodes

[‖] Note that it does not mean $A = B^{\perp}$, because Γ is expanded.

as ancestors. Then the $\obar$ is treated in the correctness criterion in exactly the same way as the $\otimes$, and sequentialization does also hold for proof graphs with cut.

As already discussed in (Lamarche and Straßburger, 2006), we need to work with an equivalence relation on proof graphs, because of the presence of the multiplicative units. This is a consequence of the PSPACE-completeness of proof equivalence in MLL (Heijltjes and Houston, 2014).

Definition 7.1. Let $\sim$ be the smallest equivalence relation on the set of proof graphs satisfying

$$\begin{array}{rcl} P[Q \parr R] \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma & \sim & P[R \parr Q] \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma \\ P[[Q \parr R] \parr S] \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma & \sim & P[Q \parr [R \parr S]] \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma \\ P(1 \otimes (1 \otimes Q)) \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma & \sim & P(1 \otimes (1 \otimes Q)) \stackrel{\nu'}{\triangleright} \Gamma \blacktriangleleft \sigma \\ P(1 \otimes [Q \parr R]) \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma & \sim & P[(1 \otimes Q) \parr R] \stackrel{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma \\ P(1 \otimes \exists a.Q) \stackrel{\nu}{\triangleright} \Gamma\{\bot\} \blacktriangleleft \sigma & \sim & P\{\exists a.(1 \otimes Q)\} \stackrel{\nu}{\triangleright} \Gamma\{\Exists a.\bot\} \blacktriangleleft \sigma \end{array}$$

where in the third line ν' is obtained from ν by exchanging the pre-images of the two 1s. In all other equations the bijection ν does not change. In the last line ν must match the 1 and $\bot$. A *proof net* is an equivalence class of $\sim$.

The first two equations in Definition 7.1 are simply associativity and commutativity of $\parr$ inside the linking. The third is a version of associativity of $\otimes$. The fourth equation could destroy multiplicative correctness, but since we defined $\sim$ only on proof graphs we do not need to worry about that.[††] The last equation says that a $\bot$ can freely tunnel through the borders of a box. Let us emphasize that this quotienting via an equivalence is due to the multiplicative units. If one wishes to use a system without units, one could completely dispose the equivalence by using n-ary $\parr$s in the linking.

The cut reduction relation $\leadsto$ is defined on pre-proof graphs as shown in Figures 7 and 8. The reductions not involving quantifiers are exactly as shown in (Lamarche and Straßburger, 2006). If we encounter a cut between two binary connectives, then we replace $[A \parr B] \obar (C \otimes D)$ by two smaller cuts $A \obar C$ and $B \obar D$. Note that if $\lfloor [A \parr B] \blacktriangleleft \sigma \rfloor = \lfloor (C \otimes D) \blacktriangleleft \sigma \rfloor^{\bot}$ then $\lfloor A \blacktriangleleft \sigma \rfloor = \lfloor C \blacktriangleleft \sigma \rfloor^{\bot}$ and $\lfloor B \blacktriangleleft \sigma \rfloor = \lfloor D \blacktriangleleft \sigma \rfloor^{\bot}$. If we have an atomic cut $a^{\bot} \obar a$, then we must have in P two "axiom links" $(a^{\bot} \otimes a)$, which are by the leaf mapping ν attached to the two atoms in the cut. It was shown in (Lamarche and Straßburger, 2006) that the two pairs $(a^{\bot} \otimes a)$ can, under the equivalence relation in Definition 7.1, be brought next to each other such that P has $[(a \otimes a^{\bot}) \parr (a \otimes a^{\bot})]$ as subformula. We can replace this by a single $(a^{\bot} \otimes a)$ and remove the cut. If we encounter a cut $1 \obar \bot$ on the units, we must have in the linking a corresponding $\bot$ and a subformula $(1 \otimes Q)$, which can (for the same reasons as for the atomic cut) be brought together, such that we have in P a subformula $[\bot \parr (1 \otimes Q)]$. We replace this by Q and remove the cut.

Let us now consider the cuts that involve the quantifiers. There are three cases, one for each of $\boldsymbol{\exists}$, $\Exists$, and $\exists$. The first two correspond to the ones in (Girard, 1987). The third one does not

[††] In (Lamarche and Straßburger, 2006) the relation $\sim$ is defined on pre-proof graphs, and therefore a side condition had to be given to that equation (see also (Hughes, 2005)).

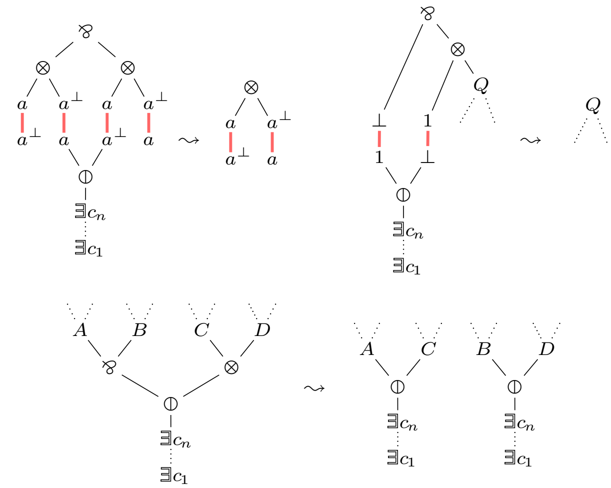
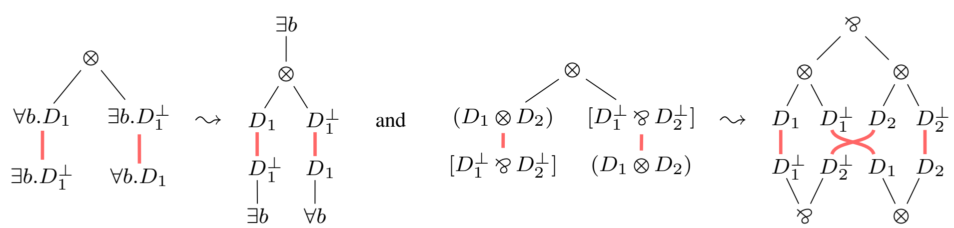


Fig. 7. Cut reduction for MLL2 proof nets (Part 1)

appear in (Girard, 1987) because there is never a $\exists$-node created when a sequent calculus proof is translated into a proof net.

If one of the cut formulas is an $\boldsymbol{\exists}$-node, then the other must be an $\forall$, which quantifies the same variable, say we have $\boldsymbol{\exists} a.A \obar \forall a.B$. Then we pick a stretching edge starting from $\boldsymbol{\exists} a.A$. Let C be the node where it ends and let $D = \lfloor C \blacktriangleleft \sigma \rfloor$. Note that by Condition 4.3-1, D is independent from the choice of the edge in case there are many of them. (If there are only negative edges, then let $D = \lfloor C \blacktriangleleft \sigma \rfloor^{\perp}$. If there are no stretching edges at all, then let $D = a$. Now we can inside the box of $\forall a.B$ substitute a everywhere by D. Then we remove all the doors of the $\forall a.B$-box and replace the cut by $A \obar B$. There are two subtleties involved in this case. First, "removing a door" means for a $\mathbb{\exists}$ that the node is removed, but for and $\exists$, it means that the node is replaced by an $\boldsymbol{\exists}$ and a stretching edge is added for every a and $a^{\perp}$ bound by the $\exists$-node to be removed. Second, by substituting a with D we get "axiom links" which are not atomic anymore, but it is straightforward to make them atomic again: one proceeds by structural induction on D, and the two reduction cases are



If one of the two cut formulas is a $\mathbb{\exists}$-node, then the other one can be anything. Say, we have $\mathbb{\exists} a.A \obar B$. Let eB be the *empire* of B, i.e, largest sub-proof graph of $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ that has B as a conclusion. Let $B_1, \ldots, B_n$ be the other doors of eB inside Γ, and let R be the door of eB in P. If eB has more than one root-node inside the linking P, then we can rearrange the $\parr$-nodes in P via the equivalence in 7.1 such that eB has a single $\parr$-root in P. Furthermore, as in the case of the atomic cut we can use the equivalence in 7.1 to get in P a subformula $[\exists a.Q \parr R]$

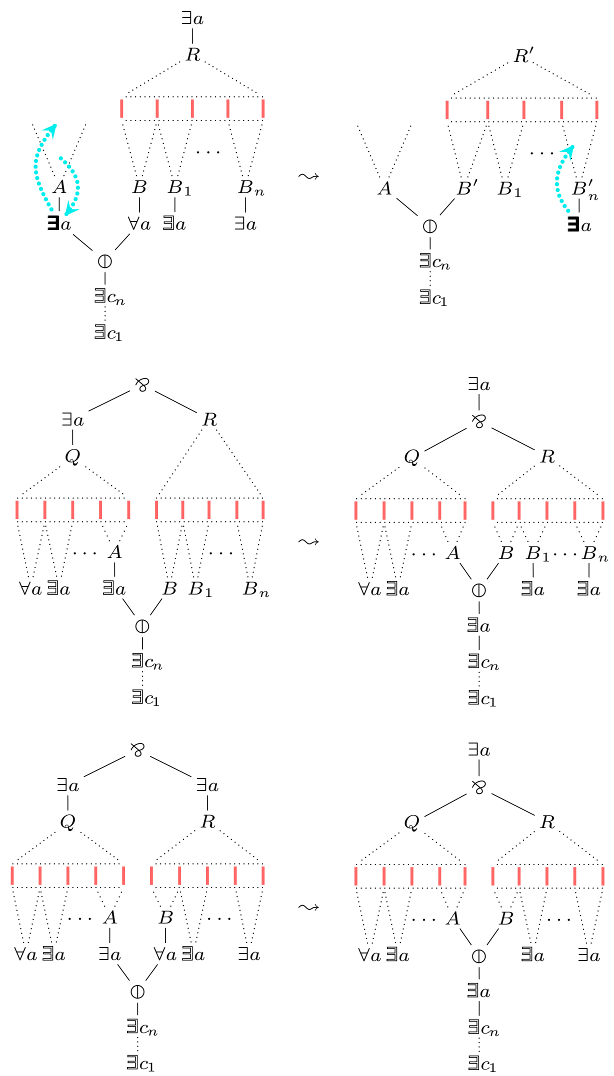


Fig. 8. Cut reduction for MLL2 proof nets

where $\exists a.Q$ is the partner of $\exists a.A$. Now we replace on P the formula $[\exists a.Q \mathbin{⅋} R]$ by $\exists a.[Q \mathbin{⅋} R]$ and in Γ the formulas $B_1, \ldots, B_n$ by $\exists a.B_1, \ldots, \exists a.B_n$. Put in plain words, we have pulled the whole empire of B inside the box of $\exists a.A$. But now we have a little problem: Morally, we should replace the cut $\exists a.A \oplus B$ by $A \oplus B$; the cut is also pulled inside the box. But by this we would break our correctness criterion, namely, the same-depth-condition 5.3-1. To solve this problem, we allow cut-nodes to have $\exists$-nodes as ancestors, and we replace the cut $\exists a.A \oplus B$ by $\exists a.(A \oplus B)$. Note that this does not cause problems for the other cut reduction steps because we can just keep all $\exists$-ancestors when we replace a cut by a smaller one.

Finally, there is the cut between an ordinary $\exists$-node and a $\forall$-node, say $\exists a.A \oplus \forall a.B$. Then we do not pull the whole empire of $\forall a.B$ inside the box of $\exists a.A$ but only the $\forall a.B$-box. This is the same as merging the two boxes into one. Formally, let $\exists a.Q$ and $\exists a.R$ be the partners of $\exists a.A$ and $\forall a.B$, respectively. Again, for the same reasons as in the case of the atomic cut, we can assume that we have the configuration $[\exists a.Q \parr \exists a.R]$ in P, which we replace by $\exists a.[Q \parr R]$. The cut is replaced by $\boldsymbol{\exists} a.(A \obar B)$.

This cut reduction relation is defined *a priori* only on pre-proof graphs. For a pre-proof graph $P \overset{\nu}{\triangleright} \Gamma \blacktriangleleft \sigma$ and a cut $A \obar B$ in Γ, we say the cut is *ready*, if the cut can immediately be reduced without further modification of P. We now can show the following:

Theorem 7.2. The cut reduction relation preserves correctness and is well-defined on proof nets.

Proof. That correctness is preserved follows immediately from inspecting the six cases. To show that cut reduction is well-defined on proof nets we need to verify the following two facts:

—Whenever the same cut is reduced in two different representations of the same proof net, then the two results also represent the same proof net.

—Whenever there is a cut in a proof net, then this cut can be reduced, i.e., there is a representation to which the corresponding reduction step in Figures 7 and 8 can be applied.

For the first statement, it suffices to observe that whenever one of the basic equivalence steps in Definition 7.1 can be performed in the non-reduced net, then the same step can be performed in the reduced net or is vacuous in the reduced net. For the second statement we have to make a case analysis on the type of cut: If the cut is $[A \parr B] \obar (C \otimes D)$ or $\boldsymbol{\exists} a.A \obar \forall a.B$, then it is trivial because these cuts are always ready. Let us now consider a cut $\exists a.A \obar \forall a.B$. Clearly, the two boxes of which $\exists a$ and $\forall a$ are doors each have a single door $\exists a$ in P, and their first common ancestor is a $\parr$ (because of the acyclicity condition). Therefore, the linking is of the shape $P[S_1\{\exists a.Q\} \parr S_2\{\exists a.R\}]$ for some contexts $S_1\{\ \}$ and $S_2\{\ \}$. Now we proceed by induction on the size of $S_1\{\ \}$ and $S_2\{\ \}$ and make a case analysis on their root-nodes:[‡‡]

—Both contexts are empty. In this case the linking has the desired shape, and we are done.

—One of them has a $\parr$-root. In this case we apply associativity of the $\parr$ and proceed by induction hypothesis.

—One of them has an $\exists$-node as root. This is impossible because it would violate the well-nested condition.

—One of them has a $\otimes$-root, and the other context is empty. Without loss of generality, the linking is of the shape $P[(1 \otimes S'_1\{\exists a.Q\}) \parr \exists a.R]$. We claim, that in this case the correctness is preserved if we replace the linking by $P(1 \otimes [S'_1\{\exists a.Q\} \parr \exists a.R])$. We leave the proof of this claim to the reader because it is very similar to the proof of Lemma 6.8. Hence, we can proceed by induction hypothesis.

—Both contexts have a $\otimes$-root. Then the linking is of the shape

$$P[(1 \otimes S'_1\{\exists a.Q\}) \parr (1 \otimes S'_2\{\exists a.R\})] \quad .$$

Now we claim that we can replace this linking with one of

$$P(1 \otimes [S'_1\{\exists a.Q\} \parr (1 \otimes S'_2\{\exists a.R\})]) \quad \text{and} \quad P(1 \otimes [(1 \otimes S'_1\{\exists a.Q\}) \parr S'_2\{\exists a.R\}])$$

[‡‡] Note the similarity to the proof of Lemma 6.4.

without destroying correctness. Again, we leave the proof to the reader because it is almost the same as the proof of Lemma 6.6. As before, we can proceed by induction hypothesis.

For a cut $\exists a.A \oplus B$ we proceed similarly. The only difference is that we first have to apply associativity and commutativity of $\parr$ to bring the proof graph in a form where the empire eB has a single root R in the linking. For cuts $a \oplus a^\perp$ and $1 \oplus \perp$ we can also proceed similarly. □

The main results of this section is now:

Theorem 7.3. The cut reduction relation $\leadsto$ is terminating and confluent.

Proof. Termination has already been shown in (Girard, 1987), and we will not repeat it here. For showing confluence it suffices to show local confluence. We will do this first for proof graphs. Suppose we have two cuts which are ready in a given proof graph. We claim that the result of reducing them is independent from the order of the reduction. There is only one critical pair, since the only possibility for overlapping redexes is when one cut is $\boldsymbol{\exists} a.A \oplus \forall a.B$ and the other is $\exists a.C \oplus \forall a.D$ and the formulas $\forall a.B$ and $\exists a.C$ are doors of the same box. If we reduce first the cut $\boldsymbol{\exists} a.A \oplus \forall a.B$, then we do first the substitution in the $\forall a.B$-box, remove its border, change the second cut to $\boldsymbol{\exists} a.C' \oplus \forall a.D$, and then do the same substitution in the $\forall a.D$-box and remove its border. If we reduce first the cut $\exists a.C \oplus \forall a.D$, then we merge the two boxes into one, and then do the substitution and remove the border of the box. Clearly, the result is the same in both cases. Hence, we have local confluence for the cut reduction on proof graphs. In the case of proof nets, it can happen that the two cuts are ready in two different representatives. With the method shown in the previous proof we can try to construct a representatives in which both cuts are ready. There are only two cases in which this fails. The first is when we have two atomic cuts using the same "axiom link". But then the result of reducing the two is a single axiom link, independent from the order. The second case is when we have two cuts $\exists a.A \oplus \forall a.B$ and $\exists a.C \oplus \forall a.D$ where $\forall a.B$ and $\exists a.C$ are doors of the same box. Here the result of reducing the two will be a big box which is the merge of all three boxes, independent of the order in which the two cuts are reduced. □

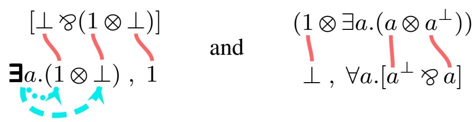
8. Some observations on the units

An important consequence of the last theorem is that we have a category of proof nets: the objects are (simple) formulas and a map $A \to B$ is a proof net with conclusion $\vdash A^\perp, B$. The composition of maps is defined by cut elimination. Unfortunately, we do not know much about this category, apart from the fact that it is *-autonomous (Lamarche and Straßburger, 2006). But there are some observations that we can make about the units, which can be expressed with the second-order quantifiers: $1 \equiv \forall a.[a^\perp \parr a]$ and $\perp \equiv \exists a.(a \otimes a^\perp)$. An interesting question to ask is whether these logical equivalences should be isomorphisms in the categorification of the logic. In the category of coherent spaces (Girard, 1987) they are, but in our category of proof nets they are not. This can be shown as follows. The two canonical maps $\forall a.[a^\perp \parr a] \to 1$ and

$1 \rightarrow \forall a.[a^\perp \parr a]$ are given in the sequent calculus by:

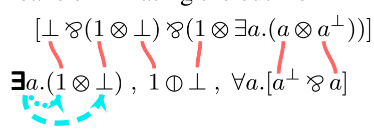
$$\exists\,\dfrac{\otimes\,\dfrac{1\,\dfrac{}{\vdash 1} \qquad \bot\,\dfrac{1\,\dfrac{}{\vdash 1}}{\vdash \bot, 1}}{\vdash (1\otimes\bot), 1}}{\vdash \exists a.(a\otimes a^\perp), 1} \quad \text{and} \quad \mathsf{weak}\,\dfrac{\forall\,\dfrac{\parr\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash a^\perp, a}}{\vdash [a^\perp \parr a]}}{\vdash \forall a.[a^\perp \parr a]}}{\vdash \bot, \forall a.[a^\perp \parr a]} \tag{14}$$

As proof nets they are given as follows:



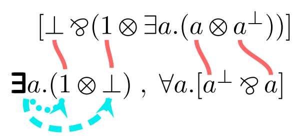
(15)

respectively. Composing them means eliminating the cut from



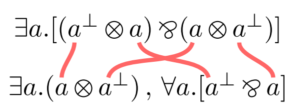
(16)

This yields



(17)

If the two maps in (15) where isos, the result (17) must be the same as the identity map $\forall a.[a^\perp \parr a] \rightarrow \forall a.[a^\perp \parr a]$ which is represented by the proof net



(18)

This is obviously not the case. Translating (17) and (18) back into the sequent calculus gives

$$\forall\,\dfrac{\exists\,\dfrac{\otimes\,\dfrac{1\,\dfrac{}{\vdash 1} \qquad \bot\,\dfrac{\parr\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash a^\perp, a}}{\vdash [a^\perp \parr a]}}{\vdash \bot, [a^\perp \parr a]}}{\vdash (1\otimes\bot), [a^\perp \parr a]}}{\vdash \exists a.(a\otimes a^\perp), [a^\perp \parr a]}}{\vdash \exists a.(a\otimes a^\perp), \forall a.[a^\perp \parr a]} \quad \text{and} \quad \forall\,\dfrac{\exists\,\dfrac{\parr\,\dfrac{\otimes\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash a, a^\perp} \qquad \mathsf{id}\,\dfrac{}{\vdash a^\perp, a}}{\vdash (a\otimes a^\perp), a^\perp, a}}{\vdash (a\otimes a^\perp), [a^\perp \parr a]}}{\vdash \exists a.(a\otimes a^\perp), [a^\perp \parr a]}}{\vdash \exists a.(a\otimes a^\perp), \forall a.[a^\perp \parr a]} \tag{19}$$

respectively.

A similar situation occurs with the additive units 0 and $\top$. They can be expressed with second-order quantifiers as follows: $0 \equiv \forall a.a$ and $\top \equiv \exists a.a$. Since we do not have 0 and $\top$ in the language, we cannot check whether we have these isos in our category. However, since 0 and $\top$ are commonly understood as initial and terminal objects of the category of proofs, we could ask whether $\forall a.a$ and $\exists a.a$ have this property: We clearly have a canonical proof for $\forall a.a \rightarrow A$ for every formula A (simply instantiate a with A), but it is not unique for all A. for example, we could prove the sequent $\vdash \exists a.a^\perp, (c\otimes[b \parr b^\perp])$ by substituting a with c. Nonetheless, one could imagine an isomorphism $0 \cong \forall a.a$ in a version of our proof nets which is extended with additives and exponentials. However, in this case 0 would not be initial.

9. Conclusions

In this paper we have investigated the relation between three different ways of presenting proofs in MLL2. First, in the sequent calculus, second, in the calculus of structures, and third, via proof graphs and expansion trees, and we have shown how these three presentations can be translated into each other. The main open question is now whether the identifications on proofs made by proof nets (i.e., equivalence classes of proof graphs) is the "right one". The observations in Section 8 show that the last word on this issue is not yet spoken. It would be important, to find independent (category theoretical) axiomatizations for the proof identity in MLL2, based on purely algebraic grounds. Then one could compare this algebraic notion of proof identity for MLL2 with the syntactic one based on proof nets.

A detailed comparison of this work to Girard's proof nets (Girard, 1987; Girard, 1990) can be found in (Straßburger, 2009; Straßburger, 2017).

Another direction for future research is the question how our method scales to larger fragments of linear logic. This concerns not only the exponentials and the additives (Hughes and van Glabbeek, 2003; Heijltjes and Hughes, 2015) but also higher-order linear logic.

References

Aler Tubella, A. (2016). *A study of normalisation through subatomic logic*. PhD thesis, University of Bath.

Andrews, P. B. (1976). Refutations by matings. *IEEE Transactions on Computers*, C-25:801–807.

Bellin, G. and van de Wiele, J. (1995). Subnets of proof-nets in MLL$^-$. In Girard, J.-Y., Lafont, Y., and Regnier, L., editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 249–270. Cambridge University Press.

Blute, R., Cockett, R., Seely, R., and Trimble, T. (1996). Natural deduction and coherence for weakly distributive categories. *J. of Pure and Applied Algebra*, 113:229–296.

Brünnler, K. (2003). *Deep Inference and Symmetry for Classical Proofs*. PhD thesis, Technische Universität Dresden.

Brünnler, K. and Tiu, A. F. (2001). A local system for classical logic. In Nieuwenhuis, R. and Voronkov, A., editors, *LPAR 2001*, volume 2250 of *LNAI*, pages 347–361. Springer.

Buss, S. R. (1991). The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53:72–102.

Chaudhuri, K., Guenot, N., and Straßburger, L. (2011). The focused calculus of structures. In Bezem, M., editor, *CSL'11*, volume 12 of *LIPIcs*, pages 159–173. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik.

Danos, V. and Regnier, L. (1989). The structure of multiplicatives. *Annals of Mathematical Logic*, 28:181–203.

Devarajan, H., Hughes, D., Plotkin, G., and Pratt, V. R. (1999). Full completeness of the multiplicative linear logic of Chu spaces. In *14th IEEE Symposium on Logic in Computer Science (LICS 1999)*.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science*, 50:1–102.

Girard, J.-Y. (1990). Quantifiers in linear logic II. Preépublication de l'Equipe de Logique, Université Paris VII, Nr. 19.

Guglielmi, A. (2007). A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64.

Guglielmi, A. and Gundersen, T. (2008). Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36.

Guglielmi, A. and Straßburger, L. (2001). Non-commutativity and MELL in the calculus of structures. In Fribourg, L., editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag.

Guglielmi, A. and Straßburger, L. (2011). A system of interaction and structure V: the exponentials and splitting. *Mathematical Structures in Computer Science*, 21(3):563–584.

Heijltjes, W. and Houston, R. (2014). No proof nets for MLL with units: proof equivalence in MLL is pspace-complete. In Henzinger, T. A. and Miller, D., editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 50:1–50:10. ACM.

Heijltjes, W. and Hughes, D. J. D. (2015). Complexity bounds for sum-product logic via additive proof nets and petri nets. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015*, pages 80–91. IEEE Computer Society.

Hughes, D. (2005). Simple free star-autonomous categories and full coherence. Preprint.

Hughes, D. and van Glabbeek, R. (2003). Proof nets for unit-free multiplicative-additive linear logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 1–10.

Lafont, Y. (1988). *Logique, Catégories et Machines*. PhD thesis, Université Paris 7.

Lafont, Y. (1995). From proof nets to interaction nets. In Girard, J.-Y., Lafont, Y., and Regnier, L., editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 225–247. Cambridge University Press.

Lamarche, F. and Straßburger, L. (2006). From proof nets to the free *-autonomous category. *Logical Methods in Computer Science*, 2(4:3):1–44.

Lyaletski, A. and Konev, B. (2006). On herbrand's theorem for intuitionistic logic. In Fisher, M., van der Hoek, W., Konev, B., and Lisitsa, A., editors, *Logics in Artificial Intelligence: 10th European Conference, JELIA 2006 Liverpool, UK*, pages 293–305. Springer Berlin Heidelberg.

Miller, D. (1987). A compact representation of proofs. *Studia Logica*, 46(4):347–370.

Retoré, C. (1993). *Réseaux et Séquents Ordonnés*. PhD thesis, Université Paris VII.

Straßburger, L. (2003). *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden.

Straßburger, L. (2009). Some observations on the proof theory of second order propositional multiplicative linear logic. In Curien, P.-L., editor, *Typed Lambda Calculi and Applications, TLCA'09*, volume 5608 of *Lecture Notes in Computer Science*, pages 309–324. Springer.

Straßburger, L. (2011). From deep inference to proof nets via cut elimination. *Journal of Logic and Computation*, 21(4):589–624.

Straßburger, L. (2017). Deep Inference, Expansion Trees, and Proof Graphs for Second Order Propositional Multiplicative Linear Logic. Research Report RR-9071, Inria Saclay.

Straßburger, L. and Guglielmi, A. (2011). A system of interaction and structure IV: The exponentials and decomposition. *ACM Trans. Comput. Log.*, 12(4):23.

Straßburger, L. and Lamarche, F. (2004). On proof nets for multiplicative linear logic with units. In Marcinkowski, J. and Tarlecki, A., editors, *Computer Science Logic, CSL 2004*, volume 3210 of *LNCS*, pages 145–159. Springer-Verlag.