Simple multiplicative proof nets with units

DOMINIC J. D. HUGHES*
Stanford University

Abstract. This paper presents a simple notion of proof net for multiplicative linear logic with units. Cut elimination is direct and strongly normalising, in contrast to previous approaches which resorted to moving jumps (attachments) of par units during normalisation. Composition in the resulting category of proof nets is simply path composition: all of the dynamics happens in $\mathbf{GoI}(\mathsf{Setp})$, the geometry-of-interaction construction applied to the category of sets and partial functions.

Keywords: multiplicative linear logic, units, proof nets, geometry of interaction.

AMS subject classification: 3B47 Substructural logics, 03F52 Linear logic.

1 Introduction

Here is a passage from Girard's *Proof Nets: the Parallel Syntax for Proof Theory* [Gir96, §A.2][1]:

> There are two multiplicative neutrals, 1 and $\bot$, and two rules, the axiom $\vdash 1$ and the weakening rule: from $\vdash \Gamma$, deduce $\vdash \Gamma, \bot$. Both rules are handled by means of links with one conclusion and no premise; however, $\bot$-links are treated like 0-ary $?$-links, *i.e.*, they must be given a default jump. Sequentialisation is immediate.
>
> At first sight, cut elimination is unproblematic: replace a cut between conclusions 1 and $\bot$ of zero-ary links with…nothing. But we notice a new problem, namely that a cut formula A can be the default jump of a $\bot$-link L, and we must therefore propose another jump for L. Usually one of the premises of the link with conclusion A works (or the jump of L' if A is the conclusion of a $\bot$-link) works. Worse, this new jump is by no means natural (if A is $B \otimes C$, the new jump can either be B or C), which is quite unpleasant. As far as we know, the only solution consists in declaring that jumps are not part of the proof-net, but rather some control structure. It is then enough to show that at least one choice of default jump is possible. This is not a very elegant solution: we are indeed working with equivalence classes of proof nets and if we want to be rigorous we shall have to endlessly check that such and such operation does not depend on the choice of default jumps.

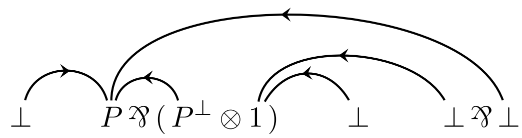
This paper presents a very simple solution: define a multiplicative proof net with units (neutrals) as a function from negative to positive formula leaves, satisfying the usual correctness criterion [Gir87, DR89]. Cut elimination on binary connectives is then trivial (as usual in the unit-free setting), and we have a direct strong normalisation by standard path composition: all of the dynamics happens in $\mathbf{GoI}(\mathsf{Setp})$, the geometry-of-interaction or feedback construction [Gir89, JSV96, Abr96] applied to the category of sets and partial functions.

The novelty here is not the directed edges between negative and positive leaves, an idea which goes back to the origins of linear logic [Gir87] and Kelly-MacLane graphs [KM71]. The key contribution is the simply defined, strongly normalising cut elimination, over $\mathbf{GoI}(\mathsf{Setp})$.

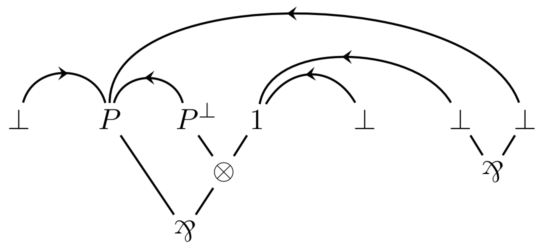
*Computer Science Department, Stanford University, CA 94305. Email dominic@theory.stanford.edu, Tel. +1 415 242-1858, Fax +1 650 725-4671.

[1] Similar remarks are in the earlier *Linear Logic: A Survey* [Gir93, §3.6].

The nets. Here is a simple example of a cut-free proof net on a four-formula sequent:

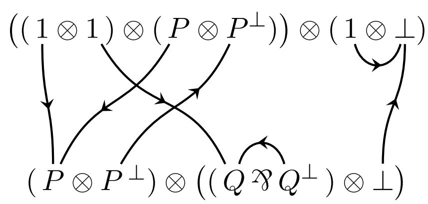


The graph of the function from negative to positive leaves is shown by the directed edges. Note that all four switchings are trees. This is easier to see if we show the parse trees:

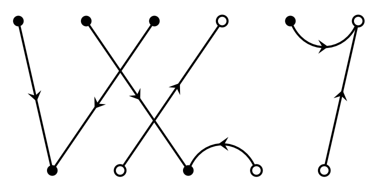


As with the unit-free case [Gue99, MO00], correctness can be checked in linear time (see Section 6).

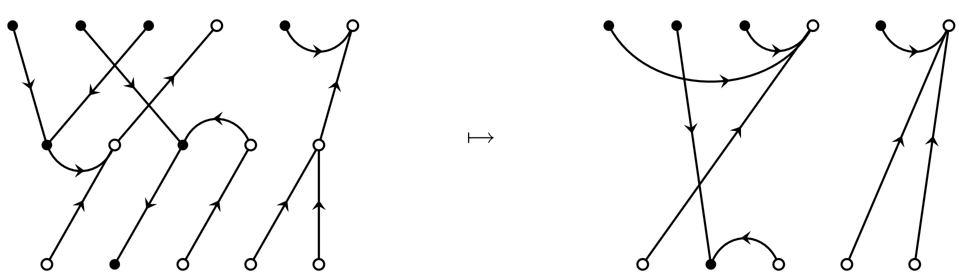
GoI dynamics. MLL formulas and proof nets form a category with a morphism $A \to B$ a cut-free proof net on $\vdash A^\bot, B$. For example,



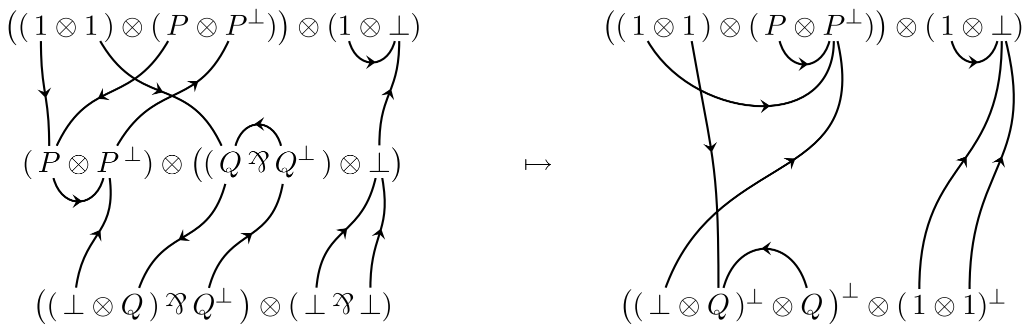
is a morphism from the upper formula to the lower formula. (We suppress the negation on the input/upper formula, flipping polarity, so tensors are switched in the input.) The underlying $\mathbf{GoI}(\mathsf{Setp})$ morphism is:



An object of $\mathbf{GoI}(\mathsf{Setp})$ is a signed set S, whose elements we shall call *leaves*, and a morphism $S \to T$ is a partial function from negative leaves to positive leaves (polarity flipped on the input side). Composition is standard path composition, *e.g.*



which provides composition (turbo cut elimination) in the category of proof nets, *e.g.*



is the path composition of the previous $\mathbf{GoI}$ diagram. This provides a simple solution to the problems articulated by Girard above.

Sliced-GoI composition for MALL nets. Section 7 continues the $\mathbf{GoI}$ theme, and shows how composition (turbo cut elimination) of MALL proof nets [HG03, HG05] can be viewed as occurring in a sliced variant of $\mathbf{GoI}(\mathsf{Setp})$: it presents a faithful functor from the category of MALL proof nets to $\mathbf{Matr}(\mathbf{GoI}(\mathsf{Setp}))$, where $\mathbf{Matr}$ is a standard categorical biproduct construction.

Related work. Proof nets with units are in [BCST96] and [LS04]. Neither solves the problems in Girard's quote: each suffers from the need to move $\perp$-jumps during elimination, so one is lumbered once again with equivalence classes. The cut-free one-sided MLL proof nets in [BCST96] are the cut-free proof nets described in Girard's quote in a circuit/wire notation, with an additional ordering on $\perp$-jumps: see Section 8.1. The paper [LS04] defines a cut-free proof net on a sequent $\vdash \Gamma$ as a separate MLL formula Θ whose leaves from left-to-right are a permutation of those of Γ. The $\perp$-jumps and axiom links are thus enveloped in an additional syntactic layer Θ: see Section 8.2. The proof nets of [MO03] for intuitionistic multiplicative linear logic with units (based on essential nets [Lam94]) involve directed edges.

Work in progress quotients the nets presented in this paper by Trimble's *empire rewiring* [Tri94], which permits a $\perp$-jump target to move so long as correctness is not broken, to construct free star-autonomous categories for full coherence (*cf.* [BCST96, KO99, MO03, LS04]).

Acknowledgement. Thanks to Robin Houston for feedback.

2 Notation

By MLL we mean multiplicative linear logic with units [Gir87]. Formulas are built from literals (propositional variables $P, Q, \ldots$ and their duals $P^\perp$, $Q^\perp, \ldots$) and units/constants/neutrals 1 and $\perp$ by the binary connectives ***tensor*** $\otimes$ and ***par*** $\parr$. Negation $(-)^\perp$ extends to arbitrary formulas with $P^{\perp\perp} = P$ on propositional variables, $\perp^\perp = 1$, $1^\perp = \perp$, and de Morgan duality $(A \otimes B)^\perp = A^\perp \parr B^\perp$ and $(A \parr B)^\perp = A^\perp \otimes B^\perp$. An ***atom*** is a literal or unit. We identify a formula with its parse tree: a tree labelled with atoms at the leaves and connectives at internal vertices. A ***sequent*** is a non-empty disjoint union of formulas. Thus a sequent is a particular kind of labelled forest. We write comma for disjoint union. Sequents are proved using the following rules:

$$\frac{}{P, P^\perp}\,\mathsf{ax} \qquad \frac{\Gamma, A \quad A^\perp, \Delta}{\Gamma, \Delta}\,\mathsf{cut} \qquad \frac{}{1}\,1 \qquad \frac{\Gamma}{\Gamma, \perp}\,\perp \qquad \frac{\Gamma, A \quad B, \Delta}{\Gamma, A \otimes B, \Delta}\,\otimes \qquad \frac{\Gamma, A, B}{\Gamma, A \parr B}\,\parr$$

Here, and throughout this document, $P, Q, \ldots$ range over propositional variables, $A, B, \ldots$ over formulas, and $\Gamma, \Delta, \ldots$ over (possibly empty) disjoint unions of formulas. Without loss of generality we restrict the axiom rule to literals [Gir87]. The propositional variables $P, Q, \ldots$ and the unit 1 are ***positive***, and their duals $P^{\perp}, Q^{\perp}, \ldots$ and $\perp$ are ***negative***. A leaf of a formula is positive/negative according to its label. A ***cut pair*** $A \smile A^{\perp}$ is a disjoint union of complementary formulas A and $A^{\perp}$ together with an undirected edge, a ***cut***, between their roots. A ***cut sequent*** is a disjoint union of a sequent and zero or more cut pairs. A ***switching*** of a cut sequent is any subgraph obtained by deleting one of the two argument edges of each $\parr$ (see [DR89]). By an ***old proof net*** we mean a proof net for MLL with units as in Girard's quote in the Introduction; see [Dan90, Reg92, GSS92, Gir93, Gir96] for history and development. (An example of an old proof net is drawn in the next section.)

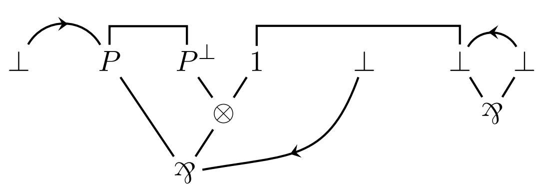
3 Proof nets

A ***leaf function*** on a cut sequent is a function from its negative leaves to its positive leaves. A ***proof net*** on a cut sequent Γ is a leaf function f on Γ satisfying:

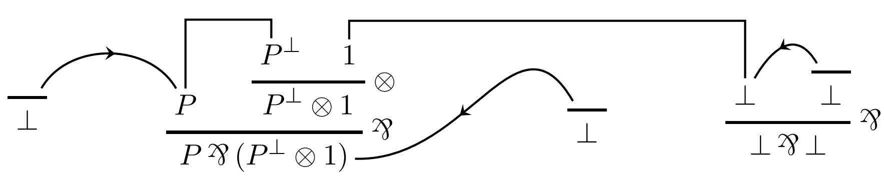
- MATCHING. For any propositional variable P, the restriction of f to P-labelled leaves is a bijection between the P-labelled leaves of Γ and the $P^{\perp}$-labelled leaves of Γ.

- SWITCHING. For any switching Γ' of Γ, the undirected graph obtained by adding the edges of f to Γ' is a tree (acyclic and connected).

See page 2 for an example. This definition amounts to a restricted case of an old proof net: restrict $\perp$-jumps to target positive leaves and reject unit axiom links (use $\perp \to 1$ jumps instead). In addition, we orient all axiom links from negative to positive. Stating this the other way round, the above definition relaxes to the old definition thus: (a) on $\perp$-labelled leaves allow f to target any vertex (equivalently subformula) of Γ, not just a positive leaf, (b) distinguish between two kinds of edges from $\perp$ to 1 (jump *versus* axiom link), and (c) draw axiom links unoriented. Here is an example of an old proof net:



which in original proof net notation is:



Axiom links are shown as three-segment straight edges, and jumps from $\perp$-links $\overline{\perp}$ are shown curved and directed.

Translation from a proof to a proof net is as usual, with a $\perp$-jump added at each $\perp$-rule, but now with choice of target restricted to positive atoms only. Note that well-definedness relies on the observation that every provable MLL sequent contains a positive atom. The translation becomes deterministic upon marking a positive leaf in the conclusion of every $\bot$-rule. For example, each of the following marked proofs translates (deterministically) into the proof net on page 2:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P\,,\,P^{\perp}}\,\mathsf{ax}}{\bot,\,P\,,\,P^{\perp}}\,\bot \qquad \dfrac{}{1\,,\,\bot}\,\mathsf{ax}}{\bot,\,P\,,\,\,P^{\perp}\otimes 1\,\,,\,\bot}\,\otimes}{\bot,\,P\,,\,\,P^{\perp}\otimes \underline{1}\,\,,\,\bot,\,\bot}\,\bot}{\bot,\,P\parr(P^{\perp}\otimes 1),\,\bot,\,\bot}\,\parr}{\bot,\,\underline{P}\parr(P^{\perp}\otimes 1),\,\bot,\,\bot\,,\,\bot}\,\bot}{\bot,\,P\parr(P^{\perp}\otimes 1),\,\bot,\,\bot\parr\bot}\,\parr$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P\,,\,P^{\perp}}\,\mathsf{ax} \qquad \dfrac{\dfrac{\dfrac{}{1}\,1}{1\,,\,\bot}\,\bot}{1\,,\,\bot\,,\,\bot}\,\bot}{P\,,\,\,P^{\perp}\otimes 1\,\,,\,\bot,\,\bot}\,\otimes}{\underline{P}\,,\,\,P^{\perp}\otimes 1\,\,,\,\bot,\,\bot\,,\,\bot}\,\bot}{P\,,\,\,P^{\perp}\otimes 1\,\,,\,\bot,\,\bot\parr\bot}\,\parr}{P\parr(P^{\perp}\otimes 1),\,\bot,\,\bot\parr\bot}\,\parr}{\bot,\,\underline{P}\parr(P^{\perp}\otimes 1),\,\bot,\,\bot\parr\bot}\,\bot$$

Marks are shown by underlining; when a sequent has just one positive atom, we leave the mark implicit. (Downward tracking of $\bot$'s is vertical, except through the tensor rule.)

THEOREM 1 (SEQUENTIALISATION) *Every proof net is a translation of a proof.*

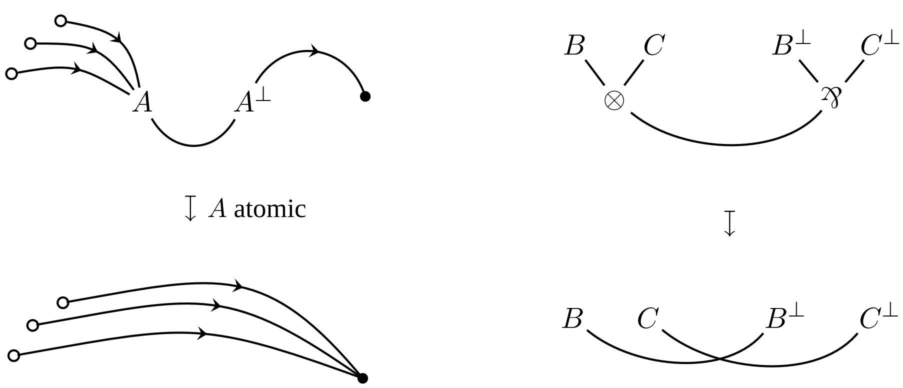
This is simply a restriction of the theorem for old proof nets. Correctness is verifiable in linear time (a simple corollary of the unit-free case [Gue99, MO00]): see Section 6.

4 Cut elimination

Let f be a proof net on the cut sequent $\Gamma, A \smallsmile A^{\perp}$. The result f' of ***eliminating*** the cut in $A \smallsmile A^{\perp}$ is:

- *Atom.* Suppose A is an atom. Without loss of generality, A is positive. Delete $A \smallsmile A^{\perp}$ and reset any f-edge to A to target $f(A^{\perp})$ instead.
- *Compound.* Suppose A is a compound formula. Without loss of generality $A = B \otimes C$ and $A^{\perp} = B^{\perp} \parr C^{\perp}$. Replace $A \smallsmile A^{\perp}$ by $B \smallsmile B^{\perp}, C \smallsmile C^{\perp}$. The leaves, and f, remain unchanged.

Schematically:

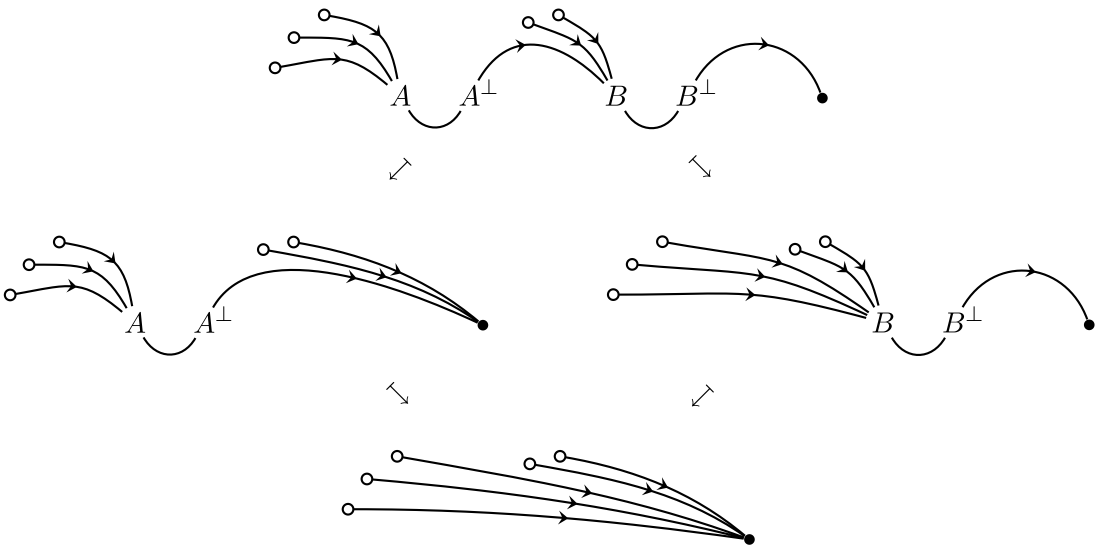


THEOREM 2 *Cut elimination is well-defined: eliminating a cut from a proof net yields a proof net.*

Proof. The atomic case is trivial, since switchings and cycles correspond before and after the elimination. The compound case is the same as the usual unit-free elimination [Gir87, DR89, Gir93]. □

PROPOSITION 1 *Cut elimination is locally confluent.*

Proof. The only non-trivial case is a pair of atomic eliminations. This case is clear from the following schematic involving two interacting atomic cut redexes $A \smile A^{\perp}$ and $B \smile B^{\perp}$.



□

THEOREM 3 *Cut elimination is strongly normalising.*

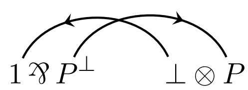
Proof. It is locally confluent, and eliminating a cut reduces the number of vertices of the cut sequent. □

Turbo cut elimination. As with standard unit-free MLL proof nets, normalisation can be completed in a single step. For l the i^{th} leaf of a formula A in a cut pair $A \smile A^{\perp}$, let $l^{\perp}$ denote the i^{th} leaf of $A^{\perp}$. The ***normal form*** of a cut sequent Γ is the sequent $|\Gamma|$ obtained by deleting all cut pairs. Given a proof net f on Γ, its ***normal form*** $|f|$ is the proof net on $|\Gamma|$ obtained by replacing every set of edges $\langle l_0, l_1\rangle, \langle l_1^{\perp}, l_2\rangle, \langle l_2^{\perp}, l_3\rangle, \ldots, \langle l_{n-1}^{\perp}, l_n\rangle$ in f in which only l_0 and l_n occur in $|\Gamma|$ by the single edge $\langle l_0, l_n\rangle$. By a simple induction on the number of vertices of cut sequents, $|f|$ is precisely the normal form of f under one-step cut elimination. (In particular, this implies $|f|$ is indeed a proof net.)

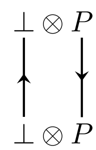
5 GoI dynamics

Cut elimination yields a category $\mathcal{N}$ of MLL proof nets. Objects are MLL formulas, and a morphism $A \to B$ is a proof net on the (cut-free) sequent $A^\perp, B$ (*cf.* [HG03, HG05], for example). The composite of $f : A \to B$ and $g : B \to C$ is the normal form of the proof net $f \cup g$ on $A^\perp, B \smile B^\perp, C$. Composition is associative because cut elimination is strongly normalising. The identity $A \to A$, a leaf function on $A^\perp, A$, has an edge between the i^{th} leaf of $A^\perp$ and the i^{th} leaf of A for each i, oriented from negative to positive.

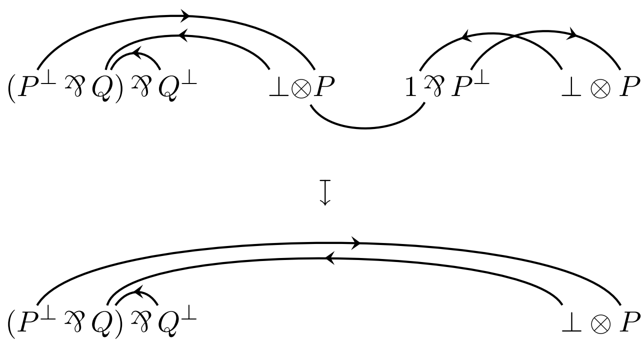
We generally draw $f : A \to B$ with A above B, and suppress the negation on A. For example, the identity $\perp \otimes P \to \perp \otimes P$



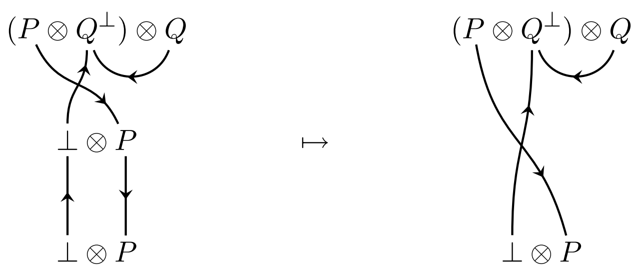
becomes



Similarly, a composition such as

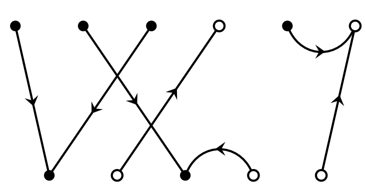


(involving the aforementioned identity $\perp \otimes P \to \perp \otimes P$) becomes:



A more interesting example of composition is on page 3 of the Introduction.

Underlying GoI category. The category $\mathbf{GoI}(\mathsf{Setp})$, the result of applying the geometry-of-interaction or feedback construction $\mathbf{GoI}$ [Gir89, JSV96, Abr96] to the category Setp of sets and partial functions, has the following structure. An object is a signed set S, whose elements we shall call *leaves* (each signed either *positive* or *negative*), and a morphism $S \to T$ is a ***partial leaf function***: a partial function from $S^+ + T^-$ to $S^- + T^+$, where $(-)^+$ (resp. $(-)^-$) restricts to positive (resp. negative) leaves. For example,

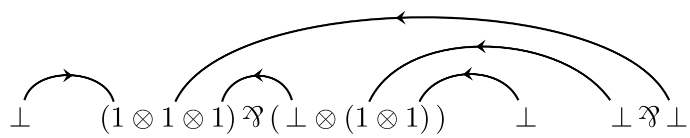


is a (total) morphism from the upper signed set (4 positive • and 2 negative ∘ leaves) to the lower one (2 positive and 3 negative leaves). Composition is simply (finite) path composition: for an example, see page 2 of the Introduction. Turbo cut elimination is the very same path composition, hence there is a forgetful (faithful) functor from the category $\mathcal{N}$ of MLL proof nets to $\mathbf{GoI}(\mathsf{Setp})$, extracting the leaves from a formula. Again, see the Introduction for examples.

6 Linear complexity of proof net correctness

THEOREM 4 (LINEAR COMPLEXITY) *Verification of proof net correctness is linear in the number of leaves: if f is a leaf function on a cut sequent Γ, then determining whether f is a proof net can be done in linear time in the number of leaves of Γ.*

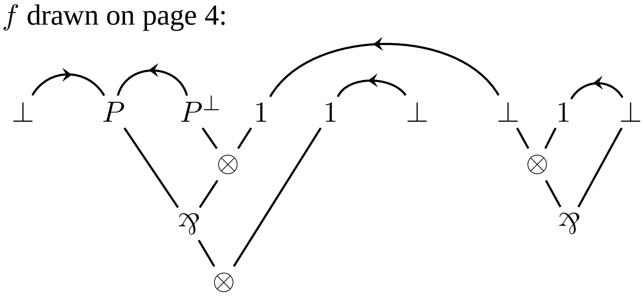
Proof. Verifying the MATCHING condition is clearly linear time. The SWITCHING condition is a simple corollary of the unit-free theorem [Gue99, MO00]: the function f determines a standard unit-free proof structure $\widehat{f}$ on $\widehat{\Gamma}$, as follows. First, replace every cut pair $A \smile A^\perp$ by $A \otimes A^\perp$. We may assume every positive leaf has an incoming f-edge: every literal does, by MATCHING; if the 1 of a subformula $A \otimes 1$ doesn't, replace $A \otimes 1$ by A; if the 1 of $A \parr 1$ doesn't, SWITCHING fails. Re-label each positive literal to 1 and each negative literal to $\bot$. Replace each 1 by 1^n where $n \geq 1$ is the number of f-edges targetting the 1, and 1^n denotes the tensor product of n copies of 1 (bracketed arbitrarily); re-target the n edges to the old 1 to each target a distinct new 1 of 1^n. Finally, view the symbols $\bot$ and 1 as complementary literals, so we have formed a standard proof structure $\widehat{f}$ on a cut-free, unit-free MLL sequent $\widehat{\Gamma}$. To clarify, here is $\widehat{f}$ for f the proof net on page 2:



By construction the original f on Γ is correct iff $\widehat{f}$ on $\widehat{\Gamma}$ is correct in the usual unit-free sense. The construction of $\widehat{f}$ is linear time in the number of leaves. □

COROLLARY 1 *The theorem above extends to old proof nets (*i.e.*, when f is a function from negative leaves to vertices of Γ, optionally with a differentiation between axiom links $\bot\ 1$ and jumps $\bot\ 1$).*

Proof. First, if differentiating, replace every axiom link $\bot\ 1$ by a jump $\bot\ 1$. Rewrite every compound subformula or negative leaf A targeted by a $\bot$-jump to $A \otimes 1$, and shift any $\bot$-jumps which targeted A to target the new 1 instead. This yields a function $\tilde{f}$ from negative leaves to positive leaves which is correct iff f is correct; apply the above theorem to $\tilde{f}$. To clarify, here is $\tilde{f}$ for the old proof net f drawn on page 4:



The construction $f \mapsto \tilde{f}$ is linear time in the number of leaves. □

7 Sliced GoI composition for MALL nets

Continuing the $\mathbf{GoI}$ theme, in this section we observe that composition (turbo cut elimination) of MALL proof nets [HG03, HG05] can be viewed as occurring in a 'sliced' variant of $\mathbf{GoI}(\mathsf{Setp})$. Specifically, we define a faithful functor

$$|-| \quad : \quad \mathsf{MALL} \quad \longrightarrow \quad \mathbf{Matr}(\mathbf{GoI}(\mathsf{Setp}))$$

where MALL is the category of MALL proof nets defined in [HG03, HG05] and $\mathbf{Matr}$ is a standard biproduct construction.[2]

A ***bag*** (or multiset) over a set or class X is a formal sum $\Sigma_{i\in I}\, x_i$ of members x_i of X for some finite indexing set I.[3] Write $\mathbf{Bag}(\mathbb{C})$ for the free commutative monoid enrichment of a category $\mathbb{C}$: objects are those of $\mathbb{C}$, a morphism $X \to Y$ in $\mathbf{Bag}(\mathbb{C})$ is a bag of morphisms $X \to Y$ in $\mathbb{C}$ (*i.e.*, a bag over the homset $\mathbb{C}(X, Y)$), and the composite of $\Sigma_{i\in I}\, f_i : X \to Y$ and $\Sigma_{j\in J}\, g_j : Y \to Z$ is pointwise, indexed by $I \times J$:

$$(\Sigma_{i\in I}\, f_i)\ ;\ (\Sigma_{j\in J}\, g_j) \quad = \quad \Sigma_{i\in I, j\in J}\,(f_i; g_j) \quad : \quad X \to Z$$

Recall the biproduct completion $\mathbf{Matr}(\mathbb{C})$ of a category $\mathbb{C}$ enriched over commutative monoids (*cf.* [Mac71, VIII Ex. 2.6]). An object of $\mathbf{Matr}(\mathbb{C})$ is a bag of objects of $\mathbb{C}$ (*i.e.*, a bag over the collection of objects of $\mathbb{C}$) and a morphism $\Sigma_{i\in I}\, A_i \;\to\; \Sigma_{j\in J}\, B_j$ is an $(I \times J)$-indexed bag of morphisms $\Sigma_{i\in I, j\in J}\, f_{ij}$ such that $f_{ij} : A_i \to B_j$ in $\mathbb{C}$, called a ***matrix***. Composition is by matrix multiplication with respect to the commutative monoid operation $\star$ in $\mathbb{C}$: the $\langle i, k\rangle^{\text{th}}$ element of the composite of $\Sigma_{i\in I, j\in J}\, f_{ij}$ and $\Sigma_{j\in J, k\in K}\, g_{jk}$ is $\bigstar_{j\in J}\,(f_{ij}; g_{jk})$ where $\bigstar$ denotes iterated $\star$.

If $\mathbb{C}$ does not come equipped with a commutative monoid enrichment, define $\mathbf{Matr}(\mathbb{C})$ as $\mathbf{Matr}(\mathbf{Bag}(\mathbb{C}))$, interposing free commutative monoid enrichment. Thus $\mathbf{Matr}(\mathbf{GoI}(\mathsf{Setp}))$ has the following compact closed structure with biproducts:

- Objects. An object is a bag (formal sum) $A \;=\; \Sigma_{i\in I}\, A_i$ of signed sets A_i, the ***slices*** of A.
- Morphisms. A morphism $\Sigma_{i\in I}\, A_i \;\to\; \Sigma_{j\in J}\, B_j$ is an $I \times J$-indexed matrix whose $\langle i, j\rangle^{\text{th}}$ element is a bag of partial leaf functions $A_i \to B_j$.
- Pointwise tensor: $\Sigma_{i\in I}\, A_i \;\otimes\; \Sigma_{j\in J}\, B_j \;=\; \Sigma_{i\in I, j\in J}\; A_i \otimes B_j$.
- Pointwise duality: $(\Sigma_{i\in I}\, A_i)^{\perp} \;=\; \Sigma_{i\in I}\,(A_i^{\perp})$.
- Biproduct: $\Sigma_{i\in I}\, A_i \;\oplus \Sigma_{j\in J}\, B_j$ is the formal sum $\Sigma_{i\in I}\, A_i \;+\; \Sigma_{j\in J}\, B_j$ (indexed by the disjoint union of I and J).

The faithful functor. Recall that an object of the category MALL of MALL proof nets is a MALL formula, generated from literals by the binary connectives $\otimes$ (tensor), $⅋$ (par), $\oplus$ (plus) and $\&$ (with) [HG03, HG05]. Henceforth identify a formula with its parse tree (a labelled binary tree with literals on leaves and connectives on internal nodes). Recall [*ibid.*] that an ***additive***

[2] One could just as well take in place of Setp either the category Rel of sets and binary relations or (since we do not consider units in this section) the category PInj of sets and partial injective functions.

[3] Formal sums are defined modulo index renaming, *i.e.*, $\Sigma_{i\in I}\, x_i$ and $\Sigma_{j\in J}\, y_j$ denote the same bag iff there exists a bijection $\widehat{(-)} : I \to J$ with $y_{\widehat{i}} = x_i \in X$.

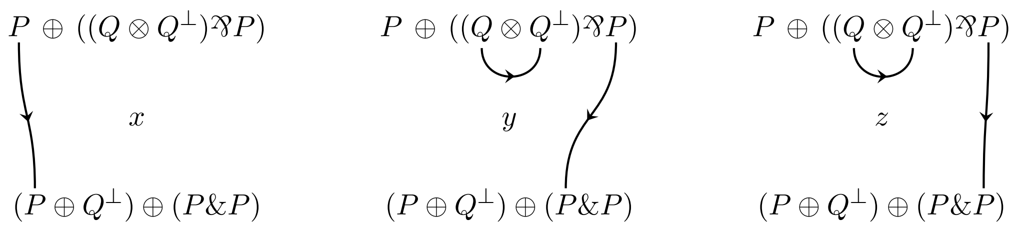
resolution of a formula A is any subtree (labelled subgraph) obtained by deleting one argument subtree of each additive connective ($\oplus$ or $\&$) of A. Define the faithful functor

$$|-| \quad : \quad \mathsf{MALL} \quad \longrightarrow \quad \mathbf{Matr}(\mathbf{GoI}(\mathsf{Setp}))$$

on an object (MALL formula) A as follows: $|A|$ comprises the signed sets underlying the additive resolutions of A. Formally, $|A| = \Sigma_{r \in R}\, \underline{r}$ where R is the set of additive resolutions of A and $\underline{r}$ denotes the underlying signed set of leaves of r. For example, if $A = P \oplus ((Q \otimes Q^{\perp}) \parr P)$ with leaves $a_1^+, a_2^+, a_3^-, a_4^+$ from left to right, then $|A| = \{a_1^+\} + \{a_2^+, a_3^-, a_4^+\}$, the formal sum of two signed sets, obtained from the two additive resolutions of A.

Recall that a morphism $f : A \to B$ in MALL is a (cut-free) MALL proof net on the formula $A \multimap B = A^{\perp} \parr B$. A proof net on a formula C is a set of leaf functions, each taking the leaves of an additive resolution of C, satisfying three correctness conditions.[4] Let R and S denote the sets of additive resolutions of A (equiv. of $A^{\perp}$) and B, respectively. Thus the set of additive resolutions of $A \multimap B$ is in bijection with $R \times S$, since $\multimap$ is multiplicative; write $r \multimap s$ for the additive resolution of $A \multimap B$ corresponding to the additive resolutions r of A and s of B. Define the $(R \times S)$-indexed matrix $|f| : |A| \to |B|$ of a morphism (proof net) $f : A \to B$ as follows: the $\langle r, s \rangle^{\text{th}}$ element is the bag comprising every leaf function in the set f whose underlying additive resolution is $r \multimap s$. By the first proof net correctness condition, each leaf function in f has a distinct underlying additive resolution, so each such bag will be at most a singleton.

For example, let $A = P \oplus ((Q \otimes Q^{\perp}) \parr P)$ and $B = (P \oplus Q^{\perp}) \oplus (P \& P)$, and let the morphism $f : A \to B$ in MALL be the proof net $f = \{x, y, z\}$ with:



Let $a_1^+, a_2^+, a_3^-, a_4^+$ and $b_1^+, b_2^-, b_3^+, b_4^+$ be the leaves of A and B, respectively, ordered left to right. Then $|A| = \{a_1^+\} + \{a_2^+, a_3^-, a_4^+\}$ (the formal sum of two signed sets, obtained from the two additive resolutions of A, as we saw earlier) and $|B| = \{b_1^+\} + \{b_2^-\} + \{b_3^+\} + \{b_4^+\}$ (the formal sum of four singleton signed sets), and $|f|$ is the 2×4 matrix

$$\begin{array}{r} \\ \{a_1^+\} \\ \{a_2^+, a_3^-, a_4^+\} \end{array} \begin{array}{c} \begin{array}{cccc} \{b_1^+\} & \{b_2^-\} & \{b_3^+\} & \{b_4^+\} \end{array} \\ \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & 0 & y & z \end{pmatrix} \end{array}$$

where 0 denotes the empty bag, and rows and columns are labelled with the signed sets of the additive resolutions of A and B, respectively.

This faithful functor $\mathsf{MALL} \to \mathbf{Matr}(\mathbf{GoI}(\mathsf{Setp}))$ suggests a relationship with the geometry of interaction for additives [Gir95, AJ94]. Since MLL units are the main focus of the present paper, exploring this relationship is best left for another occasion.

[4]The first condition requires that just one leaf function fits on any given $\&$-resolution (definition analogous to additive resolution); the second requires that each leaf function constitutes an MLL proof net (upon identifying the underlying additive resolution r with an MLL formula, by collapsing the single-argument branches of the tree r); the third restricts how the leaf functions vary between $\&$-resolutions. We shall only require the first condition here.

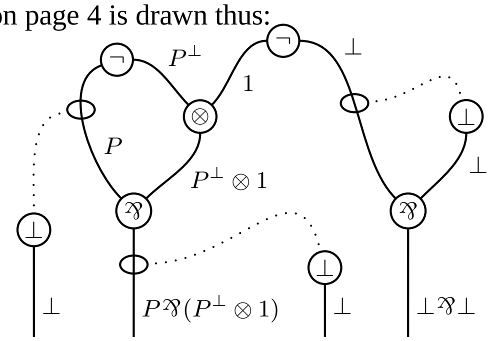
8 Previous approaches

Girard's passage quoted on the first page of the Introduction gives a convenient summary of old proof nets. Normalisation is hampered by having to move targets of $\bot$-jumps.

Proof nets for MLL with units are given in [BCST96] and [LS04]. Neither solves the problems in Girard's quote: each suffers from the need to move $\bot$-jumps during elimination, so one is lumbered once again with equivalence classes.

8.1 Circuit nets

The cut-free one-sided MLL proof nets in [BCST96] are[5] cut-free old proof nets (as described in Girard's quote, page 1) in circuit/wire notation, with an additional ordering on $\bot$-jumps. For example, the old proof net on page 4 is drawn thus:



Links are drawn as circular nodes, formulas are drawn as (labelled) wires, and $\bot$-jumps are drawn dotted. By an *MLL proof net* in the [BCST96] setting we mean the special case when the base is a set of propositional variables, and $(-)^\perp$ is restricted to propositional variables (as usual with MLL formulas). The primary net definition in [BCST96] is two-sided; a one-sided net is simply a two-sided net with the tensor unit 1 on the input side (see the paragraph following Corollary 5.3 of [BCST96]). In drawing the one-sided net above, we omitted this input unit and its jump. The minor difference with old proof nets is that when multiple $\bot$-jumps target the same wire, they are ordered along the wire; in an old proof net there is no such ordering on $\bot$-jumps targetting the same subformula.

The problem with normalisation (see Girard's passage on page 1) remains. For example, if we cut against the $P ⅋ (P^\perp \otimes 1)$ wire above, we do not have a cut redex: first we must re-wire the incoming $\bot$-jump to elsewhere in the empire of the $\bot$; we're once again resorting to equivalence classes for normalisation.

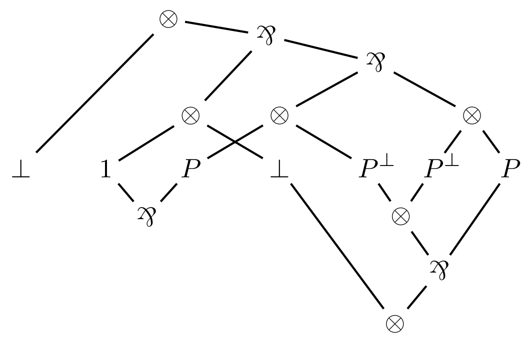
A key feature of the approach in [BCST96] is the modularity over negation and planarity. Circuit nets modulo equivalence describe the free linearly distributive and star-autonomous categories over a polygraph (*e.g.*, over a category), yielding full coherence. For an internal language presentation of free star-autonomous categories, with full coherence, see [KO99] (again modulo an equivalence/congruence).

8.2 Syntactic nets

The paper [LS04] defines a proof net on a cut sequent Γ as a separate MLL formula Θ whose leaves from left-to-right are a permutation of those of Γ. The formula Θ is shown upside down

[5]See the introduction to Section 2 of [BCST96].

above the sequent, and the permutation is represented by permitting argument edges to cross in the upper half. The $\bot$-attachments and axiom links are thus enveloped in an additional syntactic layer Θ, with $\bot$-attachments as $\,\overset{\otimes}{\frown}\,\bot$ and axiom links as $A\overset{\otimes}{\frown}A^{\perp}$. Here is an example of a proof net on the three-formula sequent $\bot,\ 1\otimes P,\ \bot\otimes((P^{\perp}\otimes P^{\perp})\parr P)$, essentially Figure 2 of [LS04]:



As with [BCST96] nets, the problem with normalisation (see Girard's passage on page 1) remains. For example[6], if Γ is the cut sequent $P^{\perp}, P\smile P^{\perp}, P\otimes Q, Q^{\perp}, \bot$ and Θ is the proof net given by the MLL formula $(P^{\perp}\otimes P)\parr\big(((P^{\perp}\otimes P)\parr(Q\otimes Q^{\perp}))\otimes\bot\big)$ (with identity permutation on leaves) then the cut cannot be reduced immediately. First one must apply invertible linear distributivity / commutativity / associativity to Θ, subject to the constraint of not breaking the correctness criterion (*i.e.*, a form of empire-rewiring [Tri94, BCST96]). Thus one is again resorting to equivalence classes for normalisation (see Theorem 4.3 of [LS04]). Syntactic nets modulo equivalence describe the free star-autonomous category with strict double involution $A = A^{\perp\perp}$ generated by a set.

References

[Abr96] S. Abramsky. Retracing some paths in process algebra. In *Proc. CONCUR '96*, pages 1–17, 1996.

[AJ94] S. Abramsky and R. Jagadeesan. New foundations for the geometry of interaction. *Information and Computation*, 111(1):53–119, 1994. Extended version of the paper published in the Proceedings of LICS'92.

[BCST96] R. F. Blute, J. R. B. Cockett, R. A. G. Seely, and T. H. Trimble. Natural deduction and coherence for weakly distributive categories. *Journal of Pure and Applied Algebra*, 113:229–296, 1996.

[Dan90] V. Danos. *La logique linéaire appliquée à l'étude de divers processus de normalisation et principalement du lambda calcul*. PhD thesis, Univ. de Paris, 1990.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gir89] J.-Y. Girard. Towards a geometry of interaction. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 69–108, 1989. Proc. of June 1987 meeting in Boulder, Colorado.

[6] This example is drawn just after Lemma 4.2 in [LS04], with different literal labels.

[Gir93] J.-Y. Girard. Linear logic: A survey. *Proceedings of the International Summer School of Marktoberdorf, NATO Advanced Science Institutes, series F94*, pages 63–112, 1993. Also in P. De Groote editor, The Curry-Howard Isomorphism, 193–255, Département de Philosophie, Université Catholique de Louvain, Cahiers du Centre de Logique 8, Academia Press.

[Gir95] J.-Y. Girard. Geometry of interaction III: Accommodating the additives. In *Advances in Linear Logic*, volume 222 of *London Math. Soc. LNS*, pages 329–389. Cambridge University Press, 1995.

[Gir96] J.-Y. Girard. Proof-nets: the parallel syntax for proof theory. In *Logic and Algebra*, volume 180 of *Lecture Notes In Pure and Applied Mathematics*. Marcel Dekker, New York, 1996.

[GSS92] J.-Y. Girard, A. Scedrov, and P. J. Scott. Bounded linear logic: A modular approach to polynomial time computability. *Theor. Comp. Sci.*, 97:1–66, 1992.

[Gue99] S. Guerrini. Correctness of multiplicative proof nets is linear. In *Proc. Logic in Computer Science '99*, 1999.

[HG03] D. J. D. Hughes and R. J. van Glabbeek. Proof nets for unit-free multiplicative additive linear logic (extended abstract). *Proc. Logic in Computer Science '03*, 2003.

[HG05] D. J. D. Hughes and R. J. van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. *ACM Transactions of Computational Logic*, 2005. To appear. Invited submission November 2003, revised January 2005. Full version of the LICS'03 paper [HG03]. Available at http://boole.stanford.edu/˜dominic/papers.

[HS03] J. M. E. Hyland and A. Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 294:183–231, 2003.

[JSV96] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Math. Proc. Camb. Phil. Soc.*, 119:447–468, 1996.

[KM71] G. M. Kelly and S. Mac Lane. Coherence in closed categories. *J. Pure and Applied Algebra*, 1:97–140, 1971.

[KO99] T. W. Koh and C.-H. L. Ong. Explicit substitution internal languages for autonomous and *-autonomous categories (preliminary version). In *Proceedings of the 8th Conference on Category Theory and Computer Science, Edinburgh, September 1999, Electronic Notes in Theoreotical Computer Science*, volume 29, 1999.

[Lam94] F. Lamarche. Proof nets for intuitionistic linear logic I: Essential nets. Unpublished note, 1994.

[LS04] F. Lamarche and L. Straßburger. On proof nets for multiplicative linear logic with units. In *Proc. CSL'04*, volume 3210 of *Lecture Notes in Computer Science*, pages 145–159, 2004.

[Mac71] Mac Lane, S. Categories for the Working Mathematician, Springer-Verlag, 1971.

[MO00] A. S. Murawski and C.-H. L. Ong. Dominator trees and fast verification of proof nets. In *Proc. Logic in Computer Science '00*, 2000.

[MO03] A. S. Murawski and C.-H. L. Ong. Exhausting strategies, joker games and full completeness for IMLL with unit. *Theor. Comp. Sci.*, 294:269–305, 2003.

[Reg92] L. Regnier. *Lambda-Calcul et Rèseaux*. PhD thesis, Univ. Paris VII, 1992.

[Tan97] A. Tan. *Full Completeness Results for Models of Linear Logic*. PhD thesis, Cambridge, Oct 1997.

[Tri94] T. H. Trimble. *Linear logic, bimodules, and full coherence for autonomous categories*. PhD thesis, Rutgers University, 1994.