Conflict nets: Efficient locally canonical MALL proof nets

Dominic Hughes
Logic Group
University of California, Berkeley

Willem Heijltjes
Department of Computer Science
University of Bath

Abstract

Proof nets for MLL (unit-free multiplicative linear logic) and ALL (unit-free additive linear logic) are graphical abstractions of proofs which are *efficient* (proofs translate in linear time) and *canonical* (invariant under rule commutation). This paper solves a three-decade open problem: are there efficient canonical proof nets for MALL (unit-free multiplicative-additive linear logic)?

Honouring MLL and ALL canonicity, in which all commutations are strictly *local* proof-tree rewrites, we define *local canonicity* for MALL: invariance under local rule commutation. We present new proof nets for MALL, called *conflict nets*, which are both efficient and locally canonical.

Categories and Subject Descriptors F.4.1 [*Mathematical Logic and Formal Languages*]: Mathematical Logic—Proof theory

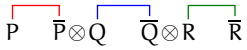
Keywords linear logic, proof nets, MALL, multiplicative-additive linear logic

1. Introduction

Proof nets for MLL (unit-free multiplicative linear logic [13]) are geometric abstractions of MLL proofs. For example, the two proofs

$$\dfrac{\dfrac{\overline{P,\overline{P}} \quad \overline{Q,\overline{Q}}}{P,\ \overline{P}\otimes Q,\ \overline{Q}} \quad \overline{R,\overline{R}}}{P,\ \overline{P}\otimes Q,\ \overline{Q}\otimes R,\ \overline{R}}\otimes \qquad \dfrac{\overline{P,\overline{P}} \quad \dfrac{\overline{Q,\overline{Q}} \quad \overline{R,\overline{R}}}{Q,\ \overline{Q}\otimes R,\ \overline{R}}\otimes}{P,\ \overline{P}\otimes Q,\ \overline{Q}\otimes R,\ \overline{R}}\otimes$$

translate to the same MLL proof net, with 3 *axiom links*:

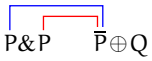


The net abstracts away the arbitrary choice of order between the independent $\otimes$ rules, one introducing $\overline{P}\otimes Q$ and the other $\overline{Q}\otimes R$, in separate parts of the sequent. MLL proof nets are *canonical* in the sense that they are invariant under rule commutation: proofs differing by a commutation of adjacent rules have the same net. For example, the net above is invariant upon commuting adjacent $\otimes$ rules.

Similarly, proof nets for ALL (unit-free additive linear logic) in binary-relation formulation [18, 21, 25, 26] are canonical geometric abstractions of ALL proofs. For example, the two ALL proofs

$$\dfrac{\overline{P,\overline{P}} \qquad \overline{P,\overline{P}}}{\dfrac{P\&P,\ \overline{P}}{P\&P,\ \overline{P}\oplus Q}\oplus_1}\& \qquad \oplus_1\dfrac{\dfrac{\overline{P,\overline{P}}}{P,\ \overline{P}\oplus Q} \qquad \dfrac{\overline{P,\overline{P}}}{P,\ \overline{P}\oplus Q}\oplus_1}{P\&P,\ \overline{P}\oplus Q}\&$$

differ by a $\oplus_1$/& rule commutation and translate to the same ALL proof net, with 2 axiom links:



Both MLL and ALL proof nets are *efficient*: a proof translates to a net in linear time. This paper solves a problem which has been open since the inception of linear logic [13]:

- **Problem:** are there efficient canonical proof nets for MALL (unit-free multiplicative-additive linear logic)?

Our solution has two parts:

1. Honouring MLL and ALL canonicity, in which every rule commutation is a strictly *local* rewrite in a proof tree, we define *local canonicity* for MALL: invariance under local rule commutation.
2. We introduce new MALL proof nets, called *conflict nets*, which are both efficient and locally canonical.

1.1 Local canonicity

Each rule commutation of MLL and ALL is a local rewrite in a proof tree. For example, here is an MLL rule commutation which raises a $\otimes$-rule up over the $\parr$-rule immediately above it:

$$\dfrac{\dfrac{\dfrac{\overline{P,\overline{P}} \quad \overline{Q,\overline{Q}}}{P,\ \overline{P}\otimes Q,\ \overline{Q}} \quad \dfrac{\overline{R,\overline{R}} \quad \overline{S,\overline{S}}}{R,\ \overline{R}\otimes S,\ \overline{S}}\otimes}{P,\ \overline{P}\otimes Q,\ \overline{Q}\otimes(\overline{R}\otimes S),\ R\parr\overline{S}}\parr}{P\parr(\overline{P}\otimes Q),\ \overline{Q}\otimes(\overline{R}\otimes S),\ R\parr\overline{S}}\otimes \quad\to\quad \dfrac{\dfrac{\dfrac{\overline{P,\overline{P}} \quad \overline{Q,\overline{Q}}}{P,\ \overline{P}\otimes Q,\ \overline{Q}} \quad \dfrac{\overline{R,\overline{R}} \quad \overline{S,\overline{S}}}{R,\ \overline{R}\otimes S,\ \overline{S}}\otimes}{P,\ \overline{P}\otimes Q,\ \overline{Q}\otimes(\overline{R}\otimes S),\ R,\ \overline{S}}\otimes}{P\parr(\overline{P}\otimes Q),\ \overline{Q}\otimes(\overline{R}\otimes S),\ R\parr\overline{S}}\parr$$

This commutation is a strictly *local* rewrite: the two upper subproofs of $P,\overline{P}\otimes Q,\overline{Q}$ and $R,\overline{R}\otimes S,\overline{S}$ (shaded grey) remain intact, as does the continuation below the commutation (also shaded). All six rule commutations of MLL and ALL are shown in Fig. 1. The example above is an instance of the $\parr/\otimes$-commutation in Fig. 1 with $\Gamma = P,\overline{P}\otimes Q$, $A=\overline{Q}$, $B=\overline{R}\otimes S$, Δ empty, $C=R$, and $D=\overline{S}$.

MALL has four additional rule commutations. Faithful to MLL and ALL locality, three of them are also local, for example,

$$\dfrac{\Gamma,A,B,C \qquad \Gamma,A,B,D}{\dfrac{\Gamma,A,B,C\&D}{\Gamma,A\parr B,C\&D}\parr}\& \quad\to\quad \dfrac{\dfrac{\Gamma,A,B,C}{\Gamma,A\parr B,C}\parr \qquad \dfrac{\Gamma,A,B,D}{\Gamma,A\parr B,D}\parr}{\Gamma,A\parr B,C\&D}\&$$

This commutation duplicates the $\parr$-rule locally, but does not duplicate either of the subproofs of Γ,A,B,C or Γ,A,B,D. The three local rule commutations of MALL are in Fig. 2.

LICS '16 July 5–8, 2016, New York, NY, USA

ACM 978-1-4503-4391-6/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2933575.2934559

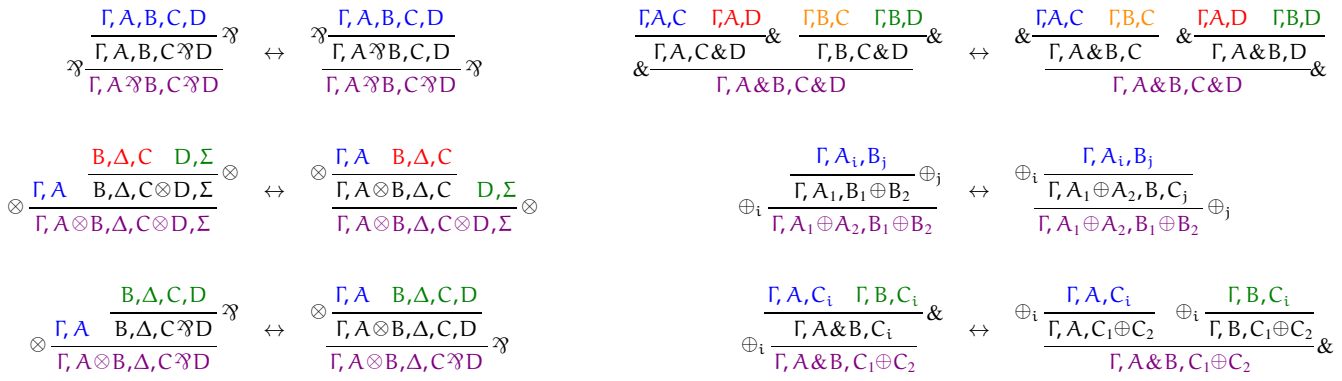
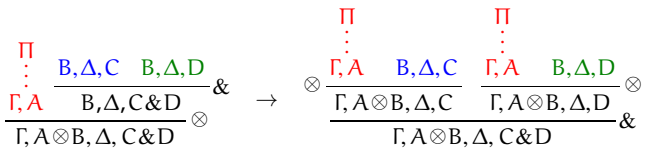


Figure 1. The rule commutations of MLL (left) and ALL (right). Each is a local rewrite in a proof tree.

However, the fourth, the ⊗/&-commutation, fails to be local because it duplicates an entire proof tree Π:



Accordingly, we distinguish two canonicity properties for a system of MALL proof nets:

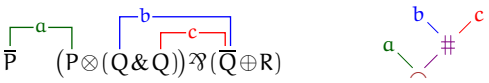
- ***Local canonicity***: invariant under *local* rule commutations.
- ***Strong canonicity***: invariant under *all* rule commutations.

The *slice nets* of Hughes and van Glabbeek [25, 26] are strongly canonical, but not efficient. Conflict nets, summarized below, are locally canonical and efficient.

1.2 Conflict nets: a whirlwind tour

This subsection is intended as a quick impressionistic overview of conflict nets, omitting details.

A conflict net is an axiom linking with a *cotree* alternating between *conflict* # and *concord* ⌢ nodes, e.g.



There are three axiom links a b c, between which # is an additive relationship (akin to a &-rule in a proof) and ⌢ is a multiplicative relationship (akin to a ⊗-rule).

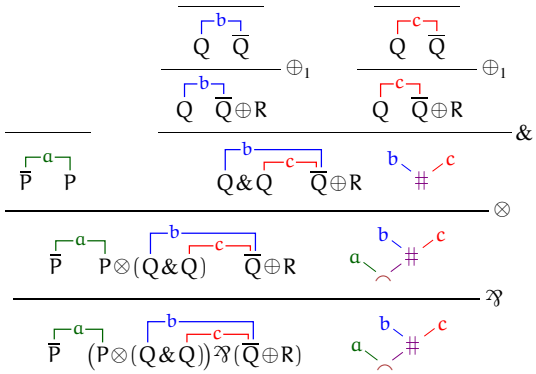
1.2.1 Efficient translation

Translation is simple and efficient (linear time):

- axiom rules descend to axiom links (just like MLL and ALL)
- ⊗/& rules join the two cotrees at a new root ⌢/#, respectively.[1]

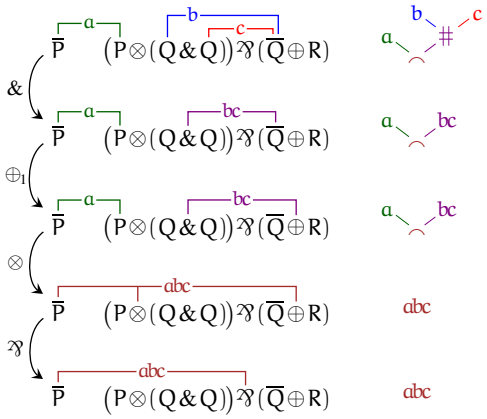
For example, here is a rule-by-rule translation of a MALL proof to the conflict net displayed above:

[1] Adjacent ⌢s or #s that result are collapsed, to recover ⌢/# alternation.



1.2.2 Coalescence correctness

Geometric correctness is *coalescence*, a form of rewriting that can be thought of as abstract, top-down sequentialization, generalizing additive coalescence [18] and multiplicative contractibility [8]. A conflict net is correct if it *coalesces* to a single link on all formula roots. An example is below. Details are in §5; our goal here is only to convey an overall impression:



The conflict net is correct because the final link touches the roots of both formulas in the sequent, and nothing else.

1.2.3 Faithfulness to MLL and ALL

Proof nets for MLL [13] and ALL in binary-relation formulation [18, 21, 25, 26] are consummate categorically, representing the free unit-free star-autonomous category [19, 20] and the free binary product-coproduct category [21]. Conflict nets remain faithful to both, by including them as sub-systems: an MLL/ALL net is a conflict net

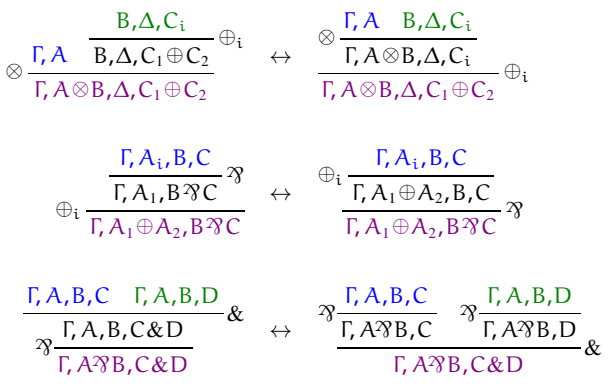
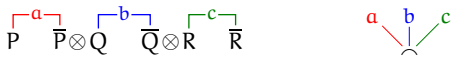


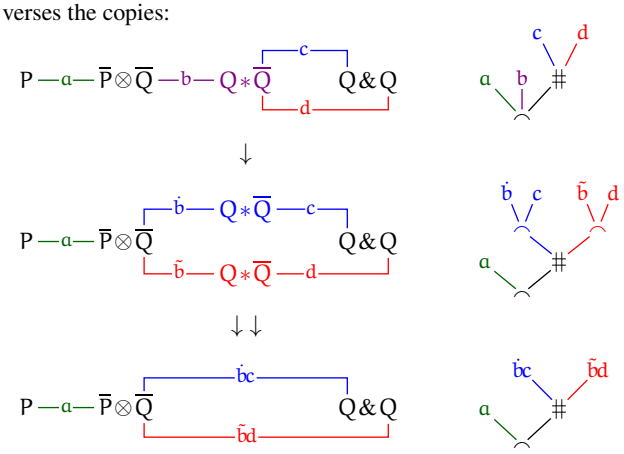
Figure 2. The additional local rule commutations of MALL. Each is a local rewrite in a proof tree.

with one $\frown/\#$. For example, the MLL net at the beginning of the paper is



1.2.4 Strongly normalizing cut elimination

Cut elimination extends that of both MLL and ALL and is strongly normalizing. The example below duplicates a cut $Q * \overline{Q}$ then traverses the copies:



1.3 The dichotomy: efficient *versus* strongly canonical

Define proof nets as *rigid* if they do not compress proofs: the size of a net is at least linear in the size of a proof. For example, standard MLL nets and ALL nets described above are rigid: an MLL/ALL proof with n axiom rules becomes a proof net with n axiom links.

A rigid MALL proof net system *cannot be both efficient and strongly canonical*: repeatedly raising $\otimes$ rules up over & rules blows up the size of a proof, so translation to strongly canonical nets must be exponential time.[2] Both slice nets [25, 26] and conflict nets are rigid. Where slice nets are strongly canonical but not efficient, conflict nets are efficient but not strongly canonical.

1.4 Related work

Box nets. Girard's box nets [13] are efficient but not canonical: the two ALL proofs displayed earlier have distinct box nets. Box nets are faithful to MLL nets, but not to ALL nets.

[2] This dichotomy is generally believed to hold even without rigidity. Current research by Mark Bagnol aims to formalize such a result.

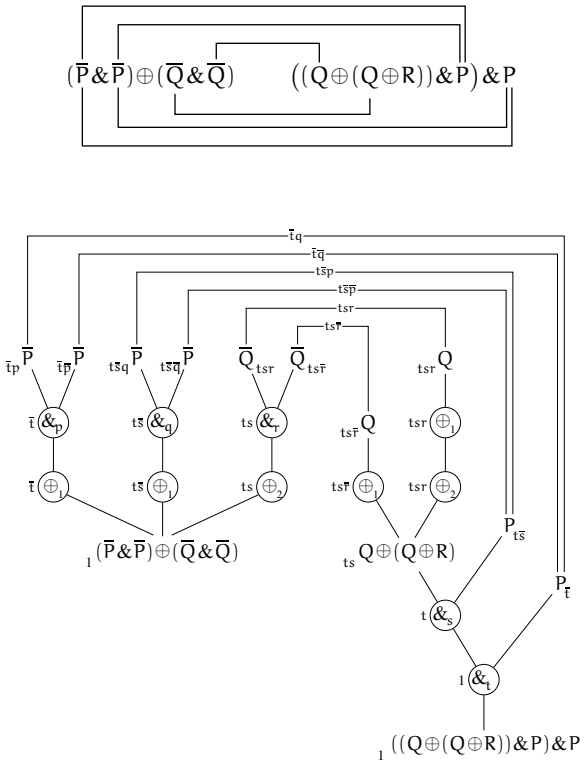


Figure 3. Illustrating the unwieldiness of monomial nets. Above: an ALL net, which is a special case of a conflict net (with a single # node, omitted). Below: the corresponding monomial net.

Monomial nets. Girard's monomial nets [14, 30] are neither efficient nor locally canonical. Efficiency fails because monomials encode all paths through a &-rule tree: see Prop. 16. Local canonicity fails since proof translation is not deterministic. With Girard's (non-surjective) deterministic translation [14, p. 7], which never merges monomials, local canonicity also fails: commuting any rule up over a &-rule yields a different monomial net. Monomial nets are faithful to MLL nets but not ALL nets. For a discussion of other issues with monomial nets, see [25, 26]. The lack of ALL faithfulness manifests in a degree of unwieldiness: see Fig. 3.

Slice nets. As discussed in §1.3, slice nets [25, 26] are strongly canonical, so necessarily inefficient (since they are rigid). They are faithful to both MLL and ALL nets.

Complexity trade-off. For MALL proof nets there is a complexity trade-off between proof translation, sequentialization and cut-elimination. The following table shows how monomial, slice and conflict nets negotiate this trade-off:

	Monomial	Slice	Conflict
P-time sequentialization	✗	✓	✓
P-time proof translation	✓	✗	✓
P-time cut elimination	?	✓	✗

Monomial net translation if P-time and sequentialization (retrieving a proof) is exponential (Prop. 18, §7). Slice nets attain strong canon-

icity by sacrificing P-time translation[3]; cut elimination and sequentialization are P-time. Conflict nets reflect sequent calculus: proof translation and sequentialization are P-time, and the cost of computation resides in cut-elimination, which is exponential (see [32], and also Prop. 17, §7, for the complexity of MALL cut-elimination).

Other related work. There is an established tradition in mathematics, computer science and proof theory of using graphs to abstract syntax, exemplified by Kelly–Mac Lane graphs for coherence in monoidal closed categories [28], connections/matings in classical logic [2, 5], string diagrams to represent maps in braided categories [27], and Girard's programme of finding proof nets for linear logic [13].

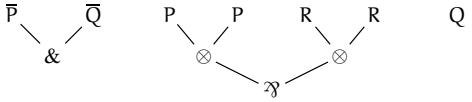
Semantically canonical proof nets have been described for several fragments of linear logic, starting with MLL [13]. For ALL, canonical representations appeared in various guises: coherence spaces [21], the connections method in proof search [12], coherence for categories with sums and products [10], and in the style of string diagrams [1]. Canonical proof nets further appeared for the additive fragment with units (ALLU) [16]; for polarised linear logic [29] and tensorial logic [33]; and for the multiplicative-additive fragment without units (MALL) [25, 26]. Canonical forms for MALL can be obtained in the sequent calculus via focussing [6]. Other MALL nets include contractible proof nets [31] and ludics nets [7, 11].

Conflict nets are a variant of combinatorial proofs for classical logic [22, 23]: each conflict net can be viewed [24] as a maximal map (homomorphism) of contractible coherence spaces (P4-free graphs), from axioms to sequent. The crude draft [24] on conflict nets had the data structure of a conflict net, but lacked both a clean correctness criterion (here coalescence) and cut elimination.

Recent work has emphasised the interplay between canonicity and complexity. Correctness of MLL proof nets is linear-time [15], as is correctness for ALLU proof nets [18], while the problem is NL-complete for strongly canonical MALL proof nets [9]. The interplay is particularly strong for proof equivalence problems, which may be decided by via translation to canonical proof nets: for MLL with units the problem is PSPACE-complete [17], effectively ruling out canonical proof nets, while for MALL it is LOGSPACE-complete [3].

2. MALL

Formulas $A, B, \ldots$ are built from literals (propositional variables $P, Q, \ldots$ and their duals $\overline{P}, \overline{Q}, \ldots$) by *tensor* $A \otimes B$, *par* $A \parr B$, *with* $A \,\&\, B$ and *plus* $A \oplus B$. Duality extends to formulas by $\overline{A \otimes B} = \overline{B} \parr \overline{A}$, $\overline{A \parr B} = \overline{B} \otimes \overline{A}$, $\overline{A \,\&\, B} = \overline{B} \oplus \overline{A}$ and $\overline{A \oplus B} = \overline{B} \,\&\, \overline{A}$.[4] We identify a formula with its parse tree, labelled with literals on leaves and connectives on internal vertices. A sequent Γ or Δ is a disjoint union of formulas (a labelled forest), using comma $(,)$ for disjoint union. For example $\overline{P} \,\&\, \overline{Q}, (P \otimes P) \parr (R \otimes R), Q$ is the labelled forest



Proofs are built via the following rules [13]:

$$\frac{}{P, \overline{P}} \qquad \frac{\Gamma, A, B}{\Gamma, A \parr B}\parr \qquad \frac{\Gamma, A \quad B, \Delta}{\Gamma, A \otimes B, \Delta}\otimes$$

$$\frac{\Gamma, A \quad \overline{A}, \Delta}{\Gamma, \Delta}\,\mathrm{cut} \qquad \frac{\Gamma, A_i}{\Gamma, A_1 \oplus A_2}\oplus_i \qquad \frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \,\&\, B}\,\&$$

[3] Multi-focussing yields strongly canonical forms within sequent calculus [6], inducing the same exponential blow-up.

[4] We define the dual of $A \otimes B$ as $\overline{B} \parr \overline{A}$, rather than $\overline{A} \parr \overline{B}$, to preserve planarity during cut elimination, making examples easier to draw.

3. Cotrees and axiom linkings

A ***link*** on a sequent Γ, or Γ-***link***, is a subsequent of Γ, i.e., a subgraph which is a well-formed sequent. For example, if Δ is the sequent

$$P \parr Q \qquad (S \oplus T) \otimes (\overline{Q} \otimes \overline{R}) \qquad \overline{T} \,\&\, \overline{S}$$

then here is a link Λ on Δ:

$$P \quad Q \qquad\qquad \overline{Q} \otimes \overline{R}$$

We usually draw a link graphically as a horizontal line with vertical line segments picking out the root vertices of the subformulas inside the sequent. For example, we draw the link Λ above as:

$$P \parr Q \qquad (S \oplus T) \otimes (\overline{Q} \otimes \overline{R}) \qquad \overline{T} \,\&\, \overline{S}$$

We write the union of disjoint links Λ and Ω as juxtaposition $\Lambda\,\Omega$.

A ***cotree*** $\mathcal{T}$ on Γ is a tree of ***conflict*** $\#$ and ***concord*** $\frown$ nodes with a Γ-link at each leaf:

$$\mathcal{T} \;::=\; \Lambda \;\mid\; \#(\mathcal{F}) \;\mid\; \frown(\mathcal{F})$$

where Λ is any Γ-link and $\mathcal{F}$ is a finite non-empty multiset of cotrees, each an ***argument*** of the node. A ***coforest*** is a multiset of cotrees, and $\mathcal{F}; \mathcal{G}$ denotes the disjoint union of coforests $\mathcal{F}$ and $\mathcal{G}$. For example, if Γ is the sequent

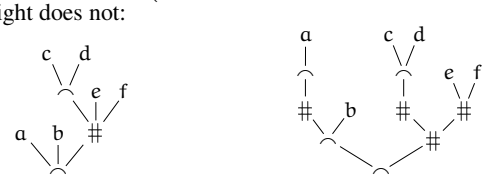
$$P \,\&\, Q \qquad \overline{Q} \otimes \overline{R} \qquad R \oplus S$$

then here is a cotree $\mathcal{T}$ on Γ:

$$\#\,\big(\; P \;\; \overline{Q} \otimes R \;\; R \oplus S \;;\; \frown(Q\overline{Q}; R\overline{R}) \;;\; Q \;\; \overline{Q} \otimes R \;\; S \;\big)$$

The $\frown$ node has two arguments (the links $Q\overline{Q}$ and $R\overline{R}$) and the $\#$ node has three arguments (the link $P \;\overline{Q} \otimes R \;\; R \oplus S$, the cotree $\frown(Q\overline{Q}; R\overline{R})$, and the link $Q \;\; \overline{Q} \otimes R \;\; S$). We generally draw both the links and the cotree in graphical form. For example, the above cotree $\mathcal{T}$ becomes:

A node is ***unary*** if it has a single argument. A cotree ***alternates*** if no node is unary, no $\#$ nodes are adjacent, and no $\frown$ nodes are adjacent. For example, the cotree $\mathcal{T}_1 = \frown\big(a; b; \#(\frown(c; d); e; f)\big)$ below-left alternates but $\mathcal{T}_2 = \frown\Big(\frown\big(\#(\frown(a)); b\big) \,;\, \#\big(\#(\frown(c; d))\big) \,;\, \#(e; f)\Big)$ below-right does not:



$\mathcal{T}_1 = \lfloor \mathcal{T}_2 \rfloor$ $\qquad\qquad$ $\mathcal{T}_2$

The ***alternating form*** $\lfloor \mathcal{T} \rfloor$ of a cotree $\mathcal{T}$ is the canonical alternating cotree associated with $\mathcal{T}$: collapse unary nodes, adjacent $\#$ nodes, and adjacent $\frown$ nodes. For example, the tree $\mathcal{T}_1$ above-left is the alternating form $\lfloor \mathcal{T}_2 \rfloor$ of the cotree $\mathcal{T}_2$ above-right. Formally, the alternating form of a cotree is the result of exhaustively applying the following rewrites on subcotrees, where $\mathcal{T}$ is any cotree and $\mathcal{F}$ and $\mathcal{G}$ are coforests:

$$\#(\mathcal{T}) \;\to\; \mathcal{T} \qquad\qquad \#\big(\#(\mathcal{F}); \mathcal{G}\big) \;\to\; \#(\mathcal{F}; \mathcal{G})$$

$$\frown(\mathcal{T}) \;\to\; \mathcal{T} \qquad\qquad \frown\big(\frown(\mathcal{F}); \mathcal{G}\big) \;\to\; \frown(\mathcal{F}; \mathcal{G})$$

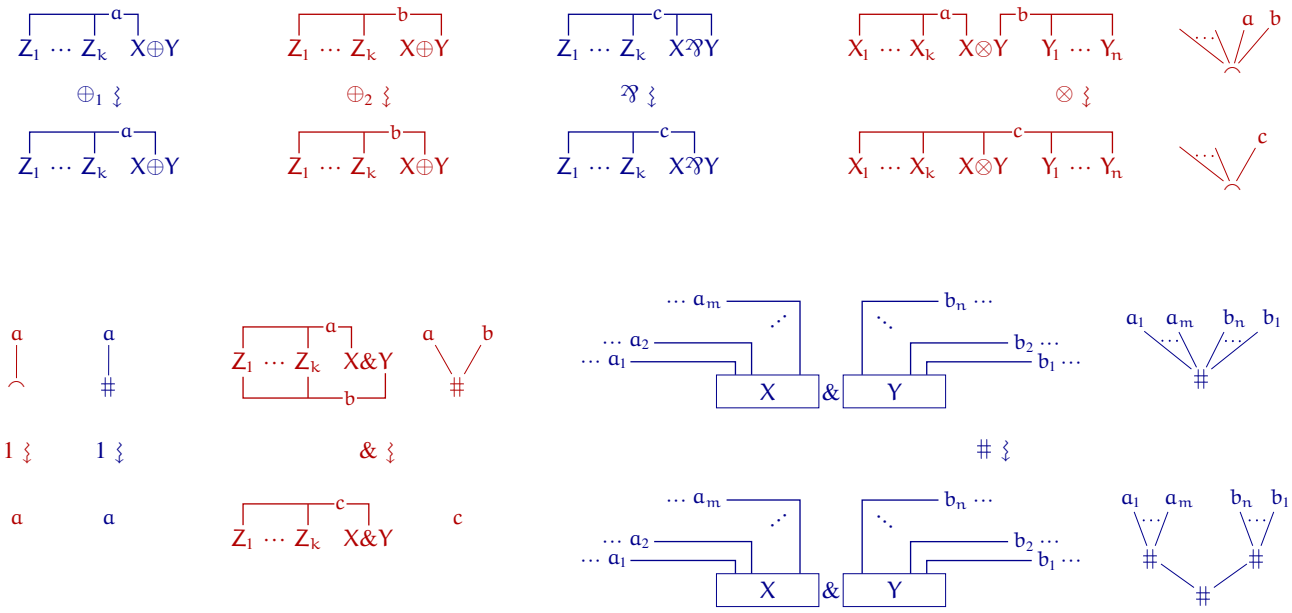


Figure 4. Coalescence steps. In the 1-steps $\mathfrak{a}$ is a link whose subsequent is not displayed. In all steps but $\#$-coalescence, the links target the (roots of the) subformulas shown. In the $\#$-coalescence step, the full horizontal extent of the formulas X and Y are shown as boxes, and the links target (roots of) subformulas of X and Y.

3.1 Axiom linkings

An ***axiom link*** is a link of the form $P\overline{P}$ for some literal P. A cotree is ***axiomatic*** if every link is an axiom link. An ***axiom linking*** is an axiomatic, alternating cotree. For example, here is an axiom linking on the sequent $P\&P, \overline{P}\otimes\overline{R}, R\&R$:

Note that d and f are instances of the same axiom link, as are b and e, and c and g.

3.2 Notation conventions

We employ the following conventions for variables and disjoint union throughout the paper:

Sequents	*Links*	*Coforests*
Formulas A B C D	Formulas X Y	Cotrees $\mathcal{T}$ $\mathcal{U}$
Sequents Γ Δ Σ	Links Λ Ω	Coforests $\mathcal{F}$ $\mathcal{G}$
$\Gamma, A\otimes B, \Delta$	$\Lambda\ X\otimes Y\ \Omega$	$\mathcal{F};\mathcal{T};\mathcal{G}$

4. Translating a proof to an axiom linking

When we add a node between alternating cotrees $\mathcal{T}$ and $\mathcal{U}$ to form $\#(\mathcal{T};\mathcal{U})$ or $\frown(\mathcal{T};\mathcal{U})$ the resulting cotree need not alternate. We obtain binary operations $\mathbin{\#}$ and $\mathbin{\frown}$ which preserve alternation by collapsing to alternating form:

$$\mathcal{T}\mathbin{\#}\mathcal{U} \;=\; \lfloor\#(\mathcal{T};\mathcal{U})\rfloor \qquad \mathcal{T}\mathbin{\frown}\mathcal{U} \;=\; \lfloor\frown(\mathcal{T};\mathcal{U})\rfloor$$

For example $(\mathcal{T}_1\mathbin{\frown}\mathcal{T}_2)\mathbin{\frown}((\mathcal{U}_1\mathbin{\#}\mathcal{U}_2)\mathbin{\#}\mathcal{U}_3) = \frown(\mathcal{T}_1;\mathcal{T}_2;\#(\mathcal{U}_1;\mathcal{U}_2;\mathcal{U}_3))$. Both operations are associative and commutative, so we can write the same cotree as $\mathcal{T}_1\mathbin{\frown}(\mathcal{U}_3\mathbin{\#}\mathcal{U}_1\mathbin{\#}\mathcal{U}_2)\mathbin{\frown}\mathcal{T}_2$.

Definition 1. A cut-free MALL proof Π of Γ ***translates*** to the axiom linking $[\![\Pi]\!]$ on Γ, defined by:

$$\left[\!\!\left[\ \overline{P, \overline{P}}\ \right]\!\!\right] \;=\; P\overline{P}$$

$$\left[\!\!\left[\ \dfrac{\Pi}{\Gamma, A, B}\ \right]\!\!\right] = [\![\Pi]\!] \qquad \left[\!\!\left[\ \dfrac{\Pi_1 \quad \Pi_2}{\Gamma, A \quad B, \Delta}\ \right]\!\!\right] = [\![\Pi_1]\!] \mathbin{\frown} [\![\Pi_2]\!]$$

$$\left[\!\!\left[\ \dfrac{\Pi}{\Gamma, A_i}\ \right]\!\!\right] = [\![\Pi]\!] \qquad \left[\!\!\left[\ \dfrac{\Pi_1 \quad \Pi_2}{\Gamma, A \quad \Gamma, B}\ \right]\!\!\right] = [\![\Pi_1]\!] \mathbin{\#} [\![\Pi_2]\!]$$

An example of step-by-step translation from a proof to an axiom linking was shown in §1.2.1.

5. Conflict nets and coalescence

Fix a sequent Γ. The ***coalescence*** relation $(\rightsquigarrow)$ is the rewrite relation on cotrees on Γ generated by the following rewrites, illustrated graphically in Fig. 4, where $\mathcal{T} \rightsquigarrow \mathcal{T}'$ generates a ***coalescence step*** $\mathcal{U} \rightsquigarrow \mathcal{U}'$ if $\mathcal{U}$ yields $\mathcal{U}'$ by replacing a subtree $\mathcal{T}$ with $\mathcal{T}'$.

$$\Lambda\, X Y \;\rightsquigarrow\; \Lambda\, X \parr Y \qquad (\parr)$$

$$\Lambda\, X_i \;\rightsquigarrow\; \Lambda\, X_1 \oplus X_2 \qquad (\oplus_i)$$

$$\frown(\mathcal{F}; \Lambda X; Y\Omega) \;\rightsquigarrow\; \frown(\mathcal{F}; \Lambda\, X\otimes Y\, \Omega) \qquad (\otimes)$$

$$\#(\Lambda X; \Lambda Y) \;\rightsquigarrow\; \Lambda\, X \& Y \qquad (\&)$$

$$\begin{aligned} \frown(\Lambda) &\;\rightsquigarrow\; \Lambda \\ \#(\Lambda) &\;\rightsquigarrow\; \Lambda \end{aligned} \qquad (1)$$

$$\#(\mathcal{F}_X; \mathcal{F}_Y) \;\rightsquigarrow\; \#\big(\#(\mathcal{F}_X); \#(\mathcal{F}_Y)\big) \qquad (\#)$$

where the $(\#)$ rewrite has the following conditions:

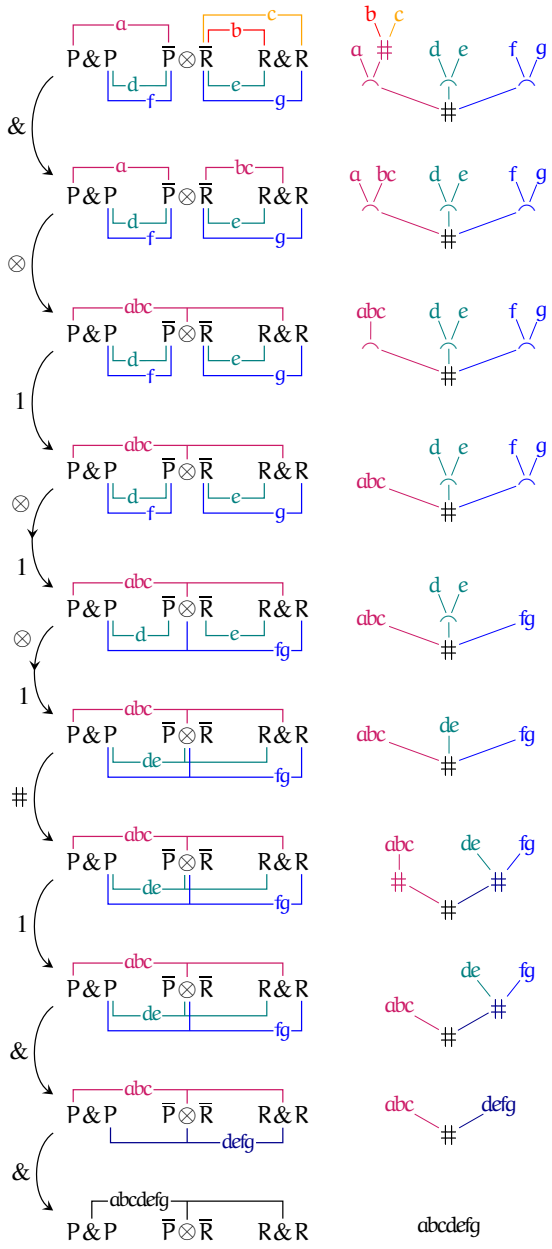


Figure 5. Verifying a conflict net via coalescence.

- $\mathcal{F}_X$ and $\mathcal{F}_Y$ are non-empty multisets of links.
- $\mathcal{F}_X;\mathcal{F}_Y$ contains three or more links.
- Γ contains a subformula $X\&Y$ such that:
  - every link in $\mathcal{F}_X$ chooses X
  - every link in $\mathcal{F}_Y$ chooses Y

  where a link ***chooses*** X if it intersects X but not Y, and vice versa.

A cotree $\mathcal{T}$ ***coalesces to*** $\mathcal{T}'$ if $\mathcal{T} \rightsquigarrow^* \mathcal{T}'$ (where $\rightsquigarrow^*$ is the reflexive-transitive closure of $\rightsquigarrow$), and it ***coalesces*** if it coalesces to Γ (the cotree comprising a single link Γ). Figure 5 shows an axiom linking coalescing, involving ($\otimes$), (&), (1) and (#) rewrites. Section §1.2.2 includes an example involving ($\parr$), ($\oplus$), ($\otimes$) and (&) rewrites (leaving two (1) steps implicit).

A ***pre-net*** $\mathcal{T} :: \Gamma$ is a sequent Γ and a cotree $\mathcal{T}$ on Γ.

Definition 2. A ***conflict net*** is a pre-net whose cotree is an axiom linking which coalesces.

5.1 Sequentialization

Coalescence is essentially top-down sequentialization: each axiom link in a conflict net corresponds to an axiom rule in a proof and each coalescence step introduces a rule. To record the proof produced by coalescence, a version of conflict nets is introduced where leaves carry proofs, rather than merely sequents.

A ***deductive cotree*** on Γ is the generalization of a cotree on Γ in which each leaf is a MALL proof of a subsequent of Γ, rather than just a subsequent. A deductive cotree on Γ projects to an ordinary cotree by replacing each proof by its concluding subsequent of Γ. Conversely, an axiomatic cotree may be considered a deductive cotree by taking each link $P\ \overline{P}$ to be the corresponding axiom rule. Coalescence extends to deductive cotrees by combining each of the four rewrite steps ($\parr$), ($\oplus$), ($\otimes$), and (&) with its corresponding MALL rule: apply the rule to the proofs in the redex to form a proof of the sequent in the contractum. For example, in ($\otimes$) a proof of $\Lambda,\Omega,X\otimes Y$ is formed by applying the $\otimes$-rule to the proofs of Λ,X and Ω,Y in the redex. The rewrites (1) and (#) leave proofs intact.

Definition 3. A deductive conflict net $\mathcal{T} :: \Gamma$ ***sequentializes*** to a proof Π with conclusion Γ if it coalesces to $\Pi :: \Gamma$.

Sequentialization is the inverse of proof translation:

Theorem 4. *A conflict net $\mathcal{T} :: \Gamma$ sequentializes to a* MALL *proof Π if and only if $[\![\Pi]\!] = \mathcal{T}$ and Π has conclusion Γ.*

Proof. Suppose $\mathcal{T} :: \Gamma$ sequentializes to $\Pi :: \Gamma$. Define cotree equivalence by $\mathcal{T} \sim \mathcal{U}$ if and only if $\lfloor \mathcal{T} \rfloor = \lfloor \mathcal{U} \rfloor$, i.e., the cotrees have the same alternating form. In each deductive coalescence step, the redex and contractum translate to equivalent pre-nets (i.e., pre-nets with equivalent cotrees). For sequentializing $\mathcal{T} :: \Gamma$ to $\Pi :: \Gamma$ this means Π translates to a conflict net $\mathcal{T}' :: \Gamma$ with $\mathcal{T} \sim \mathcal{T}'$. Since $\mathcal{T}'$ and $\mathcal{T}$ are both alternating, Π translates to $\mathcal{T} :: \Gamma$.

Conversely, suppose $[\![\Pi]\!] = \mathcal{T}$. We show by induction on Π that every subtree of $\mathcal{T}$ sequentializes to a subproof of Π, whence $\mathcal{T} :: \Gamma$ sequentializes to Π. The base case with Π an axiom rule is immediate. Observe that if $\mathcal{T} :: \Gamma \rightsquigarrow \mathcal{T}' :: \Gamma$ for Γ a subsequent of Γ' then $\mathcal{T} :: \Gamma' \rightsquigarrow \mathcal{T}' :: \Gamma'$. There are four cases [$\parr$] [$\oplus$] [$\otimes$] [&] according to the structure of Π, respectively:

$$\frac{\Pi'}{\Lambda,X,Y}{\Lambda,X\parr Y} \qquad \frac{\Pi'}{\Lambda,X_i}{\Lambda,X_1\oplus X_2} \qquad \frac{\Pi_1 \quad \Pi_2}{\Lambda,X \quad Y,\Omega}{\Lambda,X\otimes Y,\Omega} \qquad \frac{\Pi_1 \quad \Pi_2}{\Lambda,X \quad \Lambda,Y}{\Lambda,X\&Y}$$

Case [$\parr$]. $[\![\Pi]\!] = [\![\Pi']\!]$ and by induction $[\![\Pi']\!] :: \Lambda,X,Y$ sequentializes to Π'. By the observation above $[\![\Pi']\!] :: \Lambda,X\parr Y$ coalesces to $\Pi' :: \Lambda,X\parr Y$ which coalesces in one step to $\Pi :: \Lambda,X\parr Y$.

Case [$\oplus$] is like [$\parr$].

Case [$\otimes$]. $[\![\Pi]\!] = [\![\Pi_1]\!] \frown [\![\Pi_2]\!]$. There are four cases, depending on whether the root of each $[\![\Pi_i]\!]$ is $\frown$ or not; without loss of generality let $[\![\Pi_1]\!] = \frown(\mathcal{F})$ with $[\![\Pi_2]\!]$ not $\frown$-rooted, so $[\![\Pi]\!] = \frown(\mathcal{F};[\![\Pi_2]\!])$. By induction $[\![\Pi_1]\!] :: \Lambda,X$ sequentializes to Π_1, and $[\![\Pi_2]\!] :: Y,\Omega$ to Π_2. Then $[\![\Pi]\!]$ on the sequent $\Lambda,X\otimes Y,\Omega$ coalesces via $\frown(\mathcal{F},[\![\Pi_2]\!]) \rightsquigarrow^* \frown(\Pi_1,\Pi_2) \rightsquigarrow \frown(\Pi) \rightsquigarrow \Pi$.

Case [&]. $[\![\Pi]\!] = [\![\Pi_1]\!] \# [\![\Pi_2]\!]$. There are again four cases; without loss of generality let $[\![\Pi_1]\!] = \#(\mathcal{F})$ with $[\![\Pi_2]\!]$ not #-rooted, so $[\![\Pi]\!] = \#(\mathcal{F},[\![\Pi_2]\!])$ where $\mathcal{F} = \mathcal{T}_1,\ldots,\mathcal{T}_m$. By induction and prior observations $[\![\Pi_1]\!]$ and $[\![\Pi_2]\!]$ over $\Lambda,X\&Y$ coalesce to Π_1 and Π_2. We construct a coalescence sequence over $\Lambda,X\&Y$:

$$\begin{aligned}\#(\mathcal{F},[\![\Pi_2]\!]) &\rightsquigarrow^* \#(\Sigma_1;\ldots;\Sigma_m;\Pi_2)\\ &\rightsquigarrow \#(\#(\Sigma_1;\ldots;\Sigma_m);\#(\Pi_2)) \rightsquigarrow^* \#(\Pi_1;\Pi_2) \rightsquigarrow \Pi\end{aligned}$$

The stronger induction hypothesis ensures each $\mathcal{T}_i$ coalesces to a subproof Θ_i of Π_l. The above sequence works if the side conditions of the (#) rewrite hold: we must show that the conclusion of each Θ_i intersects X. This follows by induction on Π_l. Since a $\otimes$-rule would create a $\frown$-rooted cotree, Π is constructed from the subproofs Θ_i using only $\parr$, $\oplus$ and & rules. The conclusion of Π_l is Λ,X, so the conclusion of each Θ_i must intersect X. Thus $[\![\Pi]\!] :: \Lambda,X\&Y$ sequentializes to Π via the above coalescence sequence. □

5.2 Local canonicity

Two proofs are ***homeomorphic***, denoted $\Pi \simeq \Pi'$, if Π yields Π' by a sequence of (0 or more) local rule commutations (displayed in Figures 1 and 2).

Theorem 5 (Local Canonicity)**.** *Homeomorphic proofs translate to the same conflict net:* $\Pi \simeq \Pi'$ *implies* $[\![\Pi]\!] = [\![\Pi']\!]$.

Proof. A routine induction on the size of the proof. □

5.3 Confluence of coalescence

For coalescence to be a reasonable correctness criterion and sequentialization procedure we require:

- If one coalescence path succeeds, all paths eventually succeed.
- Two sequentializations of the same net must be homeomorphic.

The former means that testing any one coalescence path is sufficient to determine correctness, as opposed to testing all possible paths. This is essential for coalescence to be tractable.

The two properties can be summarised as saying that sequentialization for conflict nets should be confluent modulo local rule commutations: Theorem 6 below. Note, however, a subtlety: coalescence is confluent for conflict nets, but not necessarily for pre-nets; if coalescence fails, it can fail in many different ways.

Theorem 6. *If* $\mathcal{T} :: \Gamma$ *sequentializes to* $\Pi :: \Gamma$ *and* $\mathcal{T}$ *coalesces to* $\mathcal{T}'$ *then* $\mathcal{T}'$ *sequentializes to some* $\Pi' :: \Gamma$ *with* $\Pi' \simeq \Pi$.

Proof. We show that critical pairs of coalescence on deductive conflict nets converge modulo homeomorphism of the proofs at each link. The table shown right, pairing coalescence rewrites against each other, gives an overview: a number refers to a case considered in the proof, while a dot · indicates that rewrites do not form a critical pair. We do not for example consider

	$\parr$	$\oplus$	$\otimes$	&	1	#
#	4	4	·	·	·	5
1	·	·	·	·	·	
&	3	3	·	·		
$\otimes$	2	2	2			
$\oplus$	1	1				
$\parr$	1					

$$\Lambda X \leftsquigarrow \#(\Lambda X) \rightsquigarrow \#(\Lambda X \oplus Y)$$

a critical pair, as the (1) step leaves the link ΛX intact.

Case 1. A critical pair of ($\parr$) and ($\oplus$) steps converges by:

$$\begin{array}{ccc} \Lambda X Y Z & \rightsquigarrow & \Lambda X Y Z \oplus W \\ \wr & & \wr \\ \Lambda X \parr Y Z & \rightsquigarrow & \Lambda X \parr Y Z \oplus W \end{array}$$

The induced proofs differ by $\parr/\oplus$ commutation. The critical pairs for two ($\parr$) steps or two ($\oplus$) steps converge similarly.

Case 2. A critical pair of ($\otimes$) and ($\parr$) steps converges by:

$$\begin{array}{ccc} \frown(\mathcal{F};\Lambda X;\Omega Y Z W) & \rightsquigarrow & \frown(\mathcal{F};\Lambda X;\Omega Y Z \parr W) \\ \wr & & \wr \\ \frown(\mathcal{F};\Lambda\Omega X \otimes Y Z W) & \rightsquigarrow & \frown(\mathcal{F};\Lambda\Omega X \otimes Y Z \parr W) \end{array}$$

The induced proofs differ by a $\otimes/\parr$ commutation. The critical pairs of the ($\otimes$) step with the ($\oplus$) or ($\otimes$) step converge similarly.

Case 3. For the (&) step, the critical pair with the ($\oplus$) step converges as follows.

$$\begin{array}{ccc} \#(\Lambda X Z;\Lambda Y Z) & \rightsquigarrow & \Lambda X \& Y Z \\ \wr & & \\ \#(\Lambda X Z \oplus W;\Lambda Y Z) & & \wr \\ \wr & & \\ \#(\Lambda X Z \oplus W;\Lambda Y Z \oplus W) & \rightsquigarrow & \Lambda X \& Y Z \oplus W \end{array}$$

The induced proofs differ by a &/$\oplus$ commutation. The critical pair with the ($\parr$) step converges similarly, and there is no critical pair with the ($\otimes$) step. The (&) step cannot form a critical pair with itself since its # node must be binary.

Case 4. For the (#) step, the critical pair with the ($\oplus$) step converges as follows.

$$\begin{array}{ccc} \#(\mathcal{F}_X;\mathcal{F}_Y;\Lambda Z) & \rightsquigarrow & \#(\mathcal{F}_X;\mathcal{F}_Y;\Lambda Z \oplus W) \\ \wr & & \wr \\ \#(\,\#(\mathcal{F}_X);\#(\mathcal{F}_Y;\Lambda Z)) & \rightsquigarrow & \#(\,\#(\mathcal{F}_X);\#(\mathcal{F}_Y;\Lambda Z \oplus W)) \end{array}$$

The side-condition to the initial (#) step above is that some formula in ΛZ must be a subformula of Y in X&Y in the sequent Γ over which the cotrees are formed. This condition remains valid for $\Lambda Z \oplus W$: since Z occurs in $Z \oplus W$ and Y in X&Y, $Z \neq Y$; then if Z is a subformula of Y, so is $Z \oplus W$. The critical pair with the ($\parr$) step converges similarly, and no critical pairs are formed with the ($\otimes$), (&), and (1) steps.

Case 5. The critical pair of the (#) step with itself is

$$\begin{array}{ccc} \mathcal{T} = \#(\mathcal{F}_{XZ};\mathcal{F}_{XW};\mathcal{F}_{YZ};\mathcal{F}_{YW}) & \rightsquigarrow & \#(\#(\mathcal{F}_{XZ};\mathcal{F}_{XW});\#(\mathcal{F}_{YZ};\mathcal{F}_{YW})) \\ \wr & & \\ \#(\#(\mathcal{F}_{XZ};\mathcal{F}_{YZ});\#(\mathcal{F}_{XW};\mathcal{F}_{YW})) & & \end{array}$$

where the sequent Γ on which the cotrees are formed has subformulas X&Y and Z&W such that each of the links in $\mathcal{F}_{XZ}$ contains subformulas of X and Z, and similarly for $\mathcal{F}_{XW}$, $\mathcal{F}_{YZ}$, and $\mathcal{F}_{YW}$. We show by induction on the width of $\mathcal{T}$ that both sides sequentialize to homeomorphic proofs.

By assumption, $\mathcal{T}$ sequentializes, starting with (without loss of generality) the rightward step above. Since both the rightward and downward steps apply, no link intersects both X and Y, or both Z and W.

First we show that no $\mathcal{F}_i$ can be empty. Suppose $\mathcal{F}_{XZ}$ is empty; since the downward step from $\mathcal{T}$ applies, $\mathcal{F}_{YZ}$ is non-empty. Then $\frown(\mathcal{F}_{XZ};\mathcal{F}_{XW})$ may coalesce to a link containing W, but not Z&W, while $\frown(\mathcal{F}_{YZ};\mathcal{F}_{XW})$ may coalesce to a link containing Z&W (or Z), but not only W. This contradicts the assumption that $\mathcal{T}$ coalesces via the rightward step above.

Next, consider the case where each $\mathcal{F}_i$ is unary; then $\mathcal{F}_i$ is a proof link Π_i with conclusion Λ_i. Permuting ($\parr$) steps and ($\oplus$) steps above (&) steps (case 3. above) and (#) steps (case 4. above), we may assume the conclusions Λ_i are $\Lambda X Z$, $\Lambda X W$, $\Lambda Y Z$, and $\Lambda Y W$ respectively. The critical pair from $\mathcal{T}$ converges as follows.

$$\begin{array}{ccc} \#(\Lambda_{XZ};\Lambda_{XW};\Lambda_{YZ};\Lambda_{YW}) & \rightsquigarrow & \#(\#(\Lambda_{XZ};\Lambda_{XW});\#(\Lambda_{YZ};\Lambda_{YW})) \\ \wr & & \wr \\ \#(\#(\Lambda_{XZ};\Lambda_{YZ});\#(\Lambda_{XW};\Lambda_{YW})) & & \#(\Lambda X Z \& W;\Lambda Y Z \& W)) \\ \wr & & \wr \\ \#(\Lambda X \& Y Z;\Lambda X \& Y W) & \rightsquigarrow & \Lambda X \& Y Z \& W \end{array}$$

For the inductive step, we will show that if $\mathcal{F}_i$ is not unary, then $\#(\mathcal{F}_i)$ coalesces to Λ_i; whence the case is as above. We will only need to consider $\#(\mathcal{F}_{XZ};\mathcal{F}_{XW})$ as the case for $\#(\mathcal{F}_{YZ};\mathcal{F}_{YW})$ follows by symmetry.

Using the assumption that $\#(\mathcal{F}_{XZ};\mathcal{F}_{XW})$ coalesces (since $\mathcal{T}$ coalesces via the rightward step above), and ignoring ($\oplus$) and ($\parr$) steps as previously, consider the first (#) step. If this applies to the subformula Z&W, coalescence must be via a sequence

$$\#(\mathcal{F}_{XZ};\mathcal{F}_{XW}) \rightsquigarrow \#(\#(\mathcal{F}_{XZ});\#(\mathcal{F}_{XW})) \rightsquigarrow^* \#(\Lambda_{XZ};\Lambda_{XW})$$

Then each $\#(\mathcal{F}_i)$ coalesces to Λ_i, and the critical pair from $\mathcal{T}$ converges as above.

Otherwise, the first $(\#)$ step on $\#(\mathcal{F}_{XZ};\mathcal{F}_{XW})$ is on a subformula $U\&V$. Let it be the step

$$\#(\mathcal{F}_{XZ};\mathcal{F}_{XW}) \rightsquigarrow \#(\#(\mathcal{F}_{XZU};\mathcal{F}_{XWU});\#(\mathcal{F}_{XZV};\mathcal{F}_{XWV}))\,.$$

By the same reasoning as above, none of the $\mathcal{F}_i$ can be empty. Then the following $(\#)$ step for $Z\&W$ also applies:

$$\#(\mathcal{F}_{XZ};\mathcal{F}_{XW}) \rightsquigarrow \#(\#(\mathcal{F}_{XZU};\mathcal{F}_{XZV});\#(\mathcal{F}_{XWU};\mathcal{F}_{XWV}))\,.$$

By induction hypothesis, these two steps converge (modulo homeomorphism), which means there must be a sequence

$$\#(\#(\mathcal{F}_{XZU};\mathcal{F}_{XZV});\#(\mathcal{F}_{XWU};\mathcal{F}_{XWV})) \rightsquigarrow^* \#(\Lambda_{XZ};\Lambda_{XW})\,.$$

Then $\#(\mathcal{F}_{XZ})$ and $\#(\mathcal{F}_{XW})$ coalesce to Λ_{XZ} and Λ_{XW} respectively, and similarly for $\mathcal{F}_{YZ}$ and $\mathcal{F}_{YW}$, and the critical pair from $\mathcal{T}$ converges as above. □

5.4 Complexity of coalescence

Confluence is essential for coalescence to be tractable, since it becomes sufficient to verify just one coalescence sequence, instead of having to try every possibility. To establish that coalescence is P-time (polynomial-time decidable), we will look at:

A. the maximal length of a coalescence sequence, and

B. the cost of finding and executing the next step.

We will take the size of a conflict net to be the size of the cotree plus the size of the sequent. In what follows we are not aiming to give an efficient algorithm; we aim purely for a straightforward and accessible demonstration that coalescence is a reasonable correctness condition.

For A observe that the number of (1) and $(\#)$ steps is bounded linearly by the number of other steps: there is at most one (1) step created by each $(\otimes)$ and $(\#)$ step, and at most one $(\#)$ step for each $(\&)$ step. The other coalescence rewrites, $(\otimes)$, $(\parr)$, $(\&)$, and $(\oplus)$, are limited to one application per connective per link. This gives a quadratic bound to the length of any coalescence path.

For B, each step other than $(\#)$ involves only local pattern-matching and rewriting, which means a single traversal of the conflict net can find and execute such a step. A $(\#)$ redex in a pre-net $\mathcal{T} :: \Gamma$ can be found and coalesced as follows.

1. Find a node $\#(\Lambda_1;\ldots;\Lambda_n)$ whose children have all been coalesced to links Λ_i. This is linear-time.
2. For each Λ_i, for every subformula $X\&Y$ of Γ, mark X or Y when Λ_i intersects it. This is a simple walk from the formulas in Λ_i towards the roots of Γ; at most linear-time for Λ_i, and quadratic for $\#(\Lambda_1;\ldots;\Lambda_n)$.
3. Find a subformula $X\&Y$ with X and Y both marked, and either X or Y marked for every $i \leqslant n$; then partition $\#(\Lambda_1;\ldots;\Lambda_n)$ accordingly. This is linear-time.

Together, these algorithms for A and B give a time complexity of $\mathcal{O}(n^4)$, which justifies the following proposition:

Proposition 7. *Conflict net correctness is P-time.*

It is likely that more efficient algorithms are possible. Correctness of MLL proof nets is linear-time [15], and this likely extends to coalescing a subtree $\frown(\Lambda_1;\ldots;\Lambda_n)$, which additionally involves only $(\oplus)$ rewrite steps.

5.5 ALL coalescence

Coalescence for MALL departs from coalescence in purely additive linear logic [18]. Conflict nets restricted to ALL coincide with ALL proof nets, with a single $\#$ node covering the axiom links. However, coalescence for ALL has the following rewrite in place of the two rewrites $(\#)$ and $(\&)$.

$$\#(\mathcal{F};\Lambda X;\Lambda Y) \rightsquigarrow \#(\mathcal{F};\Lambda\, X\&Y) \qquad (A)$$

The difference between (A) and $(\&)$ is that the former allows a context $\mathcal{F}$, while the latter only applies to binary $\#$ nodes. The rewrite (A) is analogous to the $(\otimes)$ rewrite.

For ALL, where sequents consist of exactly two formulas, (A) is confluent [18] in the manner of Theorem 6: if one path succeeds, all paths do. But for sequents of three formulas or more, as occurs in MALL, it is not confluent. Consider the sequent

$$\overline{P}\&P\,,\ \overline{Q}\&Q\,,\ \overline{R}\&R$$

and the flat cotree $\mathcal{T}$ that takes all eight slices as links,

$$\mathcal{T} = \#(\overline{P}\overline{Q}\overline{R}\,;\ \overline{P}QR\,;\ P\overline{Q}R\,;\ PQ\overline{R}\,;\ \overline{P}\overline{Q}R\,;\ \overline{P}QR\,;\ P\overline{Q}\overline{R}\,;\ PQR)$$

This coalesces via (A) if the $\&$-formulas are treated in order, first $\overline{P}\&P$, then $\overline{Q}\&Q$, then $\overline{R}\&R$. However, coalescing one instance of $\overline{P}\&P$, one of $\overline{Q}\&Q$, and one of $\overline{R}\&R$ produces a deadlock:

$$\mathcal{T} \rightsquigarrow^*_A \#(\overline{P}\overline{Q}\overline{R}\,;\ \overline{P}\&P\ Q\ \overline{R}\,;\ \overline{P}\ \overline{Q}\&Q\ R\,;\ P\ \overline{Q}\ \overline{R}\&R\,;\ PQR)$$

No further (A) steps are possible, breaking confluence.

For MALL coalescence, replacing rule (A) with $(\#)$ and $(\&)$ solves this issue, yielding confluence (Theorem 6).

Where MLL contractibility $(\otimes)$ and $(\parr)$, and ALL coalescence $(\oplus)$ and (A), are top-down, the example above suggests top-down coalescence is impossible for MALL. Our coalescence combines one bottom-up rule $(\#)$ with an otherwise top-down procedure.

6. Cut elimination

In this section we extend the syntax of formulas and sequents with a binary ***cut*** connective $*$, restricted to occur in outermost position between dual formulas. Thus every cut formula has the form $A*\overline{A}$ with A cut-free.

Definition 8. A ***conflict net with cut*** is a conflict net with coalescence extended by

$$\frown(\mathcal{F};\Lambda\, A;\Lambda\, \overline{A}\,\Omega) :: \Gamma, A*\overline{A} \quad \rightsquigarrow \quad \frown(\mathcal{F};\Lambda\,\Omega) :: \Gamma \qquad (*)$$

This step deletes a cut formula $A*\overline{A}$ reached during coalescence. Thus successful coalescence results in a cut-free sequent, and a cut formula is never contracted in the context of a $(\&)$ step. The latter means cut formulas are not shared between slices. This is important for cut-elimination, which will manipulate cut formulas.

For sequentialization, the deductive variant of the above coalescence step combines the two subproofs with a cut rule. For proof translation, the cut rule translates to the same cotree as a $\otimes$ rule,

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\Pi_1 & \Pi_2\\ \Lambda, A & \overline{A}, \Omega\end{array}}{\Lambda, \Omega} \right]\!\!\right] \quad = \quad [\![\Pi_1]\!] \mathbin{\dot\frown} [\![\Pi_2]\!]$$

but the sequent over which the tree is formed is extended with $A*\overline{A}$. Thus, a proof Π of Γ with cuts translates to $[\![\Pi]\!] :: \Gamma, C_1, \ldots, C_n$ where the formulas $C_i = A_i*\overline{A}_i$ are the cut formulas of Π.

Since the $(*)$ rewrite is similar to the $(\otimes)$ rewrite, Theorem 6 extends straightforwardly to conflict nets with cut:

Theorem 9. *Sequentialization for conflict nets with cut is confluent modulo homeomorphism.*

6.1 Cut elimination

Cut elimination comprises four rewrite steps, three removing a cut (atomic, $\otimes/\parr$, and $\&/\oplus$), and a duplication step. Duplication corresponds to raising a cut rule over a $\&$ rule in sequent calculus, which duplicates the cut and its subproof:

$$\mathsf{cut}\,\dfrac{\Pi\quad \dfrac{\overline{A},\Delta,B\quad \overline{A},\Delta,C}{\overline{A},\Delta,B\&C}\,\&}{\Gamma,\Delta,B\&C}\quad\rightarrow\quad \dfrac{\mathsf{cut}\,\dfrac{\Pi\quad \Gamma,A\quad \overline{A},\Delta,B}{\Gamma,\Delta,B}\qquad \mathsf{cut}\,\dfrac{\Pi\quad \Gamma,A\quad \overline{A},\Delta,C}{\Gamma,\Delta,C}}{\Gamma,\Delta,B\&C}\,\&$$

For the corresponding duplication in conflict nets, the question is what to duplicate. The notion of subproof is not directly applicable to conflict nets, but it is accessible via coalescence: since any coalescence sequence (eventually) produces a sequentialization, any subtree or subforest that coalesces to a single link corresponds to a subproof in some sequentialization. Among subproofs suitable for duplication, we choose the smallest:

Definition 10. In a net $\frown(\mathcal{F}) :: \Gamma$ the ***kingdom*** of a subformula A of Γ, if it exists, is the smallest subforest $\mathcal{K}_A$ of $\mathcal{F}$ such that $\frown(\mathcal{K}_A) :: \Gamma$ coalesces to a link $\Delta\, A :: \Gamma$.

As in the MLL case, the kingdom of A corresponds to the smallest subproof with A in the conclusion in any sequentialization: see [4] for details. The kingdom of A may not exist, for example if A occurs in a formula $A\&B$ generated by a $\#$-rooted cotree in $\mathcal{F}$.[5]

The following notions provide the local structure needed for cut-elimination. A coforest $\mathcal{F}$ ***touches*** a formula A if a link in $\mathcal{F}$ intersects A. Non-empty coforests $\mathcal{F}_X$ and $\mathcal{F}_Y$ ***separate*** a subformula $X\otimes Y$, $X\&Y$, or $X\oplus Y$ if every tree in $\mathcal{F}_X$ touches X but not Y and every tree in $\mathcal{F}_Y$ touches Y but not X. A cotree ***generates*** a subformula Z if two immediate subtrees separate Z.

For example, if a cotree $\frown(\mathcal{F})$ generates a subformula $X\otimes Y$, then the root of $\frown(\mathcal{F})$ represents a sequent rule introducing the $\otimes$ between X and Y.

Definition 11 (***Cut reduction***)**.** A pre-net $\mathcal{T} :: \Gamma, A*\overline{A}$ where $\mathcal{T}$ generates the cut $A*\overline{A}$ reduces as follows. In each case, alternation may need to be restored.

Atomic step:

$$\frown(\mathcal{F};\underset{v}{\overline{P}}\,\underset{w}{P};\underset{x}{\overline{P}}\,\underset{y}{P}) :: \Gamma,\underset{w}{P}*\underset{x}{\overline{P}} \quad\rightarrow\quad \frown(\mathcal{F};\underset{v}{\overline{P}}\,\underset{y}{P}) :: \Gamma$$

where vertices $vwxy$ are underset, to avoid ambiguity.

Multiplicative step:

$$T :: \Gamma,(A\otimes B)*(\overline{B}\parr\overline{A}) \quad\rightarrow\quad T :: \Gamma, A*\overline{A}, B*\overline{B}$$

if $\mathcal{T}$ generates both $A\otimes B$ and $(A\otimes B)*(\overline{B}\parr\overline{A})$. This step applies analogously to the symmetric cut $(\overline{B}\parr\overline{A})*(A\otimes B)$.

Additive step:

$$\begin{array}{c}\frown(\#(\mathcal{F}_A;\mathcal{F}_B);\mathcal{F}) :: \Gamma,(A\&B)*(\overline{B}\oplus\overline{A})\\ \downarrow\\ \frown(\#(\mathcal{F}_A);\mathcal{F}) :: \Gamma, A*\overline{A}\end{array}$$

if $\mathcal{F}_A$ and $\mathcal{F}_B$ separate $A\&B$ and $\mathcal{F}$ does not touch $\overline{B}$. This step applies analogously to the symmetric cut $(\overline{B}\oplus\overline{A})*(A\&B)$.

Duplication step:

$$\begin{array}{c}\frown(\mathcal{F};\mathcal{K}_A;\#(\mathcal{F}_B;\mathcal{F}_C)) :: \Gamma,\Sigma,A*\overline{A}\\ \downarrow\\ \frown(\mathcal{F};\#(\frown(\mathcal{K}_A;\#(\mathcal{F}_B));\frown(\mathcal{K}_A;\#(\mathcal{F}_C)))) :: \Gamma,\Sigma,A*\overline{A},\Sigma,A*\overline{A}\end{array}$$

if

- $\mathcal{F};\mathcal{K}_A$ does not touch $\overline{A}$,
- $\mathcal{F}_B$ and $\mathcal{F}_C$ separate a formula $B\&C$ not a subformula of $\overline{A}$,

and where

- $\mathcal{K}_A$ is the kingdom of A and Σ are the cuts it dominates (defined below), and
- the left copy of $\Sigma, A*\overline{A}$ is associated with $\mathcal{F}_B$ and the left copy of $\mathcal{K}_A$; the right copy with $\mathcal{F}_C$ and the right copy of $\mathcal{K}_A$.

A simple example of cut duplication was shown in §1.2.4. Figure 6 shows an example with a larger duplicated kingdom. The ***domination*** order on a pre-net $\mathcal{T} :: \Gamma$ is generated by:

- the subformula ordering over Γ,
- the subtree ordering over $\mathcal{T}$,
- a link Λ in $\mathcal{T}$ dominates its Γ-subformulas,
- a subtree $\mathcal{T}'$ dominates a formula C if $\mathcal{T}'$ generates C.

Theorem 12. *Cut reduction on conflict nets preserves sequentialization modulo cut reduction and homeomorphism.*

Proof sketch. For each of the four steps $\mathcal{T} :: \Gamma \rightarrow \mathcal{T}' :: \Gamma$ we show that if $\mathcal{T} :: \Gamma$ sequentializes there are coalescence paths for $\mathcal{T}$ and $\mathcal{T}'$ that generate respective sequentializations Π and Π' such that Π cut-reduces to Π' modulo homeomorphism. □

Proposition 13. *A cut in a conflict net can always reduce.*

Proof sketch. The net coalesces, so a cut $A*\overline{A}$ must coalesce by a $(*)$ step $\frown(\mathcal{F};\Lambda A;\overline{A}\Omega) \rightsquigarrow \frown(\mathcal{F};\Lambda\Omega)$. Case distinctions are needed based on which coalescence steps produce ΛA and $\overline{A}\Omega$. □

6.2 Strong normalization

The idea that the additive connectives represent a choice for either their left or their right component is naturally captured by the (standard) notion of ***slice***: what remains of a proof net after removing one branch for each additive connective. For conflict nets, a natural notion of slice is one that chooses on $\#$ nodes. While we will not formalize such a notion, as we have no direct need for it, we will define a measure to count slices in this manner. Define the ***weight*** $\|\mathcal{T}\|$ of a cotree by:

$$\|\Lambda\| = 1 \qquad \begin{array}{rcl}\|\frown(\mathcal{T}_1;\ldots;\mathcal{T}_n)\| &=& \sum_i \|\mathcal{T}_i\|\\ \|\#(\mathcal{T}_1;\ldots;\mathcal{T}_n)\| &=& \prod_i \|\mathcal{T}_i\|\,.\end{array}$$

This measure will be used to establish that cut reduction is terminating. The idea is that upon duplication, the copies of a cut are pushed into separate collections of slices. In other words, the duplicated cuts are each shared amongst fewer slices than the original, giving a natural measure for termination.

Definition 14. In a conflict net the ***weight*** of a cut $A*\overline{A}$ is the weight $\|\mathcal{T}_A\|$ of the subtree $\mathcal{T}_A$ that generates it, and its ***size*** is the size $sz(A)$ of A. The ***weight*** of a net is the multiset comprising the pair $(\|\mathcal{T}_A\|, sz(A))$ for each cut.

Theorem 15. *Conflict net cut elimination is strongly normalizing.*

Proof. By Prop. 13 any cuts present in a net can be reduced. The weight of a net decreases upon cut reduction. Any reduction path may be completed to cut-free normal form. □

7. Complexity results

The following examples support the comparison table in §1.4.

Proposition 16. *Monomial nets are not efficient: proof translation has super-linear complexity.*

Proof sketch. Let $A_i = P_i\&P_i$, $T_k = \overline{A}_1\otimes(\ldots\otimes(\overline{A}_{k-1}\otimes\overline{A}_k)\ldots)$ and $\Gamma_k = A_1,\ldots,A_k,T_k$. One can construct a cut-free proof Π_k of Γ_k with all $\&$-rules at the bottom and $n = 2^k.k$ axiom rules whose monomial net θ_k must have n axiom links, each carrying a monomial with k distinct eigenvariables. Thus θ_k is a factor k larger than Π_k. □

Proposition 17. MALL *cut elimination is* EXPTIME.

[5] One can adjust the definition of kingdom to always exist; we have no need.

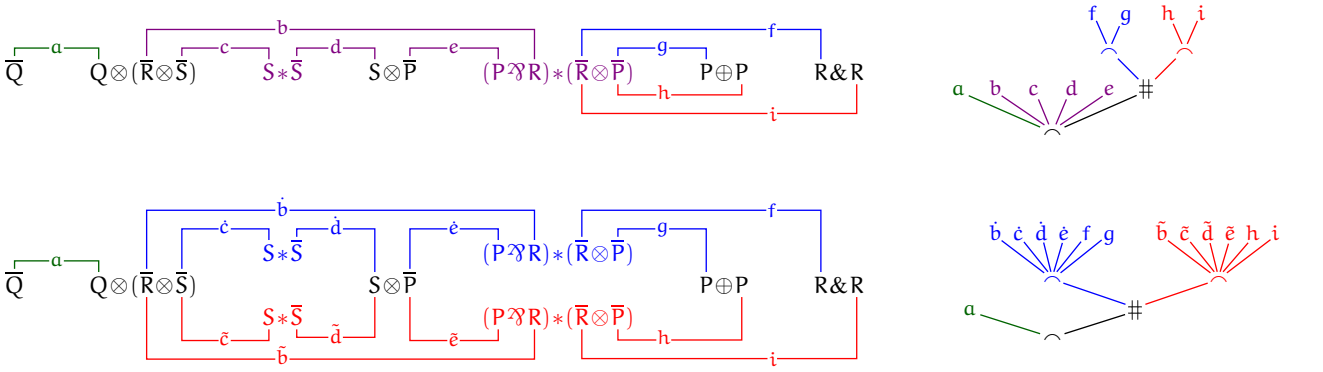


Figure 6. Illustrating cut duplication. Pattern-matching the definition of the duplication step: in the redex (the upper conflict net) A is $P\parr R$, its kingdom $\mathcal{K}_A$ is $bcde$, $\mathcal{F}_B$ is the cotree $f\frown g$, $\mathcal{F}_C$ is the cotree $h\frown i$, $\mathcal{F}$ is the cotree a (a single link), Γ is $\overline{Q}, Q\otimes(\overline{R}\otimes\overline{S}), S\otimes\overline{P}, P\oplus P, R\&R$, and Σ is $S*\overline{S}$. The kingdom $bcde$ is duplicated, once as $\dot{b}\dot{c}\dot{d}\dot{e}$ (next to $f\frown g$) and once as $\tilde{b}\tilde{c}\tilde{d}\tilde{e}$ (next to $h\frown i$).

Proof. Let $A_i = \overline{Q}_i\otimes\big((\overline{P}_i\otimes P_{i+1})\parr Q_{i+1}\big)$ and $\Gamma_n = P_1, Q_1, A_1\&A_1, \ldots, A_n\&A_n, \overline{Q}_{n+1}\otimes\overline{P}_{n+1}$. The $\otimes$s force a cut-free proof of Γ_n to stack n &-rules, introducing $A_1\&A_1, \ldots, A_n\&A_n$ bottom-up, for $6\times2^n-4$ axioms, exponential in n. Using cut, Γ_n has a $10n+2$ axiom proof, a cut-free proof of $P_1, Q_1, A_1, \ldots, A_n, \overline{Q}_{n+1}\otimes\overline{P}_{n+1}$ cut with n proofs of $\overline{A}_i, A_i\&A_i$. □

Proposition 18. *Monomial net sequentialization is* EXPTIME.

Proof sketch. Monomial nets θ_n on Γ_n defined above grow linearly in n; cut-free MALL proofs of Γ_n grow exponentially. □

Acknowledgements. Roberto Maieli for pointers on monomial and box nets. Referees for constructive feedback. EPSRC Project EP/K018868/1 *Efficient and Natural Proof Systems*. Wes Holliday for hosting Dominic at UC Berkeley.

References

[1] J. B. Almeida, J. S. Pinto, and M. Vilaça. A local graph-rewriting system for deciding equality in sum-product theories. *ENTCS*, 2007.

[2] P. B. Andrews. Refutations by matings. *Trans. Comp.*, 25(8), 1976.

[3] M. Bagnol. MALL proof equivalence is Logspace-complete, via binary decision diagrams. *TLCA*, 2015.

[4] G. Bellin and J. van de Wiele. Subnets of proof-nets in MLL$^-$. In *Advances in Linear Logic*, 1995.

[5] W. Bibel. An approach to a systematic theorem proving procedure in first-order logic. *Computing*, 12(1), 1974.

[6] K. Chaudhuri, D. Miller, and A. Saurin. Canonical sequent proofs via multi-focusing. In *IFIP*, 2008.

[7] P.-L. Curien and C. Faggian. An approach to innocent strategies as graphs. *Information and Computation*, 2012.

[8] V. Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation*. PhD thesis, Université Paris 7, 1990.

[9] P. J. De Naurois and V. Mogbil. Correctness of multiplicative additive proof structures is NL-complete. In *LICS*, 2008.

[10] K. Dosen and Z. Petric. Bicartesian coherence. *Studia Logica*, 71(3), 2002.

[11] C. Faggian and F. Maurel. Ludics nets, a game model of concurrent interaction. In *LICS*, 2005.

[12] D. Galmiche. Connection methods in linear logic and proof nets construction. *Theoretical Computer Science*, 232, 2000.

[13] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50(1), 1987.

[14] J.-Y. Girard. Proof-nets: the parallel syntax for proof-theory. *Logic and Algebra*, 1996.

[15] S. Guerrini. Correctness of multiplicative proof nets is linear. In *LICS*, 1999.

[16] W. Heijltjes. Proof nets for additive linear logic with units. In *LICS*, 2011.

[17] W. Heijltjes and R. Houston. No proof nets for MLL with units: Proof equivalence in MLL is PSPACE-complete. In *CSL-LICS*, 2014.

[18] W. Heijltjes and D. J. D. Hughes. Complexity bounds for sum–product logic via additive proof nets and Petri nets. In *LICS*, 2015.

[19] W. Heijltjes and L. Strassburger. Proof nets and semi-star-autonomous categories. *MSCS*, 2014.

[20] R. Houston. *Modelling linear logic without units*. PhD thesis, University of Manchester, 2008.

[21] H. Hu. Contractible coherence spaces and maximal maps. *Electronic Notes in Theoretical Computer Science*, 20, 1999.

[22] D. J. D. Hughes. Proofs Without Syntax. *Annals of Math.*, 143, 2006.

[23] D. J. D. Hughes. Towards Hilbert's 24$^{\text{th}}$ Problem: Combinatorial Proof Invariants. In *Proc. WOLLiC'06*, volume 165 of *LNCS*, 2006.

[24] D. J. D. Hughes. Abstract p-time proof nets for MALL: Conflict nets. http://arxiv.org/abs/0801.2421, 2008.

[25] D. J. D. Hughes and R. J. van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic (extended abstract). In *LICS*, 2003.

[26] D. J. D. Hughes and R. J. van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. *TOCL*, 6(4), 2005.

[27] A. Joyal and R. Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88, 1991.

[28] G. Kelly and S. Mac Lane. Coherence in closed categories. *Journal of Pure and Applied Algebra*, 1, 1971.

[29] O. Laurent. Polarized proof-nets: Proof-nets for LC. In *TLCA*, 1999.

[30] O. Laurent and R. Maieli. Cut elimination for monomial MALL proof nets. In *LICS*, 2008.

[31] R. Maieli. Retractile proof nets of the purely multiplicative and additive fragment of linear logic. In *LPAR*, 2007.

[32] H. G. Mairson and K. Terui. On the Computational Complexity of Cut-Elimination in Linear logic. In *ICTCS*, 2003.

[33] P.-A. Melliès. Game semantics in string diagrams. In *LICS*, 2012.