Proof nets for the grammatical base logic*

Michael Moortgat (OTS, Utrecht) and Dick Oehrle (U of Arizona)

Abstract

In this paper we present proof nets for the pure residuation logic **NL**$\Diamond$, the non-associative Lambek calculus augmented with a pair of unary multiplicatives $\Diamond, \Box$. The system **NL**$\Diamond$ occupies a central place in the categorial landscape in that systems like **L**, **NLP**, **LP** (=**MLL**) are reachable from it in terms of $\Diamond, \Box$ embeddings. A graph decomposition algorithm is given that singles out the **NL**$\Diamond$ proof nets from the wider class of proof structures and that establishes the correspondence between nets and Gentzen derivations. To the structurally free **NL**$\Diamond$ nets are then added structural links licensing more liberal resource management.

1 The pure residuation logic NL$\Diamond$

A fusion of ideas from Girard's [3] linear logic and from Lambek's type-logical tradition ([7, 8]) has led to a new organisation of categorial grammar logics. The grammatical architecture consists of three components. A core deductive engine, or *base logic*, characterizes invariants of grammatical composition, i.e. laws that are independent of non-logical properties of the composition relation. The *structural component* then defines packages of resource management postulates for various modes of composition. These packages are added to the base logic as 'plug-in' modules. They give rise to structural variation within languages (differential management for distinct *dimensions* of grammatical organization), and cross-linguistically. Most importantly, the *control* component provides systematic means for imposing structural constraints and for licensing structural relaxation under the explicit control of the logical constants $\Diamond, \Box$ — so-called 'modalities'. In [10] one finds an overview of these developments.

This paper is a contribution to the proof-theoretic study of this grammatical architecture. We characterize *proof nets* for the base logic, and establish a systematic way of introducing more liberal structural resource management options in these nets. As a prelude, we briefly review the essentials of the axiomatic and Gentzen presentations of the base logic **NL**$\Diamond$.

Consider the following formula language:

$$\mathcal{F} ::= \mathcal{A} \mid \Diamond\mathcal{F} \mid \Box\mathcal{F} \mid \mathcal{F}/\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F} \mid \mathcal{F}\backslash\mathcal{F}$$

Axiomatic presentation for **NL**$\Diamond$ is given by the residuation laws relating $\bullet$ to the left and right implications $\backslash$, and $\Diamond$ to $\Box$.

(REFL)	$A \to A$		
(TRANS)	$A \to C$	if $A \to B$	and $B \to C$
(RES-1)	$\Diamond A \to B$	iff $A \to \Box B$	
(RES-L)	$A \bullet B \to C$	iff $A \to C/B$	
(RES-R)	$A \bullet B \to C$	iff $B \to A\backslash C$	

Models for this language are based on frames $F = \langle W, R^2, R^3\rangle$ ([2, 9, 4]). One can think of W as a set of 'grammatical resources'. The 'accessibility' relations R^2, R^3 interpreting the unary and binary connectives then model 'grammatical composition'. The connectives $\Diamond$ and $\bullet$, in this setting, can be

*Versions of this paper were presented at ESSLLI'97 (Aix-en-Provence, August 1997), LACL'97 (Nancy, September 1997), and at the workshop 'Dynamic perspectives in logic and linguistics', Rome, October 1997. We thank these audiences for comments.

seen as existential modalities, decomposing a grammatical whole in its constituent part(s). The basic completeness result for **NL**$\Diamond$ holds for arbitrary frames — no structural restrictions are imposed on the composition relations R^2, R^3.

Restricting oneself to the $/, \bullet, \backslash$ fragment, one obtains the systems **L**, **NLP**, **LP** by adding packages of *structural postulates* to the base logic. The interpretation of R^3 in these systems is no longer structurally free: each postulate introduces a corresponding frame constraint.

$$\begin{array}{ll} (\textsc{comm}) & A \bullet B \rightarrow B \bullet A \\ (\textsc{ass-l}) & A \bullet (B \bullet C) \rightarrow (A \bullet B) \bullet C \\ (\textsc{ass-r}) & (A \bullet B) \bullet C \rightarrow A \bullet (B \bullet C) \end{array}$$

Global structural postulates such as the above destroy sensitivity to 'horizontal' (linear order) and/or 'vertical' (hierarchical grouping) aspects of grammatical organization. In the presence of the modalities $\Diamond, \Box$ more delicate options for grammatical resource management become available: global options can be replaced by lexically controlled choices. At the same time, it can be shown that in dropping the global structural postulates no expressivity has been lost. The embedding theorems of [5] make it possible to recover the expressivity of the systems **L**, **NLP**, **LP** in a controlled way via modalized fine-tunings of the global structural postulates above.

As an example of the embedding strategy, consider the following translation $\cdot^\sharp : \mathcal{F}(\mathbf{L}') \mapsto \mathcal{F}(\mathbf{NL}\Diamond)$. (We write $\mathbf{L}'$ for one of **L**, **NLP**, **LP**, and index the connectives of the source logic $\mathbf{L}'$ with '1', those of the target logic with '0'.)

$$\begin{array}{c} p^\sharp = p \\ (A \bullet_1 B)^\sharp = \Diamond(A^\sharp \bullet_0 B^\sharp) \\ (A/_1 B)^\sharp = \Box A^\sharp /_0 B^\sharp \\ (B \backslash_1 A)^\sharp = B^\sharp \backslash_0 \Box A^\sharp \end{array}$$

In [5] one finds embedding theorems of the following general shape.

$$\mathbf{L}' \vdash A \rightarrow B \quad \text{iff} \quad \mathbf{NL}\Diamond + \mathcal{R}_\Diamond \vdash A^\sharp \rightarrow B^\sharp$$

$\mathcal{R}_\Diamond$ here is the modalized version of the structural rule package discriminating between $\mathbf{L}'$ and **NL**. One obtains these modalized structural postulates by taking the image under $(\cdot)^\sharp$ of the global versions above.

$$\begin{array}{ll} \textsc{comm}_\Diamond & \Diamond(A \bullet_0 B) \rightarrow \Diamond(B \bullet_0 A) \\ \textsc{ass-l}_\Diamond & \Diamond(A \bullet_0 \Diamond(B \bullet_0 C)) \rightarrow \Diamond(\Diamond(A \bullet_0 B) \bullet_0 C) \\ \textsc{ass-r}_\Diamond & \Diamond(\Diamond(A \bullet_0 B) \bullet_0 C) \rightarrow \Diamond(A \bullet_0 \Diamond(B \bullet_0 C)) \end{array}$$

To establish decidability for **NL**$\Diamond$, one can move to a Gentzen presentation of the base logic. The Gentzen presentation makes use of a language of *structural* connectives parallel to the logical connectives. Structures are defined by the following grammar: $\mathcal{S} ::= \mathcal{F} \mid \langle \mathcal{S} \rangle \mid \mathcal{S} \circ \mathcal{S}$. Sequents are expressions $\Gamma \Rightarrow A$ with $\Gamma \in \mathcal{S}$ and $A \in \mathcal{F}$. We call the formula leaves of the antecedent structure term Γ together with the succedent formula A the terminal formulae of a sequent. The Gentzen presentation of **NL**$\Diamond$ is equivalent to the axiomatic presentation, and enjoys cut elimination ([9]).

$$\frac{}{A \Rightarrow A}\ \text{Ax} \qquad \frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C}\ \text{Cut}$$

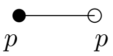
$$\frac{\Gamma[\langle A \rangle] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\ \Diamond L \quad \frac{\Gamma \Rightarrow A}{\langle \Gamma \rangle \Rightarrow \Diamond A}\ \Diamond R \qquad \frac{\langle \Gamma \rangle \Rightarrow A}{\Gamma \Rightarrow \Box A}\ \Box R \quad \frac{\Gamma[A] \Rightarrow B}{\Gamma[\langle \Box A \rangle] \Rightarrow B}\ \Box L$$

$$\frac{\Gamma \circ B \Rightarrow A}{\Gamma \Rightarrow A/B}\ /R \quad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[A/B \circ \Delta] \Rightarrow C}\ /L \qquad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta \circ B\backslash A] \Rightarrow C}\ \backslash L \quad \frac{B \circ \Gamma \Rightarrow A}{\Gamma \Rightarrow B\backslash A}\ \backslash R$$

$$\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma \circ \Delta \Rightarrow A \bullet B}\ \bullet R \qquad \frac{\Gamma[A \circ B] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C}\ \bullet L$$

2 Proof nets for the base logic

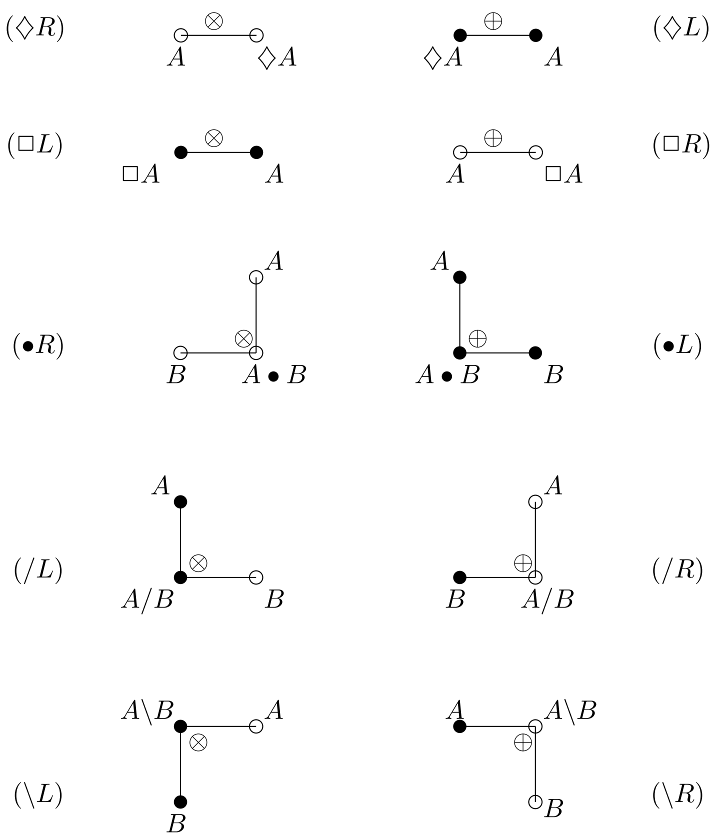
We will characterize proof nets for **NL**$\Diamond$ in terms of connected graphs built out of *polarized* vertices with a formula labeling defined on them. We use $\bullet$ for the input (source, antecedent, negative) polarity vertices, and $\circ$ for output (target, succedent, positive) polarity vertices. (See [11, 6] for **L** proof nets — we try to be consistent with the notation/terminology.) The elementary net ('axiom link') connects a single input node to a single output node. Its nodes are labeled with a prime formula (literal).

(Ax) 

To build bigger nets out of smaller ones, we have building blocks corresponding to the Gentzen rules of use and proof. These building blocks ('logical links') are of two types:

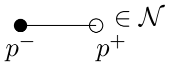
- $\otimes$-type ('tensor') links: $\Diamond R$, $\Box L$, $\bullet R$, $\backslash L$, $/L$;
- $\oplus$-type ('cotensor', 'par') links: $\Diamond L$, $\Box R$, $\bullet L$, $\backslash R$, $/R$

The conclusion of the logical links is labeled with a complex formula determined by the subformula(e) labeling the premise(s), by the polarities of the vertices involved, and by the type ($\otimes$ or $\oplus$) of the link. (The order of the premises is important!)

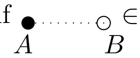


We now give an inductive definition of $\mathcal{N}$, the set of proof nets for the base logic **NL**$\Diamond$. Proof nets are built out of elementary nets according to the clauses below. Clauses (b) and (c) build a bigger net out of (a) smaller one(s) by putting in a matching $\otimes, \oplus$ pair. The labeling of the new source and target nodes is determined by the typing rules for the $\otimes, \oplus$ links. Clauses (d) and (e) replace a source-$\oplus$ link by the (left, right) residual target-$\oplus$ link. These clauses do not affect the parity of $\otimes, \oplus$ links. But they offer alternative perspectives on a given net by focusing on different source/target nodes.

(a) Axiom:

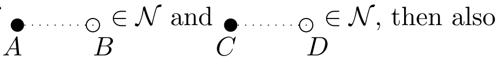


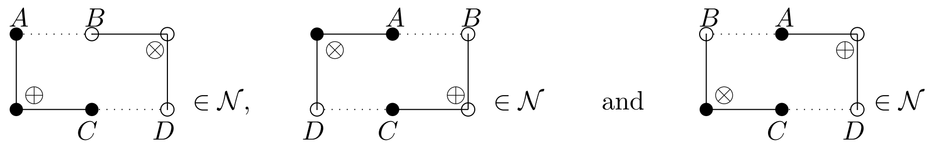
(b) Unary: if



$\in \mathcal{N}$, then also

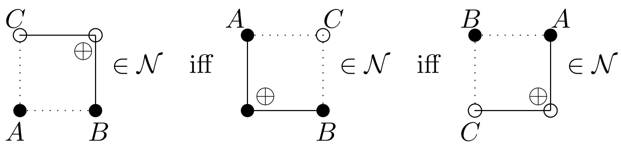
(c) Binary: if





(d) Unary shifts:

(e) Binary shifts:



2.1 A graph rewriting system for base nets

It is handy to have a *term language* to talk about the assembly of **NL**$\Diamond$ proof nets. We propose the combinator system N for that purpose.

$$1_p : p \to p$$

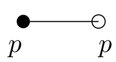
$$\frac{f : A \to B}{f^{\Diamond} : \Diamond A \to \Diamond B} \qquad \frac{f : A \to B}{f^{\Box} : \Box A \to \Box B} \qquad \frac{f : \Diamond A \to B}{\mu(f) : A \to \Box B} \qquad \frac{g : A \to \Box B}{\mu^{-1}(g) : \Diamond A \to B}$$

$$\frac{f : A \to B \quad g : C \to D}{f/g : A/D \to B/C} \qquad \frac{f : A \to B \quad g : C \to D}{f \cdot g : A \bullet C \to B \bullet D} \qquad \frac{f : A \to B \quad g : C \to D}{f\backslash g : B\backslash C \to A\backslash D}$$

$$\frac{f : A \bullet B \to C}{\beta(f) : A \to C/B} \qquad \frac{g : A \to C/B}{\beta^{-1}(g) : A \bullet B \to C} \qquad \frac{f : A \bullet B \to C}{\gamma(f) : B \to A\backslash C} \qquad \frac{g : B \to A\backslash C}{\gamma^{-1}(g) : A \bullet B \to C}$$

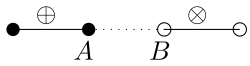
The clauses of the inductive definition of proof nets for **NL**$\Diamond$ can now be formulated in terms of the N combinators.

(a) Axiom:

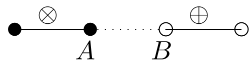


$1_p : p \to p$

(b) Unary:

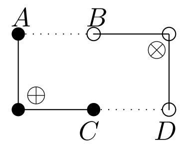


$$\frac{f : A \to B}{f^{\Diamond} : \Diamond A \to \Diamond B}$$

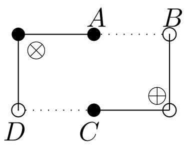


$$\frac{f : A \to B}{f^{\Box} : \Box A \to \Box B}$$

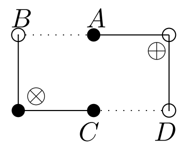
(c) Binary:



$$\frac{f : A \to B \quad g : C \to D}{f \cdot g : A \bullet C \to B \bullet D}$$



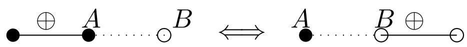
$$\frac{f : A \to B \quad g : C \to D}{f/g : A/D \to B/C}$$



$$\frac{f : A \to B \quad g : C \to D}{f\backslash g : B\backslash C \to A\backslash D}$$

(d) Unary shift:

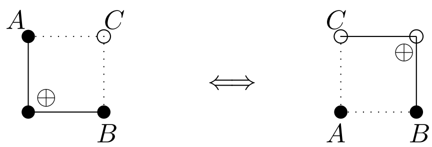
$$f : \Diamond A \to B \qquad \Rightarrow \qquad \mu(f) : A \to \Box B$$

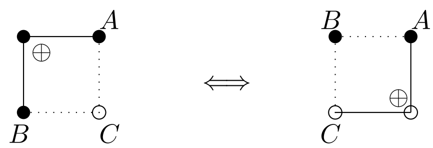


$$\mu^{-1}(g) : \Diamond A \to B \qquad \Leftarrow \qquad g : A \to \Box B$$

(e) Binary shift:

$$f : A \bullet B \to C \qquad \Rightarrow \qquad \beta(f) : A \to C/B \qquad\qquad f : A \bullet B \to C \qquad \Rightarrow \qquad \gamma(f) : B \to A\backslash C$$





$$\beta^{-1}(g) : A \bullet B \to C \qquad \Leftarrow \qquad g : A \to C/B \qquad\qquad \gamma^{-1}(g) : A \bullet B \to C \qquad \Leftarrow \qquad g : B \to A\backslash C$$

2.2 From sequents to nets

The proof that every **NL**$\Diamond$ Gentzen derivation π can be translated into a proof net $\|\pi\|$ is an easy induction on the structure of π.

$$(Ax) \qquad \| \ p \Rightarrow p \ \| \quad = \quad 1_p$$

$$(\Box R) \qquad \left\| \frac{\dfrac{\pi}{\langle\Gamma\rangle \Rightarrow A}}{\Gamma \Rightarrow \Box A} \Box R \right\| \quad = \quad \mu(\|\pi\|)$$

$$(\Diamond R) \qquad \left\| \frac{\dfrac{\pi}{\Gamma \Rightarrow A}}{\langle\Gamma\rangle \Rightarrow \Diamond A} \Diamond R \right\| \quad = \quad (\|\pi\|)^{\Diamond}$$

$$(\bullet R) \qquad \left\| \frac{\dfrac{\pi_1}{\Gamma \Rightarrow A} \quad \dfrac{\pi_2}{\Delta \Rightarrow B}}{\Gamma \circ \Delta \Rightarrow A \bullet B} \bullet R \right\| \quad = \quad \|\pi_1\| \cdot \|\pi_2\|$$

$$(/R) \qquad \left\| \frac{\dfrac{\pi}{\Gamma \circ B \Rightarrow A}}{\Gamma \Rightarrow A/B} /R \right\| \quad = \quad \beta(\|\pi\|)$$

$$(\backslash R) \qquad \left\| \frac{\dfrac{\pi}{B \circ \Gamma \Rightarrow A}}{\Gamma \Rightarrow B\backslash A} \backslash R \right\| \quad = \quad \gamma(\|\pi\|)$$

$$(\Box L) \qquad \left\| \frac{\dfrac{\pi}{\Gamma[A] \Rightarrow C}}{\Gamma[\langle\Box A\rangle] \Rightarrow B} \Box L \right\| \quad = \quad \sigma(\mu^{-1}(\rho(\|\pi\|)^{\Box})) \quad \text{where} \quad \rho \in (\mu \cup \beta \cup \gamma)^*, \sigma = \rho^{-1}$$

$$(/L) \qquad \left\| \frac{\dfrac{\pi_1}{\Delta \Rightarrow B} \quad \dfrac{\pi_2}{\Gamma[A] \Rightarrow C}}{\Gamma[A/B \circ \Delta] \Rightarrow C} /L \right\| \quad = \quad \sigma(\beta^{-1}(\rho(\|\pi_2\|)/\,\|\pi_1\|)) \quad \text{where} \quad \rho \in (\mu \cup \beta \cup \gamma)^*, \sigma = \rho^{-1}$$

$$(\backslash L) \qquad \left\| \frac{\dfrac{\pi_1}{\Delta \Rightarrow B} \quad \dfrac{\pi_2}{\Gamma[A] \Rightarrow C}}{\Gamma[\Delta \circ B\backslash A] \Rightarrow C} \backslash L \right\| \quad = \quad \sigma(\gamma^{-1}(\|\pi_1\| \backslash \rho(\|\pi_2\|))) \quad \text{where} \quad \rho \in (\mu \cup \beta \cup \gamma)^*, \sigma = \rho^{-1}$$

In the $(\Box L)$, $(/L)$, $(\backslash L)$ cases, $\rho \in (\mu \cup \beta \cup \gamma)^*$ is a sequence of SHIFT steps, displaying the A target on the left by moving context formulae to the right. $\sigma = \rho^{-1}$ puts the context formulae back.

For the $(\Diamond L)$ and $(\bullet L)$ rules, the net for the premise and for the conclusion are identical.

2.3 From nets to sequents

For the other side of the correspondence between nets and sequents, we first characterize the set of **NL**$\Diamond$ proof *structures* — a superset of the proof *nets*. We then show that by putting N in reverse mode (i.e. reading it as a graph decomposition algorithm) we obtain a decision procedure that picks out the proof nets among the proof structures.

Consider first the relation between an **L** sequent $A_1, \ldots, A_n \Rightarrow B$ and a proof net for it. An **L** proof structure consists of the row of formula decomposition trees for $(A_1)^\bullet, \ldots, (A_n)^\bullet, (B)^\circ$ together with an axiom linking over the literals. This graph qualifies as a proof net if it satisfies the conditions of acyclicity, connectedness and planarity. For **NL**$\Diamond$ proof structures/nets, information about the structural configuration of the antecedent $A_1, \ldots, A_n$ formulas has to be added. An **NL**$\Diamond$ proof structures for a sequent $\Gamma \Rightarrow B$ is a graph consisting of the following components:

- the row of formula decomposition trees of the terminal formulae $(A_1)^\bullet, \ldots, (A_n)^\bullet$, $(B)^\circ$ together with
- a set of axiom links providing a pairwise matching of the leaves of the formula decomposition trees (signed literals p^-, p^+), and
- a *structural* linking of the roots of the $(A_i)^\bullet$ trees in terms of the antecedent $\oplus$ links ($\Diamond L$) and ($\bullet L$).

For convenience, we repeat the formula decomposition clauses.

$$\frac{(A)^\bullet \quad (B)^\circ}{(A/B)^\bullet} \otimes \quad \frac{(B)^\bullet \quad (A)^\circ}{(A/B)^\circ} \oplus \qquad \frac{(A)^\bullet \quad (B)^\bullet}{(A \bullet B)^\bullet} \oplus \quad \frac{(B)^\circ \quad (A)^\circ}{(A \bullet B)^\circ} \otimes \qquad \frac{(B)^\circ \quad (A)^\bullet}{(B\backslash A)^\bullet} \otimes \quad \frac{(A)^\circ \quad (B)^\bullet}{(B\backslash A)^\circ} \oplus$$

$$\frac{(A)^\bullet}{(\Diamond A)^\bullet} \oplus \quad \frac{(A)^\circ}{(\Diamond A)^\circ} \otimes \qquad \frac{(A)^\bullet}{(\Box A)^\bullet} \otimes \quad \frac{(A)^\circ}{(\Box A)^\circ} \oplus$$

The construction of a proof structure has a deterministic phase (unfolding the terminal formulas) and two non-deterministic components: establishing the matching of literals, and establishing the structural assembly of terminals. A proof structure can fail to be a net because of an illegitimate axiom link: in the presence of *unary* connectives $\Diamond, \Box$ the usual check for cycles will not be enough to detect wrong axiom links. A proof structure can fail to be a net also because of an illegitimate structural configuration of the terminals. Again, the correctness criteria as developed for $\mathbf{L}(\mathbf{P})$ are not much help: they do not take the structural configuration of terminal formulae into account.

In order to single out $\mathbf{NL}\Diamond$ proof nets from proof structures, we will use the system N in 'reverse mode', as a graph decomposition algorithm. The backward-chaining (outside-in) perspective as a matter of fact provides a straightforward decision procedure. To decompose a proof structure $A \to B$ one matches $A \to B$ against the reduction rules MON. If the pattern-matching succeeds, a pair of $\otimes, \oplus$ links is removed. One repeats the decomposition process on the premise(s). If $A \to B$ does not match a MON reduction rule, one tries the finitely many SHIFT alternatives. Decomposition is successful if it bottoms out on elementary nets $p \to p$. From the decomposition combinator f one easily obtains a Gentzen derivation of $A \Rightarrow B$.

We have to prove that the proposed graph decomposition algorithm is *complete*, i.e. that it will recognize all $\mathbf{NL}\Diamond$ nets among the candidate proof structures. This comes down to showing the equivalence of N with the Gentzen presentation of $\mathbf{NL}\Diamond$. We do this indirectly, and compare N with L and D — the Lambek and Došen axiomatizations of $\mathbf{NL}$ extended with $\Diamond, \Box$. For D and L the equivalence with the Gentzen presentation has been established. From RES and MON one easily derives the (CO)UNIT laws. So what remains to be shown is that TRANS is an admissible rule in N.

N	ID	RES		MON
D	ID	TRANS	(CO)UNIT	MON
L	ID	TRANS	RES	

2.4 N: admissibility of TRANS

The proof is a categorical version of Gentzen's *Hauptsatz*. One proceeds by induction on the complexity of TRANS, measured in terms of the number of connectives. One shows that each application of TRANS (itself derived without the use of TRANS) can be replaced by one or two TRANS inferences of lower complexity. The base cases of the algorithm are instances of TRANS where either $f = 1$, so $g \circ f = g \circ 1 = g$, or $g = 1$, so $g \circ f = 1 \circ f = f$.

Two types of elimination steps can be distinguished. In the 'principal' cases, the cut formula ($\Diamond A, \Box A, A \bullet B, A/B, B\backslash A$) is introduced by a MON step in the two premises of the TRANS inference. One replaces the cut on a complex formula by cut(s) on the subformula(e), decreasing complexity. See CASE 1. Where the conditions for the principal case are not met, one shows that TRANS can be permuted upwards in a complexity decreasing way. See CASE 2.

Below we give the elimination steps for $\bullet$ and $/$. The $\backslash$ cases are similar. We leave it to the reader to work out the $\Diamond$ (similar to $\bullet$) and $\Box$ (similar to $/$) cases.

$$\dfrac{f : A \to B \quad g : B \to C}{g \circ f : A \to C}\ \text{TRANS (complexity } |A| + |B| + |C|)$$

CASE 1: Principal cuts.

– $f = f_1 \cdot f_2$, $g = g_1 \cdot g_2$.

$$\dfrac{\dfrac{A \to B \quad C \to D}{A \bullet C \to B \bullet D} \quad \dfrac{B \to E \quad D \to F}{B \bullet D \to E \bullet F}}{A \bullet C \to E \bullet F} \quad \rightsquigarrow \quad \dfrac{\dfrac{A \to B \quad B \to E}{A \to E} \quad \dfrac{C \to D \quad D \to F}{C \to F}}{A \bullet C \to E \bullet F}$$

– $f = f_1/f_2$, $g = g_1/g_2$.

$$\dfrac{\dfrac{A \to B \quad C \to D}{A/D \to B/C} \quad \dfrac{B \to E \quad F \to C}{B/C \to E/F}}{A/D \to E/F} \quad \rightsquigarrow \quad \dfrac{\dfrac{A \to B \quad B \to E}{A \to E} \quad \dfrac{F \to C \quad C \to D}{F \to D}}{A/D \to E/F}$$

CASE 2: Permutation cases. We classify these according to the shape of the cut formula (product, implication, atom). In the equations below, we write $f[g]$ for a proofterm f with a distinguished occurrence of a subterm g.

– The main connective of the cut formula is $\bullet$. In the left premise $f[h_1 \cdot h_2]$, the cut formula must match the MON rule for $\bullet$, immediately or after a number of steps. The TRANS inference is moved upwards and decomposed into TRANS inferences on h_1 and h_2.

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{h_1 : E \to B \quad h_2 : F \to C}{h_1 \cdot h_2 : E \bullet F \to B \bullet C}\\ \vdots\end{array}}{f[h_1 \cdot h_2] : A \to B \bullet C} \quad g : B \bullet C \to D}{g \circ f[h_1 \cdot h_2] : A \to D} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\dfrac{\dfrac{E \to B \quad \dfrac{\dfrac{\dfrac{F \to C \quad \dfrac{B \bullet C \to D}{C \to B\backslash D}}{F \to B\backslash D}}{B \bullet F \to D}}{B \to D/F}}{E \to D/F}}{E \bullet F \to D}\\ \vdots\end{array}}{A \to D}$$

$$g \circ f[h_1 \cdot h_2] = f[\gamma^{-1}(\gamma(\beta^{-1}(\beta(g) \circ h_1)) \circ h_2)]$$

– The main connective of the cut formula is $/$. In the right premise $g[h_1/h_2]$, the cut formula must match the MON rule for $/$, immediately or after a number of steps. The TRANS inference is moved upwards and decomposed into TRANS inferences on h_1 and h_2.

$$\dfrac{f : A \to B/C \qquad \dfrac{\begin{array}{c}\dfrac{h_1 : B \to E \quad h_2 : F \to C}{h_1/h_2 : B/C \to E/F} \\ \hline \vdots \end{array}}{g[h_1/h_2] : B/C \to D}}{g[h_1/h_2] \circ f : A \to D} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\dfrac{\dfrac{\dfrac{F \to C \qquad \dfrac{\dfrac{\dfrac{A \to B/C}{A \bullet C \to B} \quad B \to E}{A \bullet C \to E}}{C \to A\backslash E}}{F \to A\backslash E}}{A \bullet F \to E}}{A \to E/F} \\ \hline \vdots \end{array}}{A \to D}$$

$$g[h_1/h_2] \circ f = g[\beta(\gamma^{-1}(\gamma(h_1 \circ \beta^{-1}(f)) \circ h_2)]$$

- The cut formula is atomic. The TRANS inference is moved upwards with a decrease in complexity, because D is a subformula of C.

$$\dfrac{f : A \to B \quad \dfrac{\begin{array}{c} h : B \to D \\ \hline \vdots \end{array}}{g[h] : B \to C}}{g[h] \circ f : A \to C} \quad \rightsquigarrow \quad \dfrac{\begin{array}{c}\dfrac{A \to B \quad B \to D}{A \to D} \\ \hline \vdots \end{array}}{A \to C}$$

$$g[h] \circ f = g[h \circ f]$$

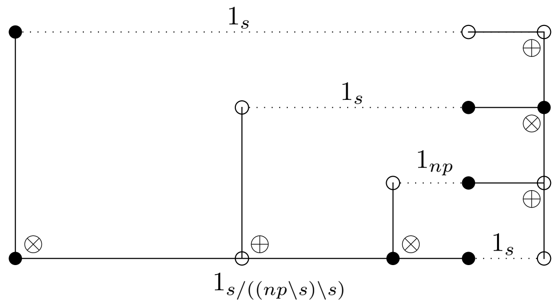
2.5 Illustrations

The $\mathbf{NL}\Diamond$ sequent $(s/(np\backslash s)) \bullet ((s/(np\backslash s))\backslash s) \to s$ has two readings, i.e. two ways of matching up the literals with axiom links. The decomposition combinators for these two readings are given below.

$$\dfrac{\dfrac{s \to s \qquad \dfrac{\dfrac{\dfrac{\dfrac{np \to np \quad s \to s}{np\backslash s \to np\backslash s}}{np \bullet (np\backslash s) \to s}}{np \to s/(np\backslash s)} \qquad s \to s}{(s/(np\backslash s))\backslash s \to np\backslash s}}{s/(np\backslash s) \to s/((s/(np\backslash s))\backslash s)}}{(s/(np\backslash s)) \bullet ((s/(np\backslash s))\backslash s) \to s} \qquad \dfrac{\dfrac{\dfrac{s \to s \qquad \dfrac{np \to np \quad s \to s}{np\backslash s \to np\backslash s}}{s/(np\backslash s) \to s/(np\backslash s)} \qquad s \to s}{(s/(np\backslash s))\backslash s \to (s/(np\backslash s))\backslash s}}{(s/(np\backslash s)) \bullet ((s/(np\backslash s))\backslash s) \to s}$$

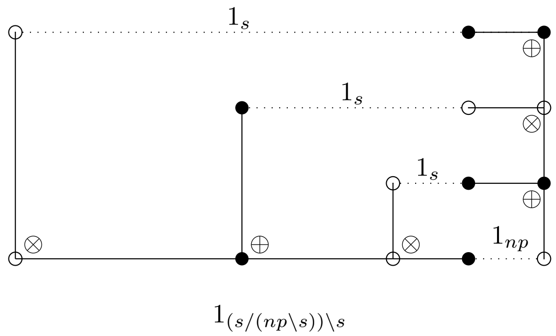
$$\beta^{-1}(1_s/(\beta(\gamma^{-1}(1_{np}\backslash 1_s))\backslash 1_s)) \qquad \gamma^{-1}((1_s/(1_{np}\backslash 1_s))\backslash 1_s)$$

Ignoring SHIFT moves, we can display the literal matching together with the $\otimes, \oplus$ matching for the first reading (subject wide scope) as follows.



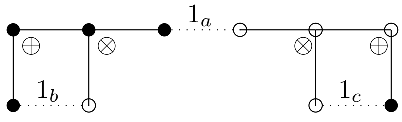
$$\dfrac{\dfrac{s \to s \quad \dfrac{\dfrac{\dfrac{\dfrac{np \to np \quad s \to s}{np\backslash s \to np\backslash s}}{np \bullet (np\backslash s) \to s}}{np \to s/(np\backslash s)} \quad s \to s}{(s/(np\backslash s))\backslash s \to np\backslash s}}{s/(np\backslash s) \to s/((s/(np\backslash s))\backslash s)}}{(s/(np\backslash s)) \bullet ((s/(np\backslash s))\backslash s) \to s}$$

Similarly for the VP wide scope reading. (The net is rotated 180 %: output vertex on the left, input on the right.)



$$\dfrac{\dfrac{\dfrac{s \to s \quad \dfrac{np \to np \quad s \to s}{np\backslash s \to np\backslash s}}{s/(np\backslash s) \to s/(np\backslash s)} \quad s \to s}{(s/(np\backslash s))\backslash s \to (s/(np\backslash s))\backslash s}}{(s/(np\backslash s)) \bullet ((s/(np\backslash s))\backslash s) \to s}$$

Next, consider the arrow $(a/b) \bullet b \to (a \bullet c)/c$. The net for this sequent is depicted below.

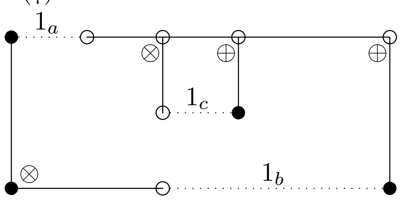
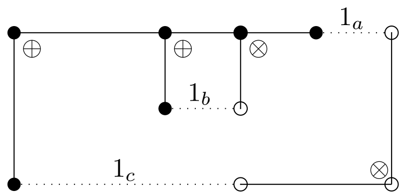


There is two ways of decomposing this net, cf the combinators below.

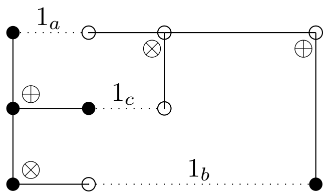
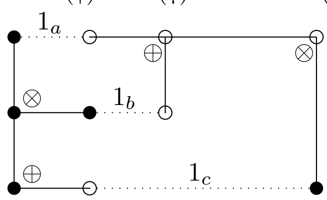
$$\dfrac{\dfrac{\dfrac{\dfrac{a \to a \quad c \to c}{(a \bullet c) \to (a \bullet c)}}{a \to ((a \bullet c)/c)} \quad b \to b}{(a/b) \to (((a \bullet c)/c)/b)}}{((a/b) \bullet b) \to ((a \bullet c)/c)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{a \to a \quad b \to b}{(a/b) \to (a/b)}}{((a/b) \bullet b) \to a} \quad c \to c}{(((a/b) \bullet b) \bullet c) \to (a \bullet c)}}{((a/b) \bullet b) \to ((a \bullet c)/c)}$$

$\beta^{-1}(\beta(1_a \cdot 1_c)/1_b)$ $\qquad$ $\beta(\beta^{-1}(1_a/1_b) \cdot 1_c)$

In order to take apart the net (identify matching $\otimes, \oplus$ pairs), one has to switch perspective. This can be done by moving the left or right $\oplus$ to the other side. The rightward move brings a matching $\otimes, \oplus$ pair in focus, cf (†). The leftward shift offers an alternative way of bringing a matching $\otimes, \oplus$ pair in focus, cf (‡).

(†)   (‡)

Performing a residuation shift on the embedded nets of (†) and (‡) we obtain (†') and (‡').

(†')   (‡')

Notice that the way we are using N here as an algorithm testing whether a *given* proof structure is in fact a $\mathbf{NL}\Diamond$ net, there is no need in pursuing these two decomposition paths. But the examples show that running the N algorithm in backward chaining fashion in fact can accomplish more: given a structural configuration Γ with terminal formulae $A_1, \ldots, A_n$ and a goal formula B, it computes all possible axiom linkings and structural $\otimes, \oplus$ matchings over the unfoldings of the $(A_i)^\bullet$ and $(B)^\circ$. If one is interested in this 'parsing' perspective on N, it is essential to keep the choice between SHIFT and REDUCE moves non-deterministic, as the following simple example shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{p \rightarrow p}{\Diamond p \rightarrow \Diamond p}}{p \rightarrow \Box\Diamond p}}{\Diamond p \rightarrow \Diamond\Box\Diamond p}}{\Box\Diamond p \rightarrow \Box\Diamond\Box\Diamond p} \qquad \dfrac{\dfrac{\dfrac{\dfrac{p \rightarrow p}{\Diamond p \rightarrow \Diamond p}}{\Box\Diamond p \rightarrow \Box\Diamond p}}{\Diamond\Box\Diamond p \rightarrow \Diamond\Box\Diamond p}}{\Box\Diamond p \rightarrow \Box\Diamond\Box\Diamond p}$$

$$((\mu((1_p)^\Diamond))^\Diamond)^\Box \qquad \mu((((1_p)^\Diamond)^\Box)^\Diamond)$$

3 Structural plug-ins

We have seen above how the base logic $\mathbf{NL}\Diamond$ can be extended with structural options for grammatical resource management. In the axiomatic presentation these take the form of structural postulates (i.e. non-logical axioms). We give some illustrative examples below. There is a systematic correspondence between these postulates and structural rules in the Gentzen presentation: for a postulate $A \rightarrow B$, the Gentzen structural rule is in fact a compiled Cut on the 'structural translations' A°, B° of the $\Diamond, \bullet$ formulas A, B.

$$\dfrac{A^\circ \Rightarrow B \quad \dfrac{\Gamma[B^\circ] \Rightarrow C}{\Gamma[B] \Rightarrow C}}{\Gamma[A^\circ] \Rightarrow C}\ \text{Cut}$$

	POSTULATE	GENTZEN RULE	
(COMM)	$A \bullet B \rightarrow B \bullet A$	$\Gamma[\Delta_1 \circ \Delta_2] \Rightarrow A$	if $\Gamma[\Delta_2 \circ \Delta_1] \Rightarrow A$
(ASS-L)	$A \bullet (B \bullet C) \rightarrow (A \bullet B) \bullet C$	$\Gamma[\Delta_1 \circ (\Delta_2 \circ \Delta_3)] \Rightarrow A$	if $\Gamma[(\Delta_1 \circ \Delta_2) \circ \Delta_3] \Rightarrow A$
(ASS-R)	$(A \bullet B) \bullet C \rightarrow A \bullet (B \bullet C)$	$\Gamma[(\Delta_1 \circ \Delta_2) \circ \Delta_3] \Rightarrow A$	if $\Gamma[\Delta_1 \circ (\Delta_2 \circ \Delta_3)] \Rightarrow A$
$(K1)$	$\Diamond(A \bullet B) \rightarrow \Diamond A \bullet B$	$\Gamma[\langle \Delta_1 \circ \Delta_2 \rangle] \Rightarrow A$	if $\Gamma[\langle \Delta_1 \rangle \circ \Delta_2] \Rightarrow A$
$(K2)$	$\Diamond(A \bullet B) \rightarrow A \bullet \Diamond B$	$\Gamma[\langle \Delta_1 \circ \Delta_2 \rangle] \Rightarrow A$	if $\Gamma[\Delta_1 \circ \langle \Delta_2 \rangle] \Rightarrow A$
(K)	$\Diamond(A \bullet B) \rightarrow \Diamond A \bullet \Diamond B$	$\Gamma[\langle \Delta_1 \circ \Delta_2 \rangle] \Rightarrow A$	if $\Gamma[\langle \Delta_1 \rangle \circ \langle \Delta_2 \rangle] \Rightarrow A$

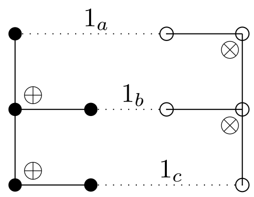
Notice that structural inferences, in the Gentzen presentation, are *explicit* steps in the derivation. As a matter of fact, we insist on this explicit treatment: typically, in the linguistic applications, we are interested in one-way structural inferences of the form $A \to B$ rather than in symmetric postulates $A \leftrightarrow B$ that allow a 'sugared' Gentzen presentation with implicit structural rules. Our strategy for extending the proof nets of the base logic with structural plug-ins will be explicit in the same way. (Compare the work of Bellin and Fleury [1] on 'braided' proof nets for MLL where the exchange rule gets an explicit representation.) For every structural option, we add to the inductive definition of nets (and to the graph decomposition algorithm N):

- a structural *axiom* σ as a new elementary graph decomposition clause
- a structural *cut* link producing a net $\sigma(f, g)$ out of nets f, g

The resulting nets offer two perspectives on the management of the grammatical resources:

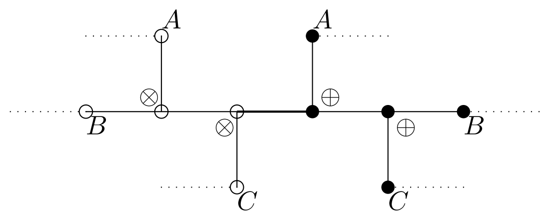
- nets containing structural cut links: explicit resource management
- the elimination of structural cuts: implicit resource management (as in the 'sugared' Gentzen presentation with implicit structural rules)

We illustrate with ASS-R. First the structural axiom decomposition clause:



$$\textbf{ass-r} : (a \bullet b) \bullet c \to a \bullet (b \bullet c)$$

Next the structural cut link:



$$\frac{f : D \to (A \bullet B) \bullet C \quad g : A \bullet (B \bullet C) \to E}{\textbf{ass-r}(f, g) : D \to E}$$

Elimination of the structural cut means replacing the link between the output vertex labeled $(A \bullet B) \bullet C$ and the input vertex $A \bullet (B \bullet C)$ by links on the matching A, B and C components, thus removing the two $\otimes, \oplus$ pairs.

We leave the systematic exploration of this graph rewriting perspective on structural inferences as a subject for further research and close this section with an exercise:

- draw the (pure residuation) net for $(a/b \bullet b/c) \bullet c \rightarrow (a/b \bullet b/c) \bullet c$, call this f;
- draw the (pure residuation) net for $a/b \bullet (b/c \bullet c) \rightarrow a$, call this g;
- connect them by the right-associativity structural link to obtain the net $\mathbf{ass\text{-}r}(f, g) : (a/b \bullet b/c) \bullet c \rightarrow a$
- shift antecedent $\oplus$ pieces to obtain the Geach laws $a/b \rightarrow (a/c)/(b/c)$, $b/c \rightarrow (a/b)\backslash(a/c)$;
- eliminate the structural cut

References

[1] Bellin, G. and A. Fleury (1996) 'Braided proof-nets for multiplicative linear logic with Mix'. Ms, Université Paris VII.

[2] Došen, K. (1992) 'A brief survey of frames for the Lambek calculus'. Zeitschr. f. math. Logik und Grundlagen d. Mathematik **38**, 179–187.

[3] Girard, J.-Y. (1987) 'Linear logic'. *Theoretical Computer Science* **50**, 1–102.

[4] Kurtonina, N. (1995) *Frames and Labels. A Modal Analysis of Categorial Inference.* Ph.D. Dissertation, OTS Utrecht, ILLC Amsterdam.

[5] Kurtonina, N. and M. Moortgat (1997) 'Structural control'. In P. Blackburn and M. de Rijke (eds.) *Specifying Syntactic Structures.* CSLI, Stanford, 75–113.

[6] Lamarche, F. and Ch. Retoré (1996) 'Proof nets for the Lambek calculus — an overview'. In M. Abrusci and C. Casadio (eds.) *Proofs and Linguistic Categories.* Proceedings of the 1996 Roma Workshop. CLUEB Bologna, 241–262.

[7] Lambek, J. (1958) 'The Mathematics of Sentence Structure', *American Mathematical Monthly* **65**, 154–170.

[8] Lambek, J. (1961) 'On the calculus of syntactic types'. In Jakobson (ed.) *Structure of language and its mathematical aspects.* Providence, RI.

[9] Moortgat, M. (1996) 'Multimodal linguistic inference'. *Journal of Logic, Language and Information,* **5**(3,4)(1996), 349–385. Special issue on proof theory and natural language. Guest editors: D. Gabbay and R. Kempson.

[10] Moortgat, M. (1997) 'Categorial Type Logics'. Chapter 2 in J. van Benthem and A. ter Meulen (eds.) *Handbook of Logic and Language.* Elsevier, Amsterdam and MIT Press, Cambridge MA, 1997, 93-177.

[11] Roorda, D. (1991) *Resource Logics. Proof-Theoretical Investigations.* Ph.D. Dissertation, Amsterdam.