Chapter 10

Proof Nets for Multimodal Categorial Grammars

RICHARD MOOT AND QUINTIJN PUITE

ABSTRACT. We present a novel way of using proof nets for multimodal categorial grammars, which provides a general treatment of both the unary and binary connectives. We also introduce a correctness criterion which is valid for a large class of structural rules and sketch basic soundness and completeness results.

10.1 Introduction

Since the introduction of proof nets as an elegant proof theory for the multiplicative fragment of linear logic in [Girard 87] a number of attempts have been made to adapt this proof theory to a variety of categorial grammars, as shown by work from e.g. [Roorda 91], [Morrill 96] and [Moortgat 97].

In this article we will present a new way of looking at proof nets for multimodal categorial grammars. We will show how we can uniformly handle both the unary and binary connectives and how we have a natural correctness criterion for the base logic $\mathbf{NL}\Diamond$ together with a set $\mathcal{R}$ of structural rules subject to some minimal condition.

After presenting the sequent calculus, we will introduce proof structures for our calculus. Then we will look at slightly more abstract graphs, called hypothesis structures, on which we will formulate a correctness criterion in the form of graph contractions. Proof nets will be those proof structures of which the hypothesis structure converts to a tree. We will briefly sketch how to prove our calculus is sound and complete with respect to the sequent calculus and conclude with a discussion comparing our proof net calculus to other approaches and addressing some open questions.

10.2 Sequent Calculus

The sequent calculus for our logic, as given in [Moortgat 97], is the following.

Proceedings of Formal Grammar 1999
Geert-Jan M. Kruijff & Richard T. Oehrle (editors).

$$\frac{}{A \vdash A}\ \textsc{Ax} \qquad \frac{\Delta[B] \vdash C \quad \Gamma \vdash B}{\Delta[\Gamma] \vdash C}\ \textsc{Cut}$$

$$\frac{\Delta[\langle A\rangle^j] \vdash C}{\Delta[\Diamond_j A] \vdash C}\ \mathrm{L}\Diamond_j \qquad \frac{\Delta \vdash C}{\langle\Delta\rangle^j \vdash \Diamond_j C}\ \mathrm{R}\Diamond_j$$

$$\frac{\Delta[A] \vdash C}{\Delta[\langle \Box^{\downarrow}_j A\rangle^j] \vdash C}\ \mathrm{L}\Box^{\downarrow}_j \qquad \frac{\langle\Delta\rangle^j \vdash C}{\Delta \vdash \Box^{\downarrow}_j C}\ \mathrm{R}\Box^{\downarrow}_j$$

$$\frac{\Delta[(A \circ_i B)] \vdash C}{\Delta[A \bullet_i B] \vdash C}\ \mathrm{L}\bullet_i \qquad \frac{\Delta \vdash A \quad \Gamma \vdash B}{(\Delta \circ_i \Gamma) \vdash A \bullet_i B}\ \mathrm{R}\bullet_i$$

$$\frac{\Gamma \vdash B \quad \Delta[A] \vdash C}{\Delta[(A/_i B \circ_i \Gamma)] \vdash C}\ \mathrm{L}/_i \qquad \frac{(\Delta \circ_i B) \vdash A}{\Delta \vdash A/_i B}\ \mathrm{R}/_i$$

$$\frac{\Gamma \vdash B \quad \Delta[A] \vdash C}{\Delta[(\Gamma \circ_i B\backslash_i A)] \vdash C}\ \mathrm{L}\backslash_i \qquad \frac{(B \circ_i \Delta) \vdash A}{\Delta \vdash B\backslash_i A}\ \mathrm{R}\backslash_i$$

In addition to the rules above, the calculus is allowed to have any number of structural rules, schematically of the form

$$\frac{\Delta[\Xi'[\Gamma_1, \ldots, \Gamma_n]] \vdash C}{\Delta[\Xi[\Gamma_{\pi_1}, \ldots, \Gamma_{\pi_n}]] \vdash C}\ [\mathrm{SR}]$$

where Ξ and Ξ' are trees built from the structural operators $(- \circ_i -)$ and $\langle - \rangle^j$ with n structural variables as leaves and where π is a permutation of these leaves. As a consequence, each structural variable occurs once in the premiss and once in the conclusion of a structural rule.

This restriction guarantees the structural rules of contraction and weakening will never be derivable in our logic and as a consequence all our connectives are multiplicatives in the sense of [DR 89].

Illustration: wh-extraction in English

To give an indication of how we can use the calculus described above to give an account of linguistic phenomena, we will look at what is often called *wh*-extraction.

To keep the current discussion simple we will only look at the *wh* word 'whom', which we analyse as a noun modifier selecting a sentence from which a noun phrase is missing. The restriction 'whom' imposes is that this missing noun phrase cannot occur in subject position, as indicated by the following examples.

(10.1) $*$ agent whom [[ ]$_{np}$ interrogated Neo]$_s$

(10.2) agent whom [Trinity escaped [ ]$_{np}$]$_s$

(10.3) agent whom [Morpheus considered [ ]$_{np}$ dangerous]$_s$

To model this behaviour, we give a very simple grammar fragment with only one binary and one unary mode. An extracted np is marked as $\Diamond_0\Box_0^{\downarrow}np$. As $\Diamond_j\Box_j^{\downarrow}A \vdash A$ is a theorem of the base logic for all j and A, this allows these constituents to function as an np. What is crucial is that the $\mathrm{L}\Box_j^{\downarrow}$ rule, read from premiss to conclusion, introduces unary brackets, which makes the following structural rules available for $\langle - \rangle^0$.

$$\frac{\Gamma[\Delta_1 \circ_0 (\Delta_2 \circ_0 \langle \Delta_3 \rangle^0)] \vdash C}{\Gamma[(\Delta_1 \circ_0 \Delta_2) \circ_0 \langle \Delta_3 \rangle^0] \vdash C}\ [\text{P1}]$$

$$\frac{\Gamma[(\Delta_1 \circ_0 \langle \Delta_3 \rangle^0) \circ_0 \Delta_2] \vdash C}{\Gamma[(\Delta_1 \circ_0 \Delta_2) \circ_0 \langle \Delta_3 \rangle^0] \vdash C}\ [\text{P2}]$$

It should be noted, however, that these rules allow a $\langle \Delta \rangle^0$ constituent to move only from a right branch of a structure to another right branch. As a subject would appear on a left branch, this prevents subject extraction as desired.

The lexicon, with which we can derive the well-formed expressions given above is the following:

lex(*agent*)	$= n$
lex(*dangerous*)	$= n/_0 n$
lex(*Neo*)	$= np$
lex(*Trinity*)	$= np$
lex(*Morpheus*)	$= np$
lex(*escaped*)	$= (np\backslash_0 s)/_0 np$
lex(*interrogated*)	$= (np\backslash_0 s)/_0 np$
lex(*considered*)	$= ((np\backslash_0 s)/_0 (n/_0 n))/_0 np$
lex(*whom*)	$= (n\backslash_0 n)/_0 (s/_0 \Diamond_0 \Box_0^{\downarrow} np)$

1 EXAMPLE Given the above lexicon and structural rules, we can give a derivation of expression 10.2 above, 'agent which Trinity escaped', as follows.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{np \vdash np}\,\text{Ax}}{np \vdash np}\,\text{Ax} \quad \dfrac{\dfrac{}{np \vdash np}\,\text{Ax} \quad \dfrac{}{s \vdash s}\,\text{Ax}}{np \circ_0 np\backslash_0 s \vdash s}\,\mathrm{L}\backslash_0}{np \circ_0 ((np\backslash_0 s)/_0 np \circ_0 np) \vdash s}\,\mathrm{L}/_0}{np \circ_0 ((np\backslash_0 s)/_0 np \circ_0 \langle \Box_0^{\downarrow} np \rangle^0) \vdash s}\,\mathrm{L}\Box_0^{\downarrow}}{(np \circ_0 (np\backslash_0 s)/_0 np) \circ_0 \langle \Box_0^{\downarrow} np \rangle^0 \vdash s}\,[\text{P1}]}{(np \circ_0 (np\backslash_0 s)/_0 np) \circ_0 \Diamond_0 \Box_0^{\downarrow} np \vdash s}\,\mathrm{L}\Diamond_0}{np \circ_0 (np\backslash_0 s)/_0 np \vdash s/_0 \Diamond_0 \Box_0^{\downarrow} np}\,\mathrm{R}/_0 \quad \dfrac{\dfrac{}{n \vdash n}\,\text{Ax} \quad \dfrac{}{n \vdash n}\,\text{Ax}}{n \circ_0 n\backslash_0 n \vdash n}\,\mathrm{L}\backslash_0}{n \circ_0 ((n\backslash_0 n)/_0 (s/_0 \Diamond_0 \Box_0^{\downarrow} np) \circ_0 (np \circ_0 (np\backslash_0 s)/_0 np)) \vdash n}\,\mathrm{L}/_0$$

10.3 Proof Structures

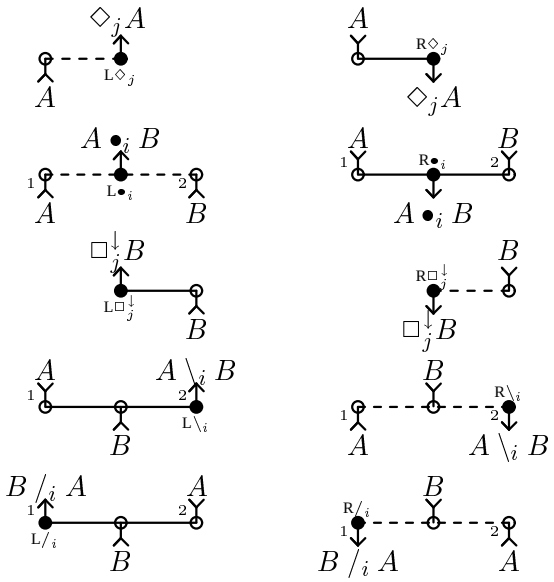
Readers familiar with proof structures as presented in e.g. [Girard 87] will note that we follow [Puite 98] in allowing a proof structure to have hypotheses in addition to conclusions. Doing so means we will have two (symmetric) links for each connective; one for when it occurs as a premiss and one for when it occurs as a conclusion.

1 DEFINITION A *link* is a sequence of premisses and a sequence of conclusions, where one of either the premisses or conclusions is the *output* or *main* formula and the other premisses and conclusions are the *input* or *active* formulas. When the output formula is a premiss we will call the link a *left* link. When the output formula is a conclusion we will call the link a *right* link.

Links are displayed as shown below, with the premisses above the link and the conclusions below, both numbered according to their linear order (when no confusion is possible we will often omit the numbering and order the premisses and conclusions from left to right).

We also need to distinguish between *tensor links* and *par links*. Conform tradition, we represent this graphically by drawing a solid horizontal line for a tensor link and a dashed horizontal line for a par link.

2 DEFINITION A *proof structure* $\langle S, \mathcal{L} \rangle$ consists of a finite set S of formulas together with a set $\mathcal{L}$ of links in S of the following forms, for each binary mode i and unary mode j:

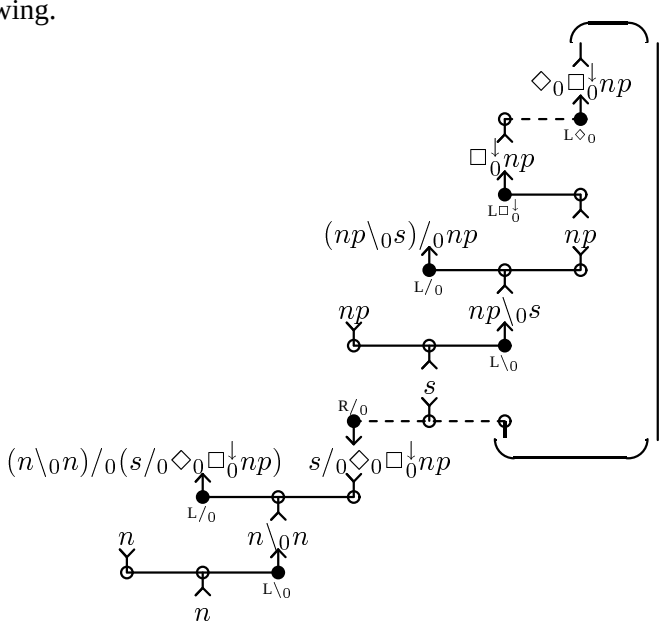


such that the following holds:
- every formula of S is at most once a conclusion of a link.
- every formula of S is at most once a premiss of a link.

Formulas which are not a conclusion of some link are the *hypotheses* H of the proof structure, while those that are not a premiss of some link are its *conclusions* Q.

Note that there are no links corresponding to the axiom or cut rule in the sequent calculus. Instead we will have axiomatic and cut *formulas.* An axiomatic formula is a formula which is not the main formula of any link, whereas a cut formula is a formula which is the main formula of two links.

2 EXAMPLE The proof structure corresponding to the sequent proof of example 1 is the following.



There are five axiomatic formulas in this proof structure: both n formulas, both np formulas and the s formula, each corresponding to one instance of the axiom rule in the sequent proof.

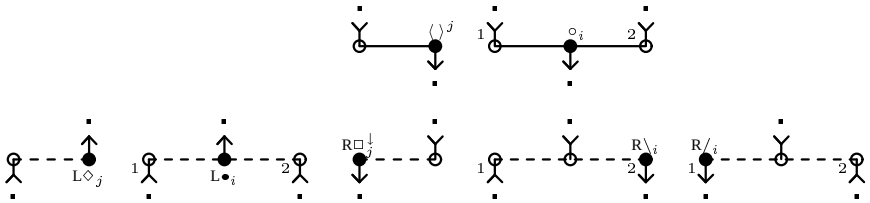
Note that the antecedent formulas of the end-sequent correspond to the hypotheses of the proof structure and the succedent formula to its conclusion, and that every logical link in the proof structure corresponds to a logical rule of the same name in the sequent proof.

We can show by simple induction on the length of the sequent proof that it is always possible to construct a proof structure having the properties mentioned in the example above for a given sequent proof. However, the converse does not hold: there are proof structures which do not correspond to any derivable sequent. A correctness criterion allows us to distinguish proof structures which *do* correspond to derivable sequents, *proof nets,* from other proof structures.

10.4 Hypothesis Structures

To formulate our correctness criterion we need slightly more abstract graphs than proof structures. We will call these structures *hypothesis structures*.

3 DEFINITION A *hypothesis structure* $\langle V, \mathcal{L} \rangle$ consists of a finite set V of vertices, where each vertex is assigned a set of premisses and a set of conclusions, together with a set $\mathcal{L}$ of links in V of the following forms:



such that the following holds:

- every vertex of V is at most once a conclusion of a link.
- every vertex of V is at most once a premiss of a link.

Furthermore, we assign to each vertex a set of premisses and a set of conclusions, where we require that each vertex which is a hypothesis (resp. conclusion) of the structure has a single formula in its set of premisses (resp. conclusions) and for all other vertices this set is empty.

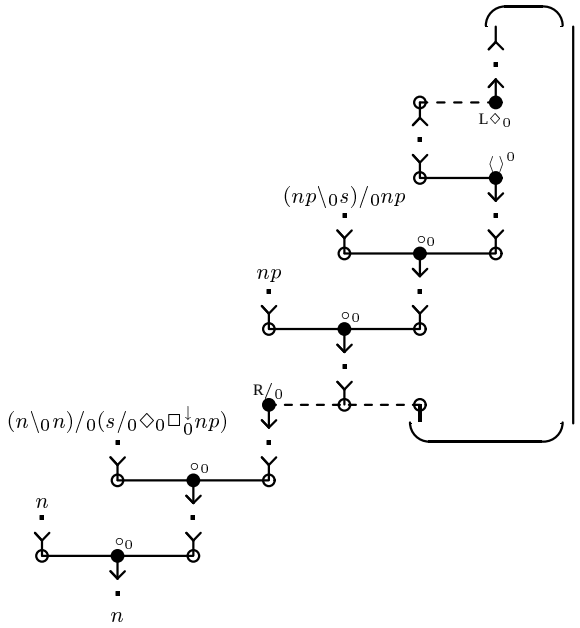
Graphically, we will display premisses above and conclusions below their vertex.

From a proof structure we can construct a unique hypothesis structure by replacing $[\mathrm{L}/_i]$, $[\mathrm{L}\backslash_i]$ and $[\mathrm{R}\bullet_i]$ links by $[\circ_i]$ links, replacing $[\mathrm{L}\Box^{\downarrow}_j]$ and $[\mathrm{R}\Diamond_j]$ link by $[\langle\,\rangle^j]$ links and erasing all formulas except those which are either a premiss or a conclusion of the proof structure.

4 DEFINITION A *hypothesis tree* is an acyclic, connected hypothesis structure containing only $[\circ_i]$ and $[\langle\,\rangle^j]$ links.

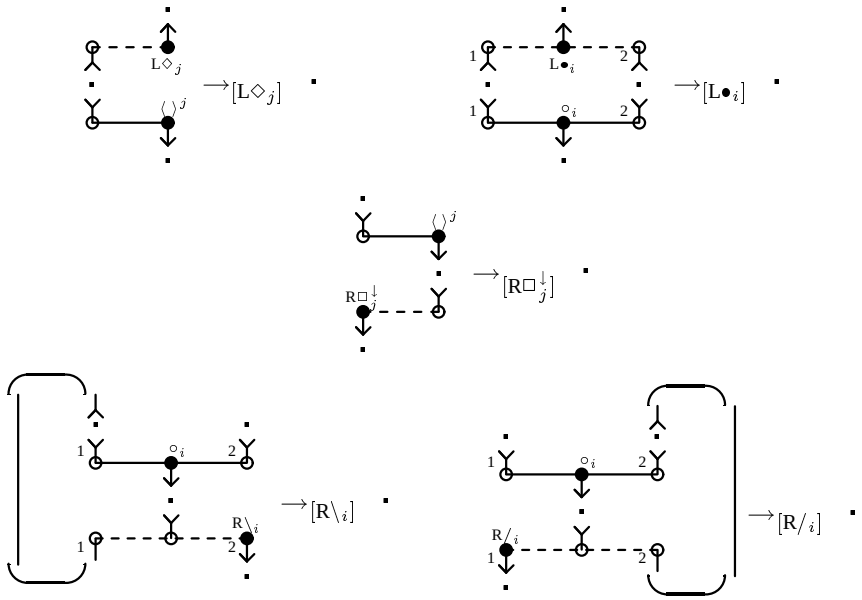
A hypothesis tree $\mathcal{H}$ with premisses $A_1, \ldots, A_n$ and conclusion C corresponds to a sequent with antecedent formulas $A_1, \ldots, A_n$ and succedent formula C in the obvious way.

3 EXAMPLE The hypothesis structure corresponding to the proof structure of example 2 is the following



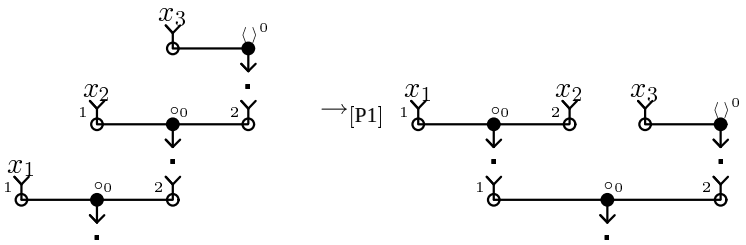
The hypothesis structure of the example above is not a hypothesis tree. This does not mean it is not a *proof net*, of course. We will define a number of conversions on hypothesis structures: contractions, which are valid in the base logic, and structural conversions, which correspond to the structural rules in the sequent calculus.

By a *contraction* we will mean the replacement of one of the following pairs of links by a single node. Contractions will be named after the par link. We require that all vertices involved in the contractions are distinct. The contracted vertex has as premisses (resp. conclusions) the union of all premisses (resp. conclusions) of the contracted vertices.



In addition to these contraction steps a grammar fragment can have a set $\mathcal{R}$ of *structural conversions*. These conversions operate on trees of tensor links only, with the condition that both trees in the conversion have the same set of leaves. This is a reflection of the same restriction on structural rules in the sequent calculus.

4 EXAMPLE The following structural conversion corresponds to sequent rule [P1] of our examples.

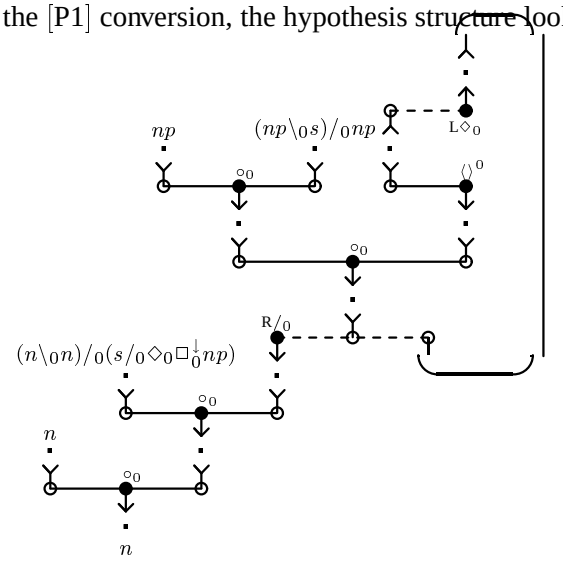


10.5 Proof Nets

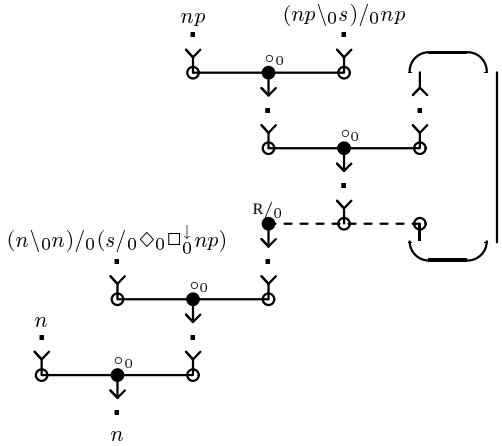
5 DEFINITION A proof structure is a *proof net* if and only if its hypothesis structure converts to a hypothesis tree.

5 EXAMPLE The hypothesis structure of example 3 contracts to a tree given the structural conversion [P1], making it a proof net for any grammar fragment with this structural rule. Without the structural rule, we will not be able to apply the $[\mathrm{R}/_0]$ contraction, which is what we expect given that the base logic **NL**$\diamond$ is non-associative.

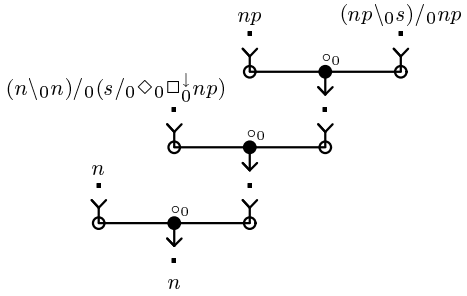
After applying the [P1] conversion, the hypothesis structure looks as follows.



Now we can apply the $[\mathrm{L}\diamond_0]$ contraction, which results in the following hypothesis structure.



Finally, we can apply the $[\mathrm{R}/_0]$ contraction and the result will be the hypothesis tree corresponding to the end-sequent $n \circ_0 ((n\backslash_0 n)/_0(s/_0\Diamond_0\Box_0^{\downarrow}np) \circ_0 (np \circ_0 (np\backslash_0 s)/_0 np)) \vdash n$ of example 1.



Note that we have *computed* the structure of the antecedent instead of assuming it as given and that we have performed the conversions in the exact same order as the corresponding rules in the sequent proof when read from premisses to conclusions.

1 THEOREM A sequent $\Gamma \vdash C$ is derivable in **NL**$\Diamond_{\mathcal{R}}$ if and only if there is a proof structure which converts to the hypothesis tree of $\Gamma \vdash C$, using only the contractions and the structural conversions in $\mathcal{R}$.

We will briefly sketch the proof.

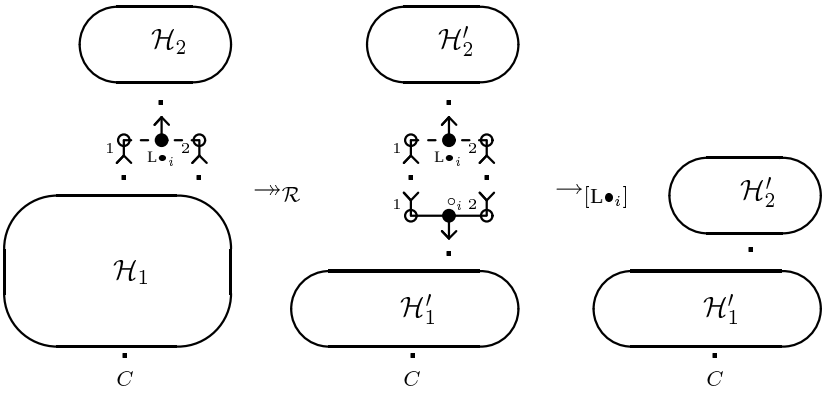
$[\Rightarrow]$ From sequents to proof nets, we proceed by induction on the length of the sequent proof. We extend the conversion sequence with a contraction or a structural conversion every whenever we encounter the corresponding rule in the sequent proof.

$[\Leftarrow]$ The sequentialisation part of the proof proceeds in a way analogous to the 'splitting par' sequentialisation proof of [Danos 90]. We proceed by induction on the number n of conversions in the conversion sequence.

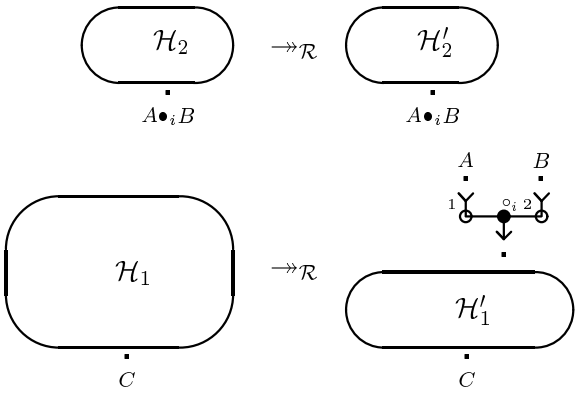
$[n = 0]$ If there are no conversions in the sequence, our hypothesis structure is already a hypothesis tree, which we can decompose by induction on the number of tensor links in it.

$[n > 0]$ We look at the last conversion in the sequence. If it is a structural conversion, the induction hypothesis gives us a sequent derivation which we can extend by applying the appropriate structural rule.

If the last conversion is a $[\mathrm{L}\bullet_i]$ contraction we are schematically in the following situation



Reasoning backwards from the hypothesis tree, we can see that the $[\mathrm{L}\bullet_i]$ link serves as a boundary and that every conversion in the sequence is either applied strictly above or strictly below it. So we can split our initial conversion sequence $\mathcal{R}$ into a conversion sequence which converts $\mathcal{H}_1$ to a hypothesis tree and a conversion sequence which converts $\mathcal{H}_2$ into a hypothesis tree as follows



Now let Γ_1 be the antecedent term corresponding to $\mathcal{H}'_1$ and Γ_2 the antecedent term corresponding to $\mathcal{H}'_2$. As both reduction sequences are strictly smaller than our initial reduction sequence, the induction hypothesis gives us a derivation $\mathcal{D}_1$ ending in $\Gamma_2 \vdash A \bullet_i B$ and a deriva-

tion $\mathcal{D}_2$ ending in $\Gamma_1[(A \circ_i B)] \vdash C$. We can combine these derivations as shown below

$$\dfrac{\Gamma_2 \vdash A \bullet_i B \quad \begin{array}{c}\vdots\ \mathcal{D}_2\\ \dfrac{\Gamma_1[(A \circ_i B)] \vdash C}{\Gamma_1[A \bullet_i B] \vdash C}\ \mathrm{L}\bullet_i\end{array}}{\Gamma_1[\Gamma_2] \vdash C}\ \textsc{Cut}$$

The other contractions are similar. □

10.6 Discussion

We have presented a proof net calculus for multimodal categorial grammars which is new, elegant and very general.

The formalism we have presented here is related to a number of other proposals, notably to Danos' graph contractions [Danos 90], of which our contractions are a special case. As a result, acyclicity and connectedness of the underlying correction graphs are a consequence of our correctness criterion.

Our approach is also related to the labeled proof nets of [Moortgat 97]. Our hypothesis structures correspond closely to the labels Moortgat assigns to proof nets. Advantages of our formalism are that we have a very direct correspondence between proof structures and their hypothesis structures and that we can handle cyclic or disconnected proof structures unproblematically, whereas only acyclic and connected proof structures can be assigned a meaningful label.

Some interesting questions remain open. It seems possible, for instance, to formulate a classical version of our proof net calculus, though it is unclear to us at the moment what kind of sequent calculus a classical categorial grammar would have. A possible solution is suggested by current research from [GL 99], who propose a sequent and proof net calculus for classical non-associative linear logic.

We would also like to have some natural notion of equality on conversion sequences. As defined now, a number of uninteresting permutations are possible in the conversion sequence, which is somewhat against the spirit of proof nets. Computationally, we could exploit such a notion to prune the search space while maintaining equivalence.

Bibliography

[Danos 90] Danos, V. *La Logique Linéaire Appliquée à l'étude de Divers Processus de Normalisation (Principalement du Lambda-Calcul).* Thèse de Doctorat, Université de Paris VII, 1990.

[DR 89] Danos, V. and L. Regnier *The Structure of Multiplicatives.* Archive for Mathematical Logic **28**, 1989, pp. 181-203.

[Girard 87] Girard, J.Y. *Linear Logic.* Theoretical Computer Science **50**, 1987, pp. 1-102.

[GL 99] Groote, P. de, and F. Lamarche *Non-associative Linear Logic*, Manuscript, 1999.

[Moortgat 97] Moortgat, M. *Categorial Type Logics.* Chapter 2 of Benthem, J. van, and A. ter Meulen (eds.) *Handbook of Logic and Language.* Elsevier, 1997.

[Morrill 96] Morrill, G. *Memoisation of Categorial Proof Nets: Parallelism in Categorial Processing.* Report de Recerca LSI-96-24-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 1996.

[Puite 98] Puite, Q. *Proof Nets with Explicit Negation for Multiplicative Linear Logic.* Preprint 1079, Department of Mathematics, Utrecht University, 1998.

[Roorda 91] Roorda, D. *Resource Logics: A Proof-theoretical Study.* PhD Thesis, University of Amsterdam, 1991.