

# CHAPTER 7

---

## CONTRACTION CRITERIA

IN the previous chapter, we have seen how we could use algebraic correctness criteria for  $\text{NL}\diamond_{\mathcal{R}}$ . In this chapter, I will present a graph theoretic correctness criterion, in the line of Danos' contraction criterion for Multiplicative Linear Logic, discussed in Section 4.5.

This chapter is based on joint research with Quintijn Puite, and parts of it have appeared before in (Puite & Moot 1999), (Moot & Puite 1999) and (Moot & Puite 2001).

### 7.1 Two Sided Proof Nets

Proof nets, as we have seen them in the previous chapters, having their roots in the one sided sequent calculus, have a certain asymmetry in that they are allowed to have conclusions, i.e. formulas which are not the premiss of a link, but not hypotheses, i.e. formulas which are not the conclusion of a link. Puite (1998) proposes a proof net calculus based on the two sided sequent calculus, where a proof net can have both hypotheses and conclusions.

In this calculus every link has a dual link. So, in addition to a normal right tensor link, we have a left tensor link as shown in Table 7.1 on the next page, which is by symmetry a par link and which has  $A \otimes B$  as a premiss and  $A$  and  $B$  as conclusions.

In the two sided calculus, this new link is a primitive link, but we can see it as a *defined* link in the one sided calculus as shown in Figure 7.1 on the following page.

If we compare the  $[L\otimes]$  link with the defined link above, it is easy to verify that for any proof structure  $\mathcal{S}$  with the primitive  $[L\otimes]$  link we can generate a proof structure  $\mathcal{S}'$  where the primitive link is replaced by the defined link

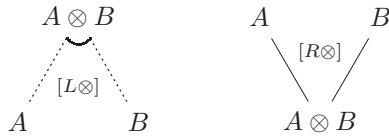


Table 7.1: Two sided tensor links

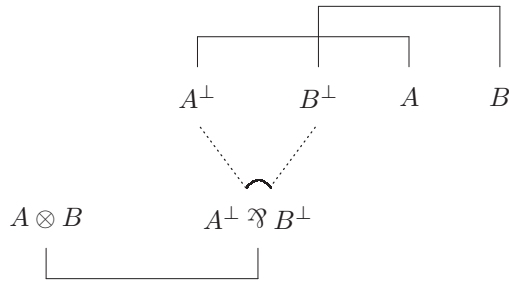


Figure 7.1: One sided representation of the  $[L \otimes]$  link

above. Now for every switching  $\omega$  the correction graph  $\omega S$  will be acyclic and connected if and only if  $\omega S'$  is acyclic and connected.

We also have a left and right par link, which look as shown in Table 7.2.

One of the advantages of a two sided calculus is that we can now have explicit negation links as shown in Table 7.3 on the facing page, a left negation link, which looks very much like an axiom link, and a right negation link, which looks very much like a cut link.

The elimination of double negation and the De Morgan equalities, which were ‘compiled away’ in the one sided calculus, now become explicitly representable in the two sided calculus.

**Example 7.1** *As an example, we construct the proof net for the De Morgan equivalence below.*

$$a \otimes b \vdash (a^\perp \wp b^\perp)^\perp$$

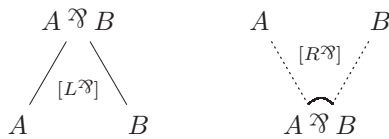


Table 7.2: Two sided par links

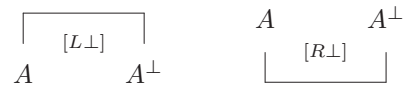


Table 7.3: Negation links

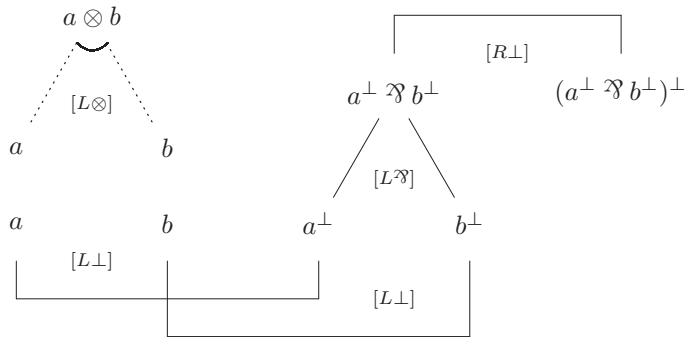


Figure 7.2: Example formula unfolding

We first unfold the formulas as before, which produces the result shown in Figure 7.2.

We can see from the example above that axiom links in the one sided proof net calculus are, from the two sided point of view, a combination of a left negation link and the identification of two syntactically identical formulas. Therefore, in the two sided calculus the axiom rule corresponds only to the unification of two formulas.

For the proof structure of Figure 7.2 there is a unique way to produce a proof net, which is shown as Figure 7.3.

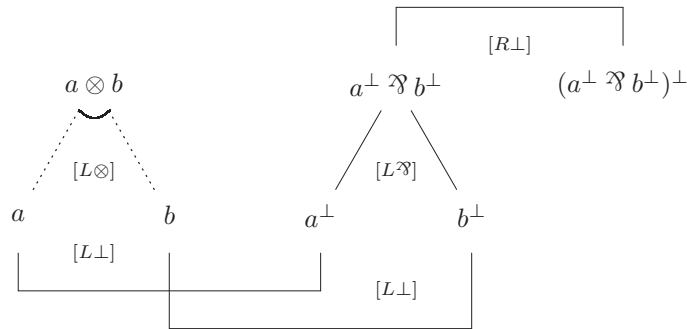


Figure 7.3: Proof net for the formula unfolding of Figure 7.2

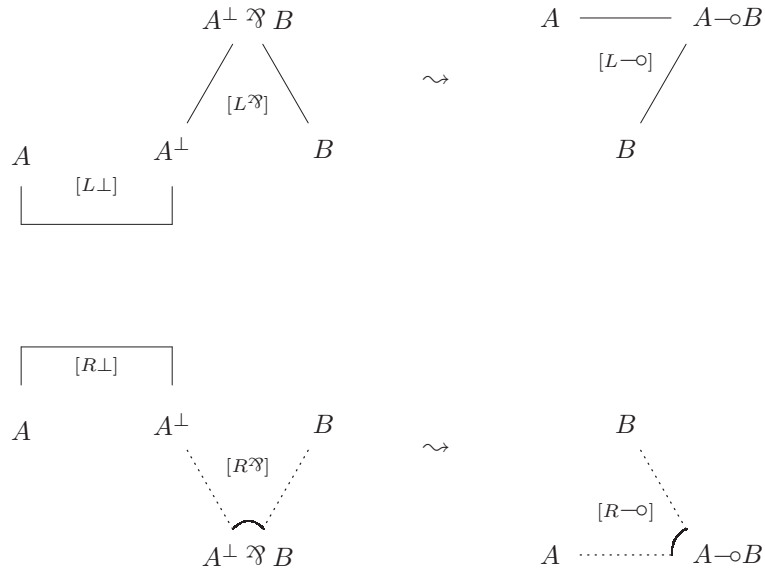


Table 7.4: Links for implication, defined and primitive

Now, given that linear implication  $A-\odot B$  is defined as  $A^\perp \wp B$ , we can abbreviate a combined par and negation link as shown in Table 7.4.

**Example 7.2** *The sequent below*

$$a-\odot b, b-\odot c \vdash a-\odot c$$

generates the proof structure shown in Figure 7.4 on the next page after connecting the  $b$  and  $c$  atomic formulas.

Note that we now want to connect the  $a$  atomic formulas, but it is difficult to portray this on a flat plane. When we imagine the plane we draw on is cylindrical, i.e. if we move up far enough we reenter the plane from below, we can see how moving upward from the top  $a$  formula we would reach the bottom  $a$  formula. However, we will choose to portray these situations by using curved connections, as shown in Figure 7.5 on the facing page.

Observe that the above representation is very close to the corresponding natural deduction proof, shown in Figure 7.6 on the next page, with the difference that instead of coindexing the  $[R-\odot]$  rule with the discharged hypothesis  $a$ , we represent the discharge by a link which removes the hypothesis  $a$  from the hypotheses of the proof structure.

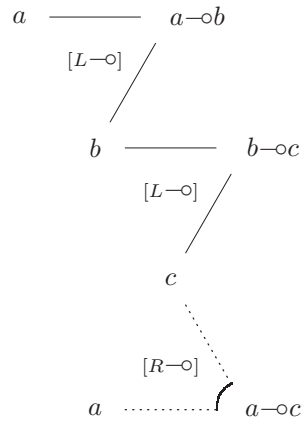


Figure 7.4: Proof structure for  $a, a \multimap b, b \multimap c \vdash a, a \multimap c$

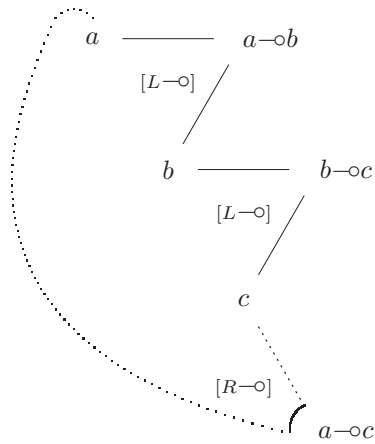


Figure 7.5: Proof net for  $a \multimap b, b \multimap c \vdash a \multimap c$

## 7.2 Sequent Calculus

Though we have seen in the previous section how to extend proof nets for multiplicative linear logic to a two sided calculus, unless we modify our cor-

$$\frac{\frac{[a \vdash a]^1 \quad \frac{b \vdash b \quad c \vdash c}{b, b \multimap c \vdash c} [L \multimap]}{a, a \multimap b, b \multimap c \vdash c} [L \multimap]}{a \multimap b, b \multimap c \vdash a \multimap c} [R \multimap]^1$$

Figure 7.6: Natural deduction proof of  $a \multimap b, b \multimap c \vdash a \multimap c$

rectness criterion, we will still be stuck in the multiplicative fragment of linear logic, albeit in the two sided formulation of it.

The rest of this chapter will be devoted to giving a correctness criterion for  $\mathbf{NL}\diamond_{\mathcal{R}}$  by stating a contraction criterion which is a special case of Danos' contraction criterion as discussed in Section 4.5. Moreover, this criterion will be modular in that it generates a correctness criterion from the structural rule component  $\mathcal{R}$  of specific grammars, as long as they are linear according to Definition 3.4. In Section 7.8, we will give a specialized contraction criterion for the original Lambek calculus  $\mathbf{L}$ .

For ease of reference, we repeat the language and the sequent calculus of  $\mathbf{NL}\diamond_{\mathcal{R}}$  here.

**Definition 7.3** *The language  $\mathcal{L}(\mathbf{NL}\diamond_{\mathcal{R}})$  consists of the following.*

[Formulas] *Given a set of atomic formulas  $\mathcal{A}$ , a set  $I$  of binary indices and a set  $J$  of unary indices, the set of formulas is defined inductively as follows*

$$\mathcal{F} ::= \mathcal{A} \mid \diamond_j \mathcal{F} \mid \square_j^\perp \mathcal{F} \mid \mathcal{F} /_i \mathcal{F} \mid \mathcal{F} \bullet_i \mathcal{F} \mid \mathcal{F} \setminus_i \mathcal{F}$$

*We will use  $A, B, C, \dots$  to denote arbitrary formulas.*

[Antecedent Terms] *Over the set of formulas  $\mathcal{F}$ , we define the set of antecedent terms  $\mathcal{T}$  as follows*

$$\mathcal{T} ::= \mathcal{F} \mid \langle \mathcal{T} \rangle^j \mid \mathcal{T} \circ_i \mathcal{T}$$

*We will use  $\Gamma, \Delta, \Xi, \dots$  to denote arbitrary antecedent terms. When we want to refer to a specific subtree occurrence  $\Delta$  of an antecedent term  $\Gamma$  we will write this as  $\Gamma[\Delta]$*

[Sequents] *A sequent is written as  $\Gamma \vdash C$ , where  $\Gamma$  is an antecedent term we will call the antecedent of the sequent and  $C$  is a formula we will call the succedent of the sequent.*

The sequent calculus for  $\mathbf{NL}\diamond$  is given by the rules in Table 7.5 on the facing page. The set of structural rules  $\mathcal{R}$  is an additional set of sequent rules, which is dependent on the application of the system. Each rule of  $\mathcal{R}$  is schematically of the form shown on the last line of Figure 7.5 on the next page, where  $\Xi$  and  $\Xi'$  are fixed trees built from the structural operators  $(- \circ_i -)$  and  $\langle - \rangle^j$  with  $n$  distinct structural variables as leaves, and where  $\pi$  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 Danos & Regnier (1989).

### Illustration: wh extraction in English

As our running example throughout this chapter we will look at what is often called *wh* extraction.

$$\begin{array}{c}
\frac{}{A \vdash A} [Ax] \qquad \frac{\Gamma[A] \vdash C \quad \Delta \vdash A}{\Gamma[\Delta] \vdash C} [Cut] \\
\\
\frac{\Gamma[\langle A \rangle^j] \vdash C}{\Gamma[\diamond_j A] \vdash C} [L\diamond_j] \qquad \frac{\Gamma \vdash C}{\langle \Gamma \rangle^j \vdash \diamond_j C} [R\diamond_j] \\
\\
\frac{\Gamma[A] \vdash C}{\Gamma[\langle \square_j^\perp A \rangle^j] \vdash C} [L\square_j^\perp] \qquad \frac{\langle \Gamma \rangle^j \vdash C}{\Gamma \vdash \square_j^\perp C} [R\square_j^\perp] \\
\\
\frac{\Gamma[(A \circ_i B)] \vdash C}{\Gamma[A \bullet_i B] \vdash C} [L\bullet_i] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{(\Gamma \circ_i \Delta) \vdash A \bullet_i B} [R\bullet_i] \\
\\
\frac{\Gamma[A] \vdash C \quad \Delta \vdash B}{\Gamma[(A/_i B \circ_i \Delta)] \vdash C} [L/_i] \qquad \frac{(\Gamma \circ_i B) \vdash A}{\Gamma \vdash A/_i B} [R/_i] \\
\\
\frac{\Gamma[A] \vdash C \quad \Delta \vdash B}{\Gamma[(\Delta \circ_i B \setminus_i A)] \vdash C} [L\setminus_i] \qquad \frac{(B \circ_i \Gamma) \vdash A}{\Gamma \vdash B \setminus_i A} [R\setminus_i] \\
\\
\frac{\Gamma[\exists'[\Delta_1, \dots, \Delta_n]] \vdash C}{\Gamma[\exists[\Delta_{\pi_1}, \dots, \Delta_{\pi_n}]] \vdash C} [SR]
\end{array}$$

Table 7.5: The sequent calculus  $\mathbf{NL}\diamond_{\mathcal{R}}$ 

To keep the current discussion simple we will only look at the *wh* word ‘whom’, which we analyze 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.

(7.1) \* agent whom [ [ ]<sub>np</sub> interrogated Neo ]<sub>s</sub>

(7.2) agent whom [ Trinity escaped [ ]<sub>np</sub> ]<sub>s</sub>

(7.3) agent whom [ Morpheus considered [ ]<sub>np</sub> dangerous ]<sub>s</sub>

To model this behavior, we give a very simple grammar fragment with only one binary and one unary mode. An extracted *np* is marked as  $\diamond_0 \square_0^\perp np$ . As  $\diamond_j \square_j^\perp A \vdash A$  is a theorem of the base logic for all *j* and *A*, this allows these constituents to function as an *np*. Crucial for this application is that the  $[L\square_j^\perp]$  rule, read top down, introduces unary brackets, which produces the proper configuration for the structural rules shown in Table 7.6.

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.

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

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

Table 7.6: Structural rules for *wh* extraction

$$\frac{\frac{\frac{}{s \vdash s} [Ax]}{np \circ_0 np \backslash_0 s \vdash s} [L \backslash_0] \quad \frac{\frac{}{np \vdash np} [Ax]}{np \vdash np} [L /_0]}{np \circ_0 ((np \backslash_0 s) /_0 np \circ_0 np) \vdash s} [L \square_0^\downarrow]}{\frac{\frac{}{np \circ_0 ((np \backslash_0 s) /_0 np \circ_0 \langle \square_0^\downarrow np \rangle^0) \vdash s} [P1]}{(np \circ_0 (np \backslash_0 s) /_0 np) \circ_0 \langle \square_0^\downarrow np \rangle^0 \vdash s} [L \diamond_0]}{np \circ_0 ((np \backslash_0 s) /_0 np) \circ_0 \diamond_0 \square_0^\downarrow np \vdash s} [R /_0]}{\frac{\frac{\frac{}{n \vdash n} [Ax]}{n \circ_0 n \backslash_0 n \vdash n} [L \backslash_0] \quad \frac{\frac{}{n \vdash n} [Ax]}{n \vdash n} [L /_0]}{np \circ_0 (np \backslash_0 s) /_0 np \vdash s /_0 \diamond_0 \square_0^\downarrow np} [L /_0]}{n \circ_0 ((n \backslash_0 n) /_0 (s /_0 \diamond_0 \square_0^\downarrow np) \circ_0 (np \circ_0 (np \backslash_0 s) /_0 np)) \vdash n}$$

Figure 7.7: Sequent derivation for ‘agent whom Trinity escaped’

The lexicon, with which we can derive the well-formed expressions given above is shown in Table 7.7.

$\text{lex}(\text{agent})$	$= n$
$\text{lex}(\text{dangerous})$	$= n /_0 n$
$\text{lex}(\text{Neo})$	$= np$
$\text{lex}(\text{Trinity})$	$= np$
$\text{lex}(\text{Morpheus})$	$= np$
$\text{lex}(\text{escaped})$	$= (np \backslash_0 s) /_0 np$
$\text{lex}(\text{interrogated})$	$= (np \backslash_0 s) /_0 np$
$\text{lex}(\text{considered})$	$= ((np \backslash_0 s) /_0 (n /_0 n)) /_0 np$
$\text{lex}(\text{whom})$	$= (n \backslash_0 n) /_0 (s /_0 \diamond_0 \square_0^\downarrow np)$

Table 7.7: Example lexicon

**Example 7.4** Given the above structural rules and the lexicon of Figure 7.7, we can give a derivation of expression 7.2, ‘agent whom Trinity escaped’, as shown in Figure 7.7.



## 7.3 Proof Structures

We will now present proof structures for  $\mathbf{NL}\diamond_{\mathcal{R}}$ .

**Definition 7.5** A link  $L$  is a tuple  $\langle \tau, \nu, P, Q, p, q \rangle$ , where  $\tau$  is either ' $\otimes$ ' or ' $\wp$ ',  $\nu$  is a label indicating the name of the link,  $P$  is a sequence of formulas which we call the premisses of  $L$ ,  $Q$  is a sequence of formulas which we call the conclusions of  $L$ ,  $p$  is a subsequence of  $P$  and  $q$  is a subsequence of  $Q$  such that  $\text{length}(p) + \text{length}(q) \leq 1$ .

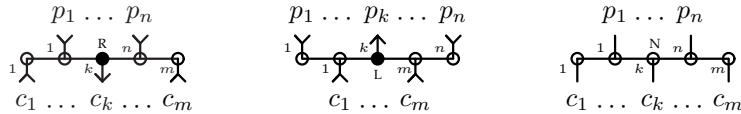
If  $\tau = '\otimes'$ , we will call the link a tensor link. If  $\tau = '\wp'$ , we will call the link a par link.

If  $\text{length}(p) = 1$ , we will call the link a left link, the element  $A$  of  $p$  the output or main formula and the other elements of  $P$  and the elements of  $Q$  the input or active formulas.

If  $\text{length}(q) = 1$ , we will call the link a right link, the element  $B$  of  $q$  the output or main formula and the elements of  $P$  and the other elements of  $Q$  the input or active formulas.

If  $\text{length}(p) = \text{length}(q) = 0$  we will call the link a neutral link.

We will display our links a bit different from the way we did in Section 7.1 and portray them, following Puite (1998), as shown below, with the premisses above the horizontal line and the conclusions below it, 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). For left and right links, we indicate the main formula of the link by an arrow moving *to* that formula, while the active formulas have an arrow moving *from* them. Neutral links are displayed without arrows. We attach the label  $\nu$  to the horizontal line.



We distinguish tensor links and par links graphically by drawing a solid horizontal line for a tensor link and a dashed horizontal line for a par link.

**Definition 7.6** 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 forms shown in Table 7.8, 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 the conclusion of any link are the *hypotheses*  $H$  of the proof structure, while those that are not the premiss of any link are its *conclusions*  $Q$ .

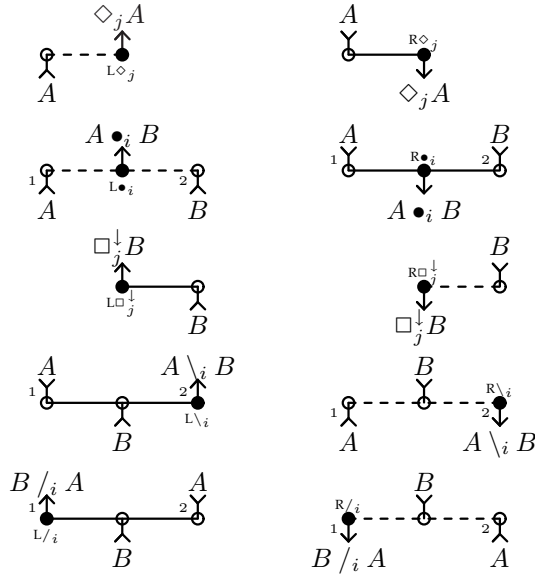


Table 7.8: Links for  $\mathbf{NL}\diamond\mathcal{R}$

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. We will call all formulas which are neither cut nor axiomatic formulas flow formulas.

We can determine which formulas are axiomatic and cut formulas in the graphical representation of a proof structure as follows. An axiomatic formula has no arrows pointing to it; depending on whether it is a hypothesis or a conclusion of the proof structure, there are four possibilities. A cut formula has two arrows pointing to it.

$A$	$\begin{array}{c} A \\ \downarrow \\ \text{Y} \end{array}$	$\begin{array}{c} \downarrow \\ A \end{array}$	$\begin{array}{c} \downarrow \\ A \\ \downarrow \\ \text{Y} \end{array}$	$\begin{array}{c} \downarrow \\ A \\ \uparrow \end{array}$
(Axiom)	(Axiom)	(Axiom)	(Axiom)	(Cut)
hypothesis	hypothesis	-hypothesis	-hypothesis	
conclusion	-conclusion	conclusion	-conclusion	

**Example 7.7** The proof structure corresponding to the sequent proof of Figure 7.7 is shown in Figure 7.8 on the facing page.

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.

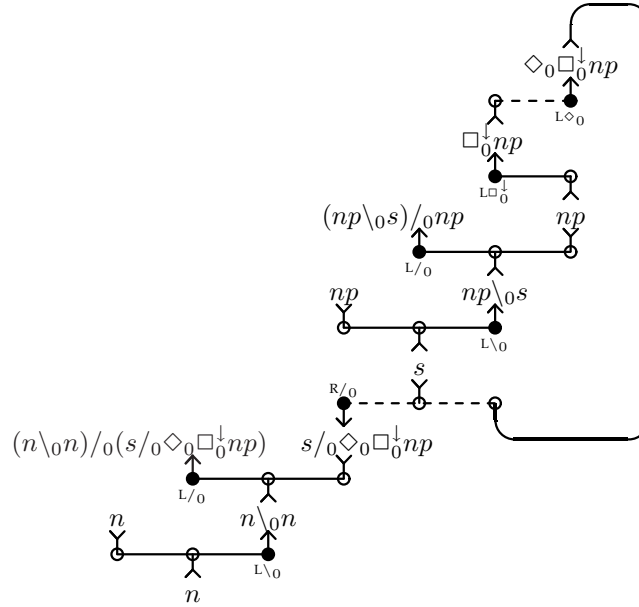


Figure 7.8: Proof structure corresponding to the derivation of Figure 7.7

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 rule of the sequent proof corresponds to a link of the same name in the proof structure.

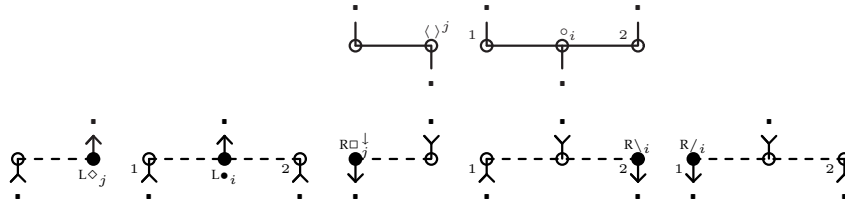
We can show by simple induction on the length of the sequent proof that for any given sequent proof we can construct a proof structure having the properties mentioned in the example above.

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.

## 7.4 Abstract Proof Structures

To formulate our correctness criterion we need to convert our proof structures into slightly more abstract graphs. We will call these graphs *abstract proof structures* (aps's).

**Definition 7.8** An abstract proof structure  $\langle V, \mathcal{L} \rangle$  consists of a finite set  $V$  of vertices, where each vertex is assigned a sequence of premisses and a sequence of conclusions, together with a set  $\mathcal{L}$  of links in  $V$  of the following forms.



such that the following holds.

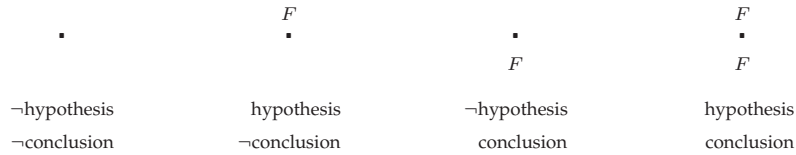
- every vertex of  $V$  is at most once a conclusion of a link,
- every vertex of  $V$  is at most once a premiss of a link.

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

From a proof structure  $S$  we obtain an abstract proof structure  $\mathcal{A}$  by replacing all  $[L/_i]$ ,  $[L\setminus_i]$  and  $[R\bullet_i]$  links by  $[\circ_i]$  links and all  $[L\Box_j^{\perp}]$  and  $[R\Diamond_j]$  links by  $[\langle \rangle^j]$  links. That is, we forget about the inputs and outputs of all tensor links.

In addition, we replace every formula  $F$  of  $S$  by a vertex  $v$ , which is assigned  $[F]$  as its sequence of premisses if  $F$  is a hypothesis of  $S$  and  $[\ ]$  otherwise, and which is assigned  $[F]$  as its sequence of conclusions if  $F$  is a conclusion of  $S$  and  $[\ ]$  otherwise.

Graphically, we will display premisses above and conclusions below their vertex, as shown below. The premisses and conclusion play no active role in our correctness criterion, they merely allow us to keep track of which formula occurrences are assigned to the hypotheses and conclusions of the proof structure.



We will write  $S \mapsto \mathcal{A}$  to indicate that the aps  $\mathcal{A}$  is obtained from the proof structure  $S$  in this fashion. We will often write  $\widehat{S}$  for the abstract proof structure  $\mathcal{A}$  obtained from  $S$ .

**Definition 7.9** A hypothesis tree is an acyclic, connected abstract proof structure containing only  $[\circ_i]$  and  $[\langle \rangle^j]$  links.

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

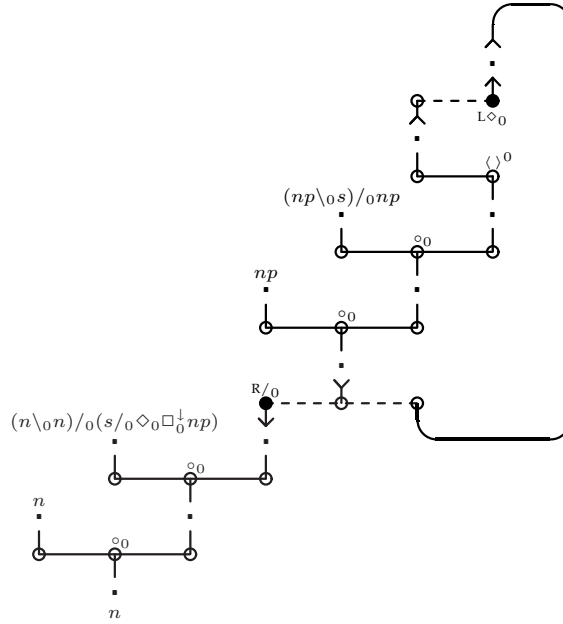


Figure 7.9: Abstract proof structure corresponding to Figure 7.8

We will write  $\Gamma_C$  for the hypothesis tree corresponding to the sequent  $\Gamma \vdash C$ .

**Example 7.10** The abstract proof structure computed for the proof structure of Figure 7.8 from Example 7.7 is shown in Figure 7.9.

The abstract proof structure of the example above is not a hypothesis tree. This does not mean the original proof structure is not a *proof net*, of course. We will define a number of conversions on abstract proof 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 pairs of links shown in Table 7.9 on the following page by a single node. Contractions will be named after the par link. We require that all vertices shown in the redices of Table 7.9 are distinct.  $\sigma_H$  and  $\sigma_Q$  represent the sequence of hypotheses and the sequence of conclusions of the displayed vertex respectively.

Note that all contractions are a variation on the same theme: in every redex the ‘active vertices’ of a par link are connected to a single, neutral tensor link in a way which respects the left to right ordering and the reduct is a single node.

In addition to these contraction steps a grammar fragment can have a set  $\mathcal{R}$  of *structural conversions*. These conversions operate on trees of neutral 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.

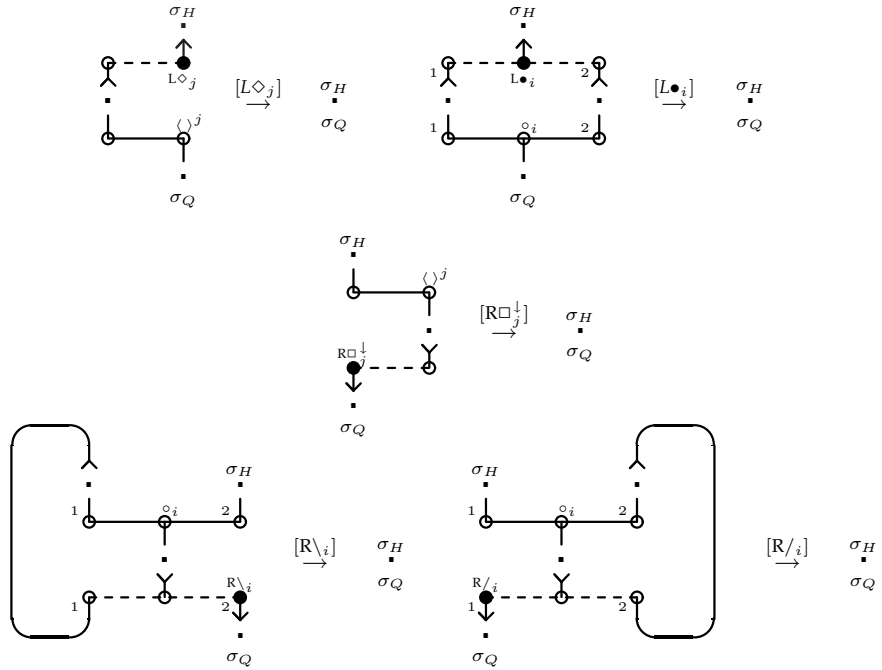
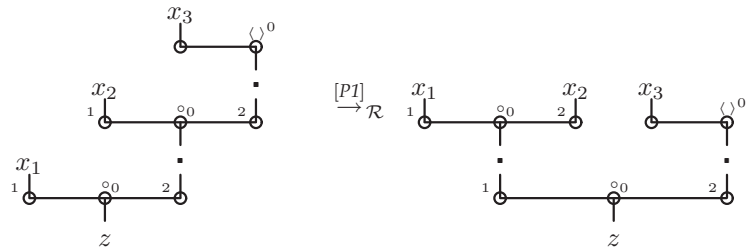


Table 7.9: The contractions for  $NL\Diamond$

**Example 7.11** *The following structural conversion corresponds to sequent rule [P1] of our examples.*

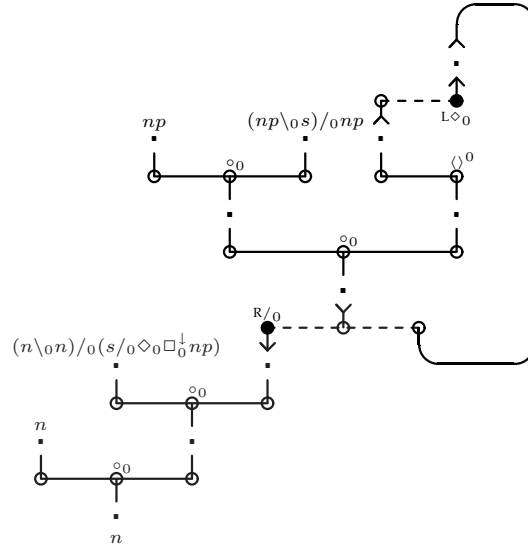


### 7.5 Proof Nets

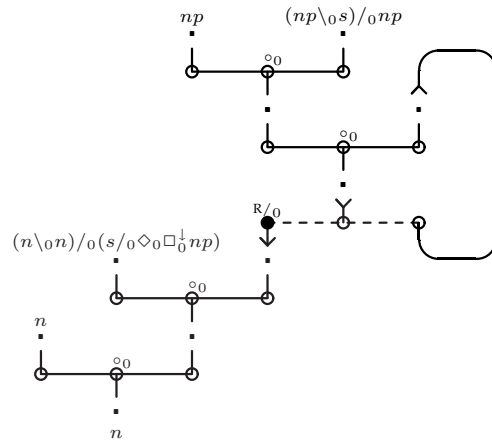
**Definition 7.12** *A proof structure is a proof net if and only if its abstract proof structure converts to a hypothesis tree.*

**Example 7.13** *The abstract proof structure of Example 7.10 converts to a tree given the structural conversion [P1], making this proof structure a proof net for any grammar fragment with this structural rule.*

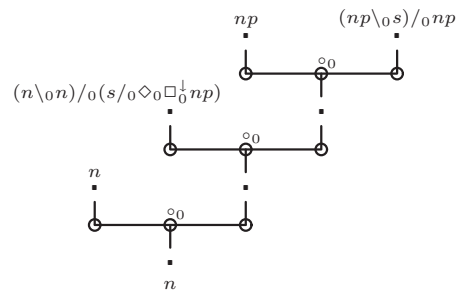
*After applying the [P1] conversion, the abstract proof structure looks as follows.*



Now we can apply the  $[L \diamond_0]$  contraction, which results in the following abstract proof structure.



Finally, we can apply the  $[R /_0]$  contraction and the result will be the following hypothesis tree



which corresponds to the end-sequent

$$n \circ_0 ((n \setminus_0 n) /_0 (s /_0 \diamond_0 \square_0^\perp np)) \circ_0 (np \circ_0 (np \setminus_0 s) /_0 np) \vdash n$$

of Example 7.4.

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 axioms to end-sequent.

Before we prove our main theorem, we will first prove the following short lemma.

**Lemma 7.14** *If  $\mathcal{S}$  is a non-trivial proof structure such that the underlying abstract proof structure  $\widehat{\mathcal{S}}$  is actually a hypothesis tree, then at least one of the leaves (conclusion and hypotheses) of  $\mathcal{S}$  is the main formula of its link.*

**Proof** To prove: if every hypothesis is an active formula of its link, then the conclusion is the main formula of its link. We proceed by induction on the hypothesis tree  $\Gamma$ .

The trivial case  $\Gamma = \cdot^A$  cannot occur.

In case  $\Gamma = \Gamma_1 \circ_i \Gamma_2$ , assume every hypothesis is an active formula of its link. We write  $L$  for the final  $\circ_i$  link, connecting  $\Gamma_1$  and  $\Gamma_2$ . If  $\Gamma_1$  is trivial, the assumption entails that the corresponding formula in  $\mathcal{S}$  is an active formula of  $L$ . If  $\Gamma_1$  is non-trivial, by induction hypothesis we know that its conclusion is the main formula of the link above, whence of the form  $\diamond_j A$  or  $A \bullet_i B$ . This implies that it is not the main formula of  $L$ , which would be  $\square_j^\perp B$ ,  $A \setminus_i B$  or  $B /_i A$ . Hence it is an active formula of  $L$ . The same holds for the second premiss of  $L$ . As both premisses are active, the conclusion of  $L$  must be main, as desired.

The case  $\Gamma = \langle \Gamma_1 \rangle^j$  is proved analogously.  $\square$

**Theorem 7.15** *A sequent  $\Gamma \vdash C$  is derivable in  $\mathbf{NL} \diamond_{\mathcal{R}}$  if and only if there is a proof structure  $\mathcal{S}$  which converts to the hypothesis tree  $\Gamma_C$ , using only the contractions and the structural conversions in  $\mathcal{R}$ .*

**Proof**

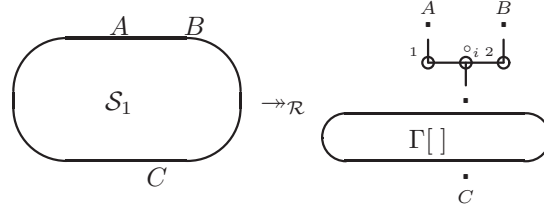
[ $\Rightarrow$ ] From sequents to proof nets, we proceed by induction on the length of the sequent proof. We extend the conversion sequence with a contraction or a structural conversion whenever we encounter the corresponding rule in the sequent proof. By ‘applying a conversion step to a proof structure’ we will mean ‘applying a conversion step to its underlying abstract proof structure’.

For a  $[L \bullet_i]$  rule

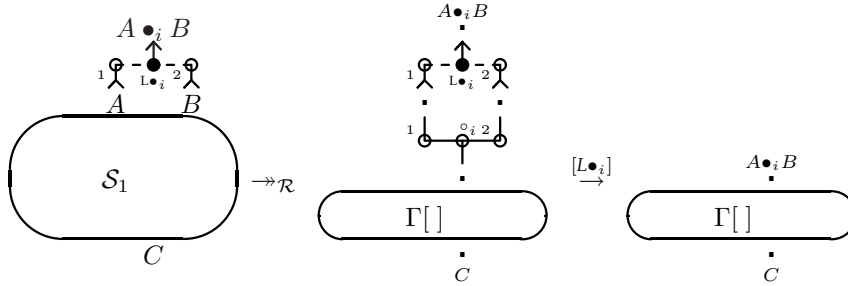
$$\frac{\Gamma[(A \circ_i B)] \vdash C}{\Gamma[A \bullet_i B] \vdash C} [L \bullet_i]$$

we know the following by induction hypothesis.





We can keep the original conversion sequence, where we attach a  $[L\bullet_i]$  link to the abstract proof structure and we extend the conversion sequence with a  $[L\bullet_i]$  contraction as follows.



$[\Leftarrow]$  The sequentialization part of the proof proceeds in a way analogous to the ‘splitting par’ sequentialization proof of Danos (1990). We proceed by induction on the length  $l$  of the conversion sequence.

$[l = 0]$  If there are no conversions in the sequence, our proof structure corresponds to an abstract proof structure which is already a hypothesis tree. We proceed by induction on the number of tensor links in the proof structure.

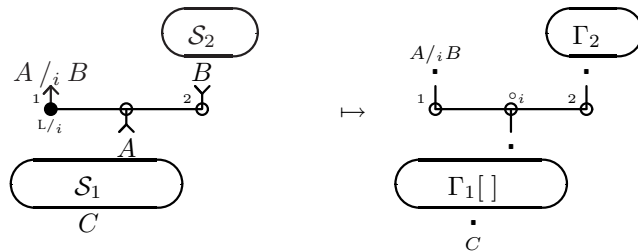
If there are no tensor links, our proof structure looks as follows

$$A \mapsto \begin{array}{c} A \\ \bullet \\ A \end{array}$$

and the corresponding sequent proof is  $\frac{}{A \vdash A} [Ax]$ .

If there are tensor links, then by Lemma 7.14 we know the proof structure has at least one main leaf, call it  $D$ .

In the sub-case where  $D$  is the main formula of a  $[L/i]$  link,  $D$  is of the form  $A/iB$  and must be the first premiss of this link. Now the proof structure  $S$  and the underlying hypothesis tree  $\Gamma_C$  are of the form



By induction hypothesis there are derivations  $\mathcal{D}_2$  of  $\Gamma_2 \vdash B$  and  $\mathcal{D}_1$  of  $\Gamma_1[A] \vdash C$ , which may be combined into

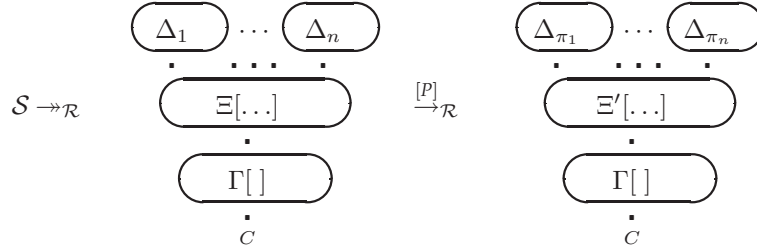
$$\frac{\begin{array}{c} \vdots \mathcal{D}_2 \\ \Gamma_2 \vdash B \end{array} \quad \begin{array}{c} \vdots \mathcal{D}_1 \\ \Gamma_1[A] \vdash C \end{array}}{\Gamma_1[(A/iB \circ_i \Gamma_2)] \vdash C} [L/i]$$

which is a derivation of  $\Gamma \vdash C$ .

The remaining sub-cases, where  $D$  is the main formula of a  $[R\Diamond_j]$ ,  $[R\bullet_i]$ ,  $[L\Box^j]$  or  $[L\setminus_i]$  link, are proved similarly.

$[l > 0]$  We look at the last conversion in the sequence.

If it is a structural conversion  $\Xi[x_1, \dots, x_n] \xrightarrow{[P]}_{\mathcal{R}} \Xi'[x_{\pi_1}, \dots, x_{\pi_n}]$ , we are in the following situation.



The induction hypothesis gives us a derivation  $\mathcal{D}_1$  of

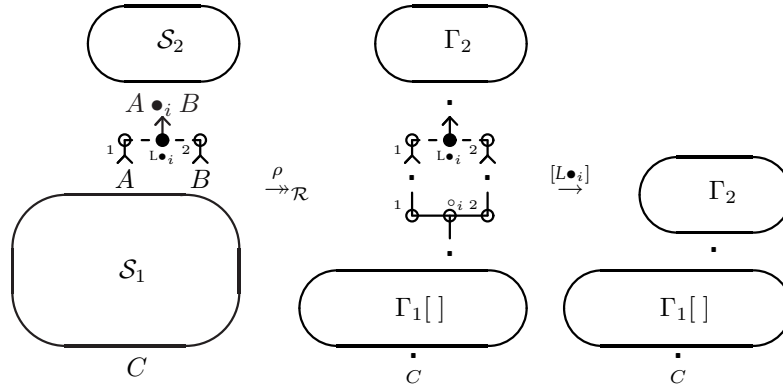
$$\Gamma[\Xi[\Delta_1, \dots, \Delta_n]] \vdash C$$

which we can extend as follows

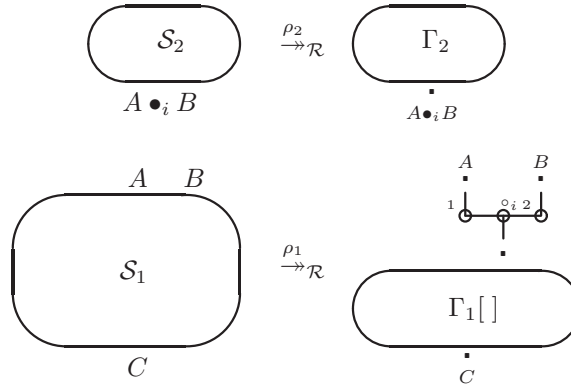
$$\frac{\begin{array}{c} \vdots \mathcal{D}_1 \\ \Gamma[\Xi[\Delta_1, \dots, \Delta_n]] \vdash C \end{array}}{\Gamma[\Xi'[\Delta_{\pi_1}, \dots, \Delta_{\pi_n}]] \vdash C} [P]$$

to give us a derivation of  $\Gamma \vdash C$ .

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



Reasoning backwards from the hypothesis tree, we can see that the  $[L\bullet_i]$  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  $\rho$  into a conversion sequence  $\rho_1$  which converts  $S_1$  to a hypothesis tree and a conversion sequence  $\rho_2$  which converts  $S_2$  into a hypothesis tree as follows



As both conversion sequences are strictly smaller than our initial conversion sequence, the induction hypothesis gives us a derivation  $\mathcal{D}_2$  ending in  $\Gamma_2 \vdash A \bullet_i B$  and a derivation  $\mathcal{D}_1$  ending in  $\Gamma_1[(A \circ_i B)] \vdash C$ . We can combine these derivations as shown below

$$\frac{\frac{\Gamma_2 \vdash A \bullet_i B \quad \frac{\Gamma_1[(A \circ_i B)] \vdash C}{\Gamma_1[A \bullet_i B] \vdash C} [L\bullet_i]}{\Gamma_1[\Gamma_2] \vdash C} [Cut]}{\Gamma_1[\Gamma_2] \vdash C} [Cut]$$

The other contractions are similar.  $\square$

The sequentialization part of our proof has the somewhat inelegant property that it produces sequent proofs which use the  $[Cut]$  rule even for proof nets without cut formulas. However, we can refine the proof of Theorem 7.15 in such a way that the sequent proofs we produce have exactly one  $[Cut]$  rule application for each cut formula. For this purpose we first state the following lemma.

**Lemma 7.16 (Substitution)** *Let  $\mathcal{D}_1$  be a derivation of  $\Gamma_1 \vdash C_1$  and  $\mathcal{D}_2$  be a derivation of  $\Gamma_2[C_1] \vdash C_2$ .*

- (i) *If  $C_1 \vdash C_1$  is an axiom of  $\mathcal{D}_1$ , the succedent formula of which coincides with the succedent formula of  $\Gamma_1 \vdash C_1$ , then we can substitute  $\mathcal{D}_2$  into  $\mathcal{D}_1$  in order to get a derivation  $\mathcal{D}_1[\mathcal{D}_2]$  of  $\Gamma_2[\Gamma_1] \vdash C_2$ .*

$$\frac{\begin{array}{c} C_1 \vdash \mathbf{C}_1 \\ \vdots \\ \Gamma_1 \vdash \mathbf{C}_1 \end{array} \quad \begin{array}{c} \vdots \\ \Gamma_2[\mathbf{C}_1] \vdash C_2 \end{array}}{\Gamma_2[\Gamma_1] \vdash C_2} [\text{Cut}] \quad \text{becomes} \quad \begin{array}{c} \vdots \\ \Gamma_2[\mathbf{C}_1] \vdash \mathbf{C}_2 \\ \vdots \\ \Gamma_2[\Gamma_1] \vdash C_2 \end{array}$$

- (ii) If  $C_1 \vdash C_1$  is an axiom of  $\mathcal{D}_2$ , the antecedent formula of which coincides with the occurrence in  $\Gamma_2[\mathbf{C}_1] \vdash C_2$ , then we can substitute  $\mathcal{D}_1$  into  $\mathcal{D}_2$  in order to get a derivation  $\mathcal{D}_2[\mathcal{D}_1]$  of  $\Gamma_2[\Gamma_1] \vdash C_2$ .

$$\frac{\begin{array}{c} \mathbf{C}_1 \vdash C_1 \\ \vdots \\ \Gamma_1 \vdash C_1 \end{array} \quad \begin{array}{c} \vdots \\ \Gamma_2[\mathbf{C}_1] \vdash C_2 \end{array}}{\Gamma_2[\Gamma_1] \vdash C_2} [\text{Cut}] \quad \text{becomes} \quad \begin{array}{c} \vdots \\ \Gamma_1 \vdash C_1 \\ \vdots \\ \Gamma_2[\Gamma_1] \vdash C_2 \end{array}$$

**Proof** In general every leaf of a tree determines a path to the root. In particular every axiom rule of a derivation determines a path of sequents from that axiom to the conclusion of the derivation. Let  $\Gamma \vdash \Delta$  and  $\Gamma' \vdash \Delta'$  be two successive sequents in a certain path  $\beta$ , i.e.  $\Gamma' \vdash \Delta'$  is the conclusion of an inference rule with  $\Gamma \vdash \Delta$  among its hypotheses.

For a binary inference rule  $R$  we say that  $\beta$  passes  $R$  via the left (right) hypothesis if  $\Gamma \vdash \Delta$  is the first (second) hypothesis of  $R$  with respect to the formulation of Figure 7.5.

- (i) As the occurrence  $\mathbf{C}_1$  is preserved along the path  $\beta$  in  $\mathcal{D}_1$  between  $C_1 \vdash \mathbf{C}_1$  and  $\Gamma_1 \vdash \mathbf{C}_1$ , the possible inference rules  $\beta$  passes are  $[\text{Cut}]$  (via the left hypothesis), the left logical rules (if binary, then via the left hypothesis), or a structural rule. Each of these rules has the property that if

$$\frac{\Gamma_1 \vdash \mathbf{C}_1 \quad (\Gamma_0 \vdash C_0)}{\Gamma_3 \vdash \mathbf{C}_1}$$

is an instance, then so is

$$\frac{\Gamma_2[\Gamma_1] \vdash \mathbf{C}_2 \quad (\Gamma_0 \vdash C_0)}{\Gamma_2[\Gamma_3] \vdash \mathbf{C}_2}$$

- (ii) As the occurrence  $\mathbf{C}_1$  is preserved along the path  $\beta$  in  $\mathcal{D}_2$  between  $\mathbf{C}_1 \vdash C_1$  and  $\Gamma_2[\mathbf{C}_1] \vdash C_2$ , it will never be an active formula in any inference rule  $\beta$  passes. Hence if

$$\frac{\Gamma_2[\mathbf{C}_1] \vdash C_2 \quad (\Gamma_0 \vdash C_0)}{\Gamma_3[\mathbf{C}_1] \vdash C_3}$$

is an instance of a rule, then so is

$$\frac{\Gamma_2[\Gamma_1] \vdash C_2 \quad (\Gamma_0 \vdash C_0)}{\Gamma_3[\Gamma_1] \vdash C_3}$$

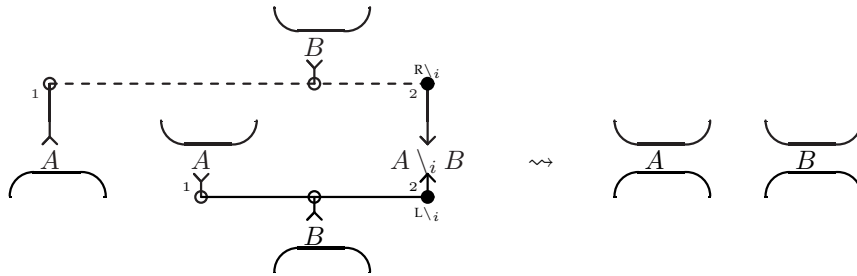
□

Now extend the proof of Theorem 7.15 by simultaneously showing that every axiomatic formula corresponds to an  $[Ax]$  rule, and moreover that every axiomatic conclusion corresponds to an axiom as in Lemma 7.16.1 and every axiomatic hypothesis corresponds to an axiom as in Lemma 7.16.2. We adapt the proof in the case that  $l > 0$  and the last conversion step is a contraction: if the main formula of  $L$  (say:  $D$ ) is a cut formula of  $S$  we proceed as described earlier. However, if  $D$  is not a cut formula, then  $D$  is an axiomatic leaf of one of the two substructures, whence we can apply the par rule followed by the appropriate substitution.

## 7.6 Cut Elimination

One important property of proof net calculi and logics in general is cut elimination. Given a proof net with a number of cut formulas, we want to find a proof net without cut formulas which converts to the same hypothesis tree. In investigating cut elimination, we will give our notion of conversion sequence slightly more structure, touching upon reordering the conversions in such a way that the sequence satisfies certain properties necessary for cut elimination.

Recall that a cut formula is a formula which is the main formula of two dual links. A cut reduction step,  $S \rightsquigarrow S'$ , is defined as deleting these links and the cut formula, while pairwise identifying the active formulas in case they are different (as occurrence of the same formula), or deleting them if they are identical.



Let  $D$  be a cut formula and  $L$  the corresponding par link. We will show that if  $(S, \rho)$  is a conversion sequence ending in hypothesis tree  $\Gamma_C$ , then so is  $(S', \rho')$ , where  $S \rightsquigarrow S'$ , and  $\rho'$  consists of the same set of conversion steps as  $\rho$ , except the contraction  $\alpha$  of  $L$ , in a sense to be made precise shortly.

Before proving the cut elimination theorem, we first introduce the auxiliary notations of component and block and make some observations about their properties.

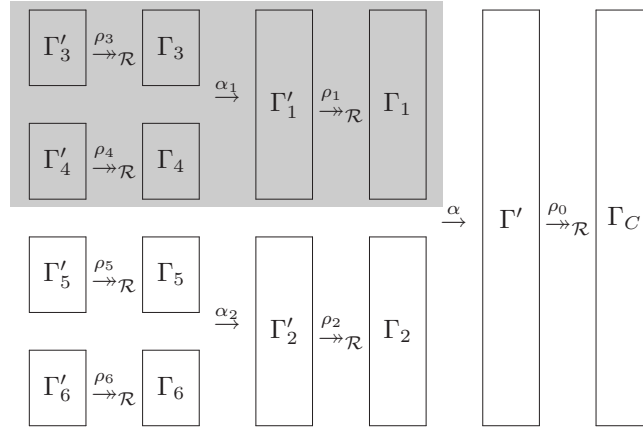


Figure 7.10: The components of a certain conversion sequence with three contractions

**Definition 7.17** Let  $\mathcal{A} = \langle V, \mathcal{L} \rangle$  be an abstract proof structure.  $\mathcal{A}' = \langle V', \mathcal{L}' \rangle$  is a component of  $\mathcal{A}$  if  $\mathcal{A}'$  is a substructure of  $\mathcal{A}$ , where  $\mathcal{L}'$  contains only tensor links and  $\mathcal{A}'$  is maximally connected with respect to the tensor links of  $\mathcal{L}$ .

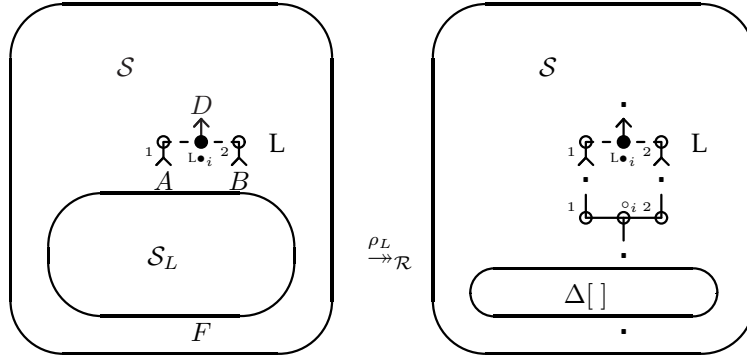
For a proof structure  $\mathcal{S}$ , a substructure  $\mathcal{S}'$  is a component iff  $\widehat{\mathcal{S}}'$  is a component of  $\widehat{\mathcal{S}}$ .

When we delete all  $p$  par links (but not their vertices) from an abstract proof structure  $\mathcal{A}$  the notion of component as defined above coincides with the definition of component from graph theory. Observe that a component may consist of one vertex only.

For a *proof net*, its components are  $p + 1$  hypothesis trees. This even holds for all intermediate abstract proof structures between  $\mathcal{S}$  and  $\Gamma_C$ : reasoning backwards from the final hypothesis tree, we start with one component ( $p = 0$ ). After a number of structural conversions, a contraction  $\alpha$  splits this component  $\Gamma'$  into two parts and replaces one node by a redex. The par link  $L$  of this redex now serves as a boundary between the two new components  $\Gamma_1$ , which is attached to the active formulas  $A$  and  $B$  of  $L$ , and  $\Gamma_2$ , which is attached to the main formula  $D$  of  $L$ . At this moment  $\Gamma_1$  is a *nice* hypothesis tree w.r.t.  $L$ , i.e. attaching  $L$  enables its contraction  $\alpha$ .

All next structural conversions take place completely within one of the two components, and the next contraction takes place in exactly one of the two components as well. In this way every par contraction replaces one component by two new components, yielding  $p + 1$  components in each abstract proof structure.

Figure 7.10 gives an illustration of how we can factor the different conversions of  $\rho$  into components. Another way of seeing this is that  $\rho$  determines a rooted, ordered *tree* of hypothesis trees, with the initial components as the leaves and the final hypothesis tree as the root, where, reasoning backwards

Figure 7.11: The block of a  $[L\bullet_i]$  link  $L$ 

from the final hypothesis tree, every structural conversion produces a unary branch and every contraction a binary branch.

**Definition 7.18 (Block)** Let  $(S, \rho)$  be a conversion sequence ending in  $\Gamma_C$ . Let  $L$  be a par link of  $S$  and  $\alpha$  be the contraction in  $\rho$  corresponding to  $L$ .

The conversion sequence  $\rho$  looks as follows.

$$S \xrightarrow{\rho_1} \mathcal{S}_1 \xrightarrow{\alpha} \mathcal{S}_2 \xrightarrow{\rho_2} \Gamma_C$$

We define a subnet  $(S_L, \rho_L)$  called the block of  $L$  by induction on the length of  $\rho_1$  as follows. As before, when we talk about applying a conversion to a proof structure we will mean applying the conversion to the underlying abstract proof structure.

If  $\|\rho_1\| = 0$  then  $S_L$  is the component of the active formulas of  $L$  and  $\rho_L$  is empty.

If  $\|\rho_1\| > 0$  then we are in the following situation.

$$S \xrightarrow{\delta_0} S' \xrightarrow{\rho'_1} \mathcal{S}_1$$

Induction hypothesis gives us  $(S'_L, \rho'_L)$ , which is the block for the shorter conversion sequence starting with  $S'$ . If the reduct of  $\delta_0$  is not in  $S'_L$  then  $(S_L, \rho_L) = (S'_L, \rho'_L)$ . If the reduct of  $\delta_0$  is in  $S'_L$  then  $\rho_L$  is  $\rho'_L$  with the conversion  $\delta_0$  prefixed and  $S_L$  is  $S'_L$  with the reduct of  $\delta_0$  replaced by its redex.

Figure 7.11 gives an illustration what the block of the  $[L\bullet_i]$  looks like schematically with respect to the full proof net  $S$ . By construction, we can replace the conversion sequence  $\rho$  for  $S$  by the following, where  $\tilde{\rho}_L$  is the conversion sequence which contains all conversions of  $\rho$  not in  $\rho_L$  with the exception of  $\alpha$ .

$$\begin{array}{ccccc} S & \xrightarrow{\rho_L} & \mathcal{A} & \xrightarrow{\alpha} & \mathcal{B} & \xrightarrow{\tilde{\rho}_L} & \Gamma_C \\ \cup & & \cup & & & & \\ S_L & \xrightarrow{\rho_L} & \Delta_F & & & & \end{array}$$

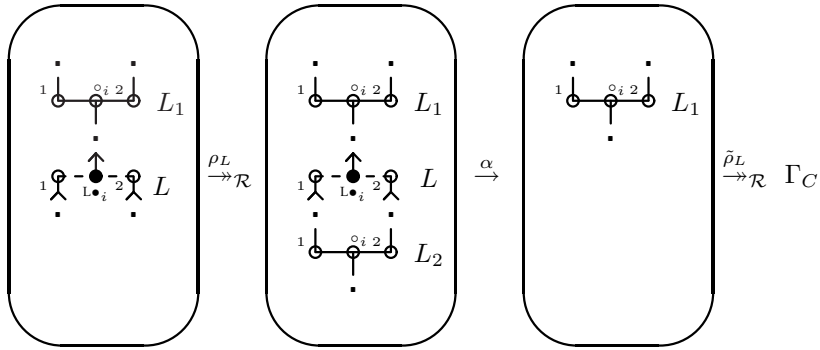
**Example 7.19** In Figure 7.10, assuming  $\Gamma_1$  is the component containing the active formulas of  $L$ , the block of  $L$  is  $(S_L, \rho_L)$ , where  $S_L$  consists of the components  $\Gamma'_3$  and  $\Gamma'_4$  connected by the par link contracted in step  $\alpha_1$  and where the sequence  $\rho_L$  consists of the conversions in the gray area of the figure.

**Theorem 7.20 (Cut elimination)** If  $S$  is a proof net converting to  $\Gamma_C$ , and  $S \rightsquigarrow S'$  by a cut reduction step, then  $S'$  is a proof net converting to  $\Gamma_C$  as well.

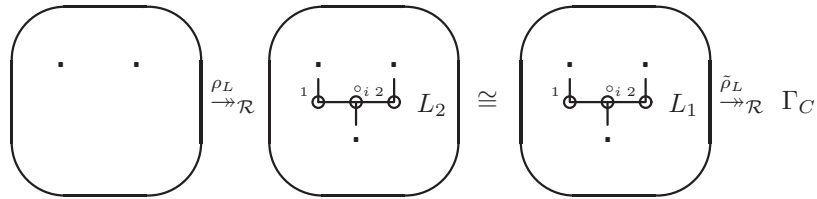
**Proof** Let  $\alpha$  be the contraction corresponding to the par link  $L$  which is removed by the cut reduction step. As observed above, we can replace the conversion sequence  $\rho$  by

$$\widehat{S} \xrightarrow{\rho_L} \mathcal{A} \xrightarrow{\alpha} \mathcal{B} \xrightarrow{\tilde{\rho}_L} \Gamma_C$$

that is, we are schematically in the following situation, where  $L_1$  remains untouched during  $\rho_L$ .



Executing a cut reduction step yields the situation pictured below.



which proves the result. □

## 7.7 Abstract Proof Structures and Labels

We will now briefly sketch the relationship between the structural labels of the previous chapter and the abstract proof structures of the current chapter.

The basic idea is that for a given structural label every sublabel corresponds to a vertex in the corresponding abstract proof structure and every



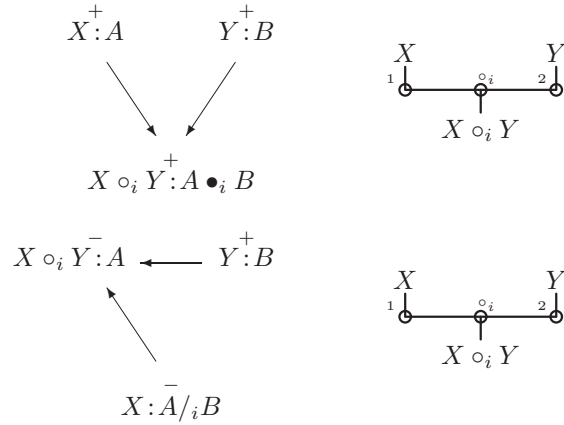


Table 7.10: Structural labels and abstract proof structures: tensor cases

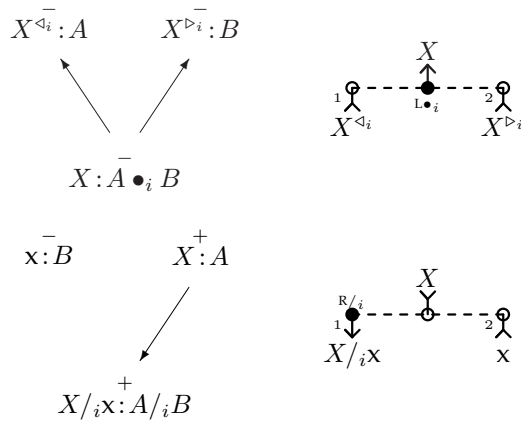


Table 7.11: Structural labels and abstract proof structures: par cases

auxiliary constructor corresponds to a par link in the abstract proof structure and the other constructors correspond to tensor links.

First, note that the dynamic graphs for the axiom and the cut link, shown in Table 6.3 on page 94 have no reflex of this link in the structural label. Similarly, axiom and cut do not correspond to a link in the abstract proof structures.

Now compare the dynamic graphs for the tensor links of ‘ $\bullet$ ’ and ‘ $/$ ’ to the corresponding abstract proof structures. Table 7.10 shows this relation. To make the relation more clear we label the vertices of the aps with the corresponding term label. The formula which is labeled with the complex label is always the conclusion of the link in the abstract proof structure.

Finally, compare the labels for the par links for ‘ $\bullet$ ’ and ‘ $/$ ’ with the corresponding abstract proof structure. Table 7.11 shows them next to each other.

As shown in the table, the single  $[L \bullet_i]$  link corresponds two auxiliary constructors: the  $X^{<i}$  and the  $X^{>i}$  constructor. Because the structural labels are

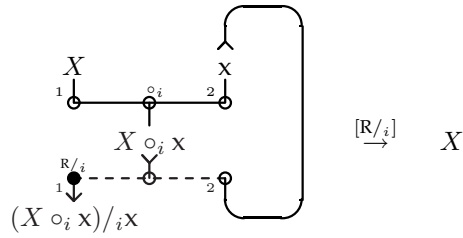


Figure 7.12: Contraction for a  $[R/i]$  link with labeling

$\frac{}{A \vdash A} [Ax]$	$\frac{\Gamma, A, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, \Delta, \Gamma' \vdash C} [Cut]$
$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \bullet B, \Gamma' \vdash C} [L\bullet]$	$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \bullet B} [R\bullet]$
$\frac{\Gamma, A, \Gamma' \vdash C \quad \Delta \vdash B}{\Gamma, A/B, \Delta, \Gamma' \vdash C} [L/]$	$\frac{C, \Gamma, B \vdash A}{C, \Gamma \vdash A/B} [R/]$
$\frac{\Gamma, A, \Gamma' \vdash C \quad \Delta \vdash B}{\Gamma, \Delta, B \setminus A, \Gamma' \vdash C} [L\setminus]$	$\frac{B, \Gamma, C \vdash A}{\Gamma, C \vdash B \setminus A} [R\setminus]$

Table 7.12: The sequent calculus **L**

essentially trees, the par link for  $[L\bullet_i]$  cannot be expressed by a single constructor. In the abstract proof structure, we treat the two occurrences of the  $X$  label as the same vertex. For the  $[R/i]$  link the  $x$  label will also occur twice in the structural label and we will treat both occurrences as the same vertex in the abstract proof structure.

When we compare the label conversions of Table 6.6 on page 97 to the graph contractions of Table 7.9 on page 112 we note that they express the same restrictions. Figure 7.12 shows the  $[R/i]$  contraction on an abstract proof structure, again with the structural labels on the vertices to make the correspondence more clear.

## 7.8 Lambek Calculus

In this section a contraction criterion for the Lambek calculus will be formulated and proved. This criterion is a combination of Danos' contraction criterion for one sided **MLL** (Danos 1990) and Lafont's criterion for parsing boxes (Lafont 1995). The contraction relation is terminating, though not confluent. However, we achieve confluence on a restricted domain, leading us to the main contraction theorem, Theorem 7.28. Our contraction criterion has the special property that a priori there is no order on the leaves of the

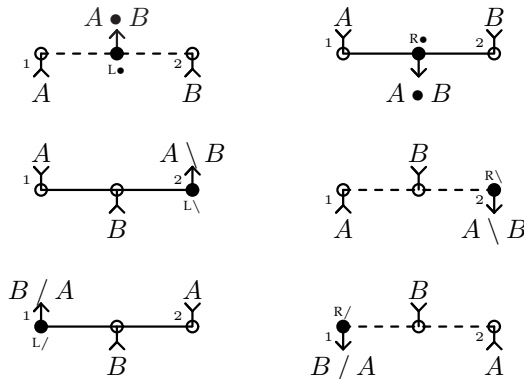
proof structure; if the proof structure is *correct* (in the sense that it contracts properly), our criterion a posteriori provides the unique order of the leaves.

The Lambek calculus (**L**), as introduced by Lambek (1958), is defined by the inference rules of Table 7.12 on the facing page, where the antecedent part of each sequent is a non-empty sequence rather than a structure tree. In our formulation of the calculus, preservation of non-empty antecedent parts during applications of the rules  $[R/]$  and  $[R\setminus]$  is forced by the presence of the extra formula  $C$ .

The Lambek calculus is equivalent to the special case of  $\mathbf{NL}\diamond_{\mathcal{R}}$  with zero unary modes, one binary mode and no structural rules but associativity. The latter mimics the fact that each sequent has a sequence instead of a structure tree as antecedent part. In this way Theorem 7.15 provides us with a correctness criterion for **L**, since **L** derives  $C_1, \dots, C_n \vdash C$  precisely if this special instance of  $\mathbf{NL}\diamond_{\mathcal{R}}$  derives  $C_1 \circ (C_2 \circ (\dots (C_{n-1} \circ C_n) \dots)) \vdash C$ .

However, we can obtain a more attractive correctness criterion when by adapting our theory in such a way that the structural rules become part of the theory and are not present explicitly anymore. This is done by a generalization of the links in the definition of abstract proof structure.

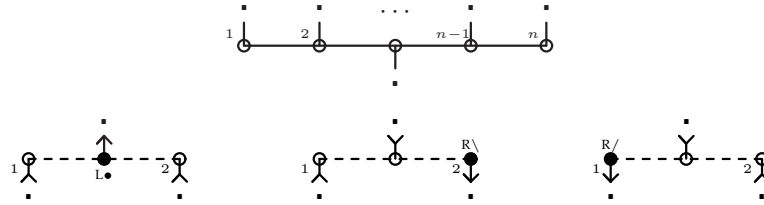
**Definition 7.21** An **L**-proof structure  $\langle S, \mathcal{L} \rangle$  consists of a finite set  $S$  of ( $\diamond$ - and  $\square^\perp$ -free and unimodal) formulas together with a set  $\mathcal{L}$  of links in  $S$  of the following forms.



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.

**Definition 7.22** An abstract **L**-proof structure  $\langle V, \mathcal{L} \rangle$  consists of a finite set  $V$  of nodes together with a set  $\mathcal{L}$  of links in  $V$  of the following forms (where  $n \geq 2$ ).



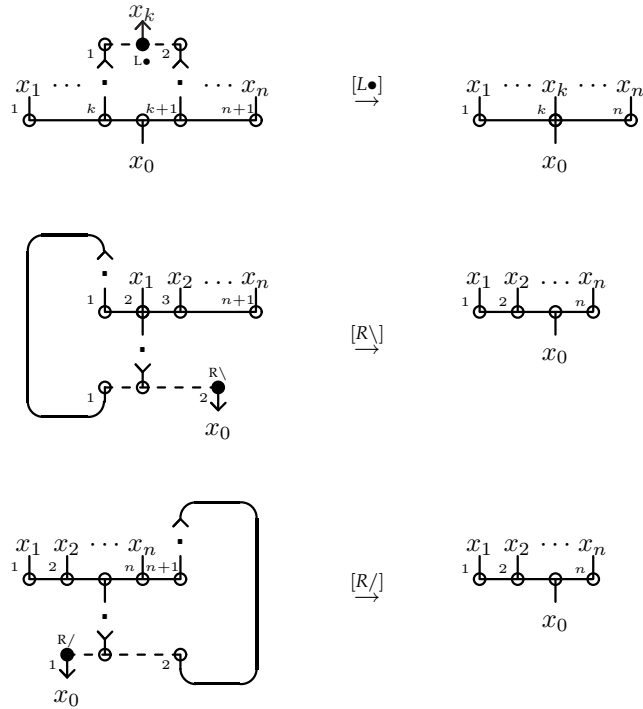
such that the following holds.

- every node of  $V$  is at most once a conclusion of a link,
- every node of  $V$  is at most once a premiss of a link.

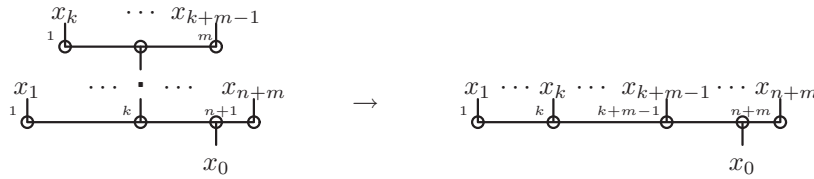
Furthermore, we assign to each node a sequence of premisses and a sequence of conclusions, as in Definition 7.8.

The generalized tensor link will be called an  $n$ -comb ( $n \geq 2$ ). For practical reasons, we define a 1-comb to be a single node; notice that thus a 1-comb is not a link, contrary to  $n$ -combs with  $n \geq 2$ .

The redex of a contraction consists of a par link and an  $(n + 1)$ -comb ( $n + 1 \geq 2$ ), as depicted below (where we require — as usual — all nodes to be distinct). Observe that in every case the par link is attached to two successive formulas of the  $(n + 1)$ -comb, when we order them in a cyclic way. It is replaced by an  $n$ -comb (which is a single node if  $n = 1$ ), and all nodes keep their labels. The contraction will be named after the par link ( $[L\bullet]$ ,  $[R\backslash]$ ,  $[R/]$ ).



By a *structural conversion* we mean the following composition of combs ( $n + 1, m \geq 2$ ).



Now, starting with a proof structure  $\mathcal{S}$ , we can form the underlying abstract proof structure  $\widehat{\mathcal{S}}$  in the usual way (which — besides nodes — consists of par links and 2-combs only).

For any non-empty sequence  $\Gamma$  and formula  $C$ , let  $\|\Gamma\|$  be the multiset of elements in  $\Gamma$ ; let  $\Gamma_C$  be the obvious abstract proof structure (consisting of one  $n$ -comb,  $n \geq 1$ ) with conclusion node (lower) labeled by  $C$ . Any abstract proof structure of this form will be called a *hypothesis comb*. Let  $\rightarrow$  be the transitive, reflexive closure of  $\rightarrow$ , by which we mean the contractions as well as the structural conversions.

It is easy to see that this conversion relation is terminating; in each conversion step at least one link disappears.

**Theorem 7.23**  $\Gamma \vdash C$  is derivable in  $L$  if and only if there is a proof structure  $\mathcal{S}$  such that  $\widehat{\mathcal{S}} \rightarrow \Gamma_C$ .

**Proof** The proof is similar to that of Theorem 7.15: it can be shown that for any derivation  $\mathcal{D}$  of  $\Gamma \vdash C$  the corresponding proof structure converts to the hypothesis comb  $\Gamma_C$ .

The other way around, we can prove that a proof structure  $\mathcal{S}$  that converts to a hypothesis comb  $\Gamma_C$  is actually the proof structure of a derivation  $\mathcal{D}$  of  $\Gamma \vdash C$ . □

Given a proof structure  $\mathcal{S}$  with  $p$  par links, we define a switching  $\omega$  for  $\mathcal{S}$  to be a choice, for each par link  $L$ , of one of the active ends of  $L$ . The *correction graph*  $\omega\mathcal{S}$  of  $\mathcal{S}$  under the switching  $\omega$  is obtained by replacing each par link by the chosen active end. Let  $\mathcal{PS}'$  denote the collection of those elements  $\mathcal{S}$  of  $\mathcal{PS}$ , the set of all proof structures, for which all  $2^p$  correction graphs  $\omega\mathcal{S}$  are trees.

**Lemma 7.24** Let  $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{PS}$  and suppose  $\mathcal{S}_1 \rightarrow \mathcal{S}_2$ . Then  $\mathcal{S}_1 \in \mathcal{PS}'$  if and only if  $\mathcal{S}_2 \in \mathcal{PS}'$ .

In particular, the conversion steps are well defined on  $\mathcal{PS}'$  (i.e. they do yield an element of  $\mathcal{PS}'$  when applied on an element of  $\mathcal{PS}'$ ).

Since hypothesis combs belong to  $\mathcal{PS}'$ , we immediately obtain the next result.

**Corollary 7.25** *If a proof structure  $S$  converts to a hypothesis comb  $\Gamma_C$ , then  $S \in \mathcal{PS}'$ .*

So proof nets (the proof structures that convert to a hypothesis comb) will only be found in  $\mathcal{PS}'$ . Now confluence of this conversion relation on  $\mathcal{PS}'$  is easily proved. This is a consequence of firstly the absence of cycles in the correction graphs, and secondly the absence of the unary connectives which already destroy confluence on general  $\mathbf{NL}\diamond_{\mathcal{R}}$ .

**Lemma 7.26** *If  $S \in \mathcal{PS}'$  converts in one step to  $S_1$  and  $S_2$ , then both  $S_1$  and  $S_2$  convert in at most one step to a common  $S_3 \in \mathcal{PS}'$ .*

By means of Lemma 7.26 and termination, we can sharpen Theorem 7.23 into the following.

**Theorem 7.27** *Let  $\Gamma \vdash C$  be a sequent. Then the following are equivalent.*

- (i)  $\Gamma \vdash C$  is derivable in  $\mathbf{L}$ ;
- (ii) *There is a proof structure  $S$  such that all conversion sequences  $S \twoheadrightarrow S'$  (where  $S'$  is a hypothesis comb) satisfy  $S' = \Gamma_C$ .*

**Theorem 7.28** *Let  $S$  be a proof structure and  $S \twoheadrightarrow S'$  be an arbitrary conversion sequence to a normal form. Then  $S$  is the proof structure of a derivation if and only if  $S'$  is a hypothesis comb.*

Again the following holds.

**Theorem 7.29 (Cut elimination)** *If  $S$  is a proof net converting to  $\Gamma_C$ , and  $S \rightsquigarrow S'$  by a cut reduction step, then  $S'$  is a proof net converting to  $\Gamma_C$  as well.*

## 7.9 Discussion

We have presented a proof net calculus for the multimodal Lambek calculus which is new, elegant and very general. By giving a correctness criterion for  $\mathbf{L}$ , we have also shown how our correctness criterion can function as a sort of meta correctness criterion which can be used to produce a correctness criterion for special instances of  $\mathbf{NL}\diamond_{\mathcal{R}}$ .

The formalism we have presented here is related to a number of other proposals, notably to Danos's (1990) graph contractions, 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.

We have also sketched the relation between abstract proof structures and the structural labels of the labeled proof nets of Moortgat (1997) we discussed in Chapter 6. Advantages of our formalism are that we have a very direct correspondence between proof structures and abstract proof structures and

that cyclic or disconnected proof structures are unproblematically disqualified by our correctness criterion. The algebraic correctness criterion will fail to compute a meaningful label for cyclic or disconnected proof structures.

It is possible to overcome the formal difference between proof structures and abstract proof structures. Puite (2001) introduces the notion of *link graph* for this purpose. Link graphs comprise both proof structures and hypothesis trees, which also play a role as sequents for the calculus. By means of this new notion Puite proves a correctness criterion for **CNL**, the classical non-associative Lambek calculus (de Groote & Lamarche 2001), along the lines of the proof of Theorem 7.15.

