For the multiplicative fragment of linear logic, we have a particularly elegant proof theory, called proof nets. Proof nets were introduced by Girard (1987), both for the multiplicative fragment and for full linear logic, though Girard already noted that the proof net calculus was considerably less elegant for the full system. Improvements to the original formulation of proof nets for the multiplicative fragment are due to Danos & Regnier (1989) and Danos (1990).

Let's take a closer look at the one sided sequent calculus for multiplicative linear logic, repeated here for convenience as Table 4.1. Throughout this chapter, with the exception of Section 4.7 where we discuss non-commutative proof nets, we will keep the commutativity rule implicit.

A problem with this calculus is that a sequent like

\[ \vdash a, a^\perp \otimes b, b^\perp \otimes c, c^\perp \]

has two different sequent proofs, depending on whether we apply the \([\otimes]\) rule first to \(a^\perp \otimes b\) or to \(b^\perp \otimes c\).

\[
\begin{array}{c}
\vdash a, a^\perp \\
\vdash b, b^\perp \\
\vdash c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
\vdash a, a^\perp \otimes b, b^\perp \otimes c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
[\otimes]
\end{array}
\]

\[
\begin{array}{c}
\vdash a, a^\perp \\
\vdash b, b^\perp \\
\vdash c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
\vdash a, a^\perp \otimes b, b^\perp \\
\vdash b, b^\perp \otimes c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
[\otimes]
\end{array}
\]

\[
\begin{array}{c}
\vdash a, a^\perp \\
\vdash b, b^\perp \\
\vdash c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
\vdash a, a^\perp \otimes b, b^\perp \otimes c, c^\perp
\end{array}
\]

\[
\begin{array}{c}
[\otimes]
\end{array}
\]
We would like to claim that the 'essence' of these proofs is the same, that is, all logical rules in both proofs are applied to the same formula occurrences, only the context formulas $\Gamma$ and $\Delta$ of the rules are managed differently. Computationally, this kind of derivational ambiguity, sometimes called spurious ambiguity, is also quite harmful, because when we search for proofs using the sequent calculus, we may find equivalent proofs many, many times.

In a proof net, the different logical rules are applied in parallel, and the result is a system which is much like a natural deduction system with many conclusions.

### 4.1 Proof Nets

The inductive construction rules for proof nets mimic the sequent rules quite closely, but they abstract away from the context formulas.

**Definition 4.1 (Proof Net)** The set of proof nets is inductively defined as follows

[Axiom] If $A$ is a formula of MLL, then the following is a proof net with conclusions $A, \perp$.

$$
\begin{array}{c}
A \\
| \\
\perp
\end{array}
$$

The axiom link is symmetric, i.e. the order of the conclusions of the rule is irrelevant.

[Cut] If $S_1$ is a proof net with conclusion $A$ and $S_2$ is a proof net with conclusion $A^{\perp}$, then we can combine the with a cut link as follows.

$$
\begin{array}{c}
S_1 \\
A \\
| \\
\perp
\end{array} \\
\begin{array}{c}
S_2 \\
A^{\perp} \\
| \\
\perp
\end{array}
$$

Like the axiom link, the cut link is symmetric.
[Par] If $S$ is a proof net with conclusions $A$ and $B$, then we can attach a par link to it as follows.

\[ A \quad B \]

The order of the formulas $A$ and $B$ is relevant, as it determines whether the conclusion of the new proof net is $A \otimes B$ or $B \otimes A$.

[Tensor] If $S_1$ is a proof net with conclusion $A$ and $S_2$ is a proof net with conclusion $B$, then we can combine them with a tensor link as follows.

\[ A \quad B \]

Like the par link, the order of $A$ and $B$ with respect to the tensor link is relevant.

Example 4.2 When we follow the inductive definition to construct the proof net corresponding to the sequent proof on page 49 we see that the proof net will look as shown in Figure 4.1 regardless of the sequence in which we apply the two tensor rules.
The question we are interested in is the following: given a list of formulas, is there a proof net with these formulas as a conclusion? We answer this question in the following way. First, in Section 4.2, we will give a definition of proof structures, candidate proof nets which can be enumerated for any given list of formulas. In Section 4.3, we will give a correctness criterion which will identify the proof nets among the proof structures. Finally, in Section 4.4, we show that the proof net calculus satisfies cut elimination, that is, to show that we can restrict ourselves, without loss of generality, to proof nets which do not contain cut links.

### 4.2 Proof Structures

**Definition 4.3 (Proof Structure)** A proof structure $(S, L)$ consists of a set $S$ of formulas and a set $L$ of links in $S$, where the links are as shown in Table 4.2, Furthermore, a proof structure must satisfy the following conditions.
– every formula is at most once the premiss of a link,
– every formula is exactly once the conclusion of a link.

Formulas which are not the premiss of any link are called the conclusions of a proof structure.

Now, given a sequent, we can enumerate all possible proof structures for it by unfolding all connectives until we reach the atomic formulas and their negations and connect these using axiom links.

### 4.3 Soundness and Completeness

Obviously, not all proof structures are proof nets. For example, given that the sequent \( \vdash (a \land b) \otimes b \land a \) is underivable, the proof structure shown in Figure 4.2 is not a proof net.

We need a criterion which allows us to identify the proof nets from other proof structures. Girard (1987) gave a condition based on travel instructions through a proof net. The criterion presented below is an improvement due to Danos & Regnier (1989). We will discuss another criterion in Section 4.5.

**Definition 4.4** For a proof structure \( S \), a switching \( \omega \) for \( S \) is a choice for every par link of one of its premisses.

**Definition 4.5** From a proof structure \( S \) and a switching \( \omega \) we obtain a correction graph \( \omega S \) by replacing all par links

\[
\begin{array}{c}
B \\
\downarrow
\\
A
\end{array}
\quad
\begin{array}{c}
C
\end{array}
\]

by one of the following links, depending on the premiss of the link selected by \( \omega \).
Figure 4.3: Correction graphs for the incorrect proof structure of Figure 4.2 on the preceding page

For a tensor link with premisses $A$ and $B$ and with conclusion $C$ the correction graph has an (undirected) edge both from $A$ to $C$ and from $B$ to $C$, whereas a par link, being an disjunction, produces an edge either from $A$ to $C$ or from $B$ to $C$.

**Example 4.6** Before sketching the proof of the main theorem, we show that the proof structure above is not a proof net. It has the following two correction graphs.

We can see the correction graph on the right is both cyclic and disconnected and conclude this proof structure is incorrect.

**Lemma 4.7** A proof net without terminal par links has a splitting tensor link, that is, removing the tensor link and its conclusion will yield two disjoint proof nets.

Lemma 4.7 was first proved by Girard (1987). For a proof more in line with the acyclicity and connectedness criterion we are using, we refer the reader to Danos & Regnier (1989) or Bellin & van de Wiele (1995).

**Lemma 4.8** A proof net which has at least one par link has a splitting par link. Removing the par link will yield two disjoint proof nets, where one of the proof nets will have the conclusion $C$ of the par link as a hypothesis, that is, a formula which is not the conclusion of any link.

We refer the reader to Danos (1990) for a proof of this lemma.

The main theorem shows that the proof nets, i.e. the proof structures which are sequentializable, are exactly those proof structures of which all correction graphs are acyclic and connected.
4.3 Soundness and Completeness

Theorem 4.9  A proof structure is a proof net iff all its correction graphs are acyclic and connected.

Proof

\[ \Rightarrow \]  Induction.

\[ \Leftarrow \]  We have to prove that every proof structure of which all correction graphs are acyclic and connected corresponds to a sequent proof. To reduce the number of cases in the proof, we will treat a cut link as a tensor link with a special formula (Cut) as its conclusion.

We use induction on the number of logical links in the proof net. In the case there are no logical links in the proof net, then by disconnectedness it must consist of a single axiom link and the sequent proof corresponding to this proof net consists of just the axiom rule.

If the proof net does contain logical links, there are two basic ways to continue the sequentialization part of the proof. The splitting tensor sequentialization proof uses Lemma 4.7 and is perhaps the most familiar way of proving sequentialization. The splitting par sequentialization proof uses Lemma 4.8.

Splitting Tensor  We proceed by a case analysis; if the proof net has a terminal par link, it must be of the form shown in Figure 4.4.

When we remove the par link and its conclusion, the resulting proof structure is still a proof net, so we can apply the induction hypothesis to give us a sequent proof \( \vdash \Gamma, A, B \), which we can extend as follows.

\[
\begin{array}{c}
\vdash \Gamma, A, B \\
\vdash \Gamma, A \oplus B^{[\vee]} \\
\vdash \Gamma, A \otimes B^{[\otimes]}
\end{array}
\]

When all terminal links are tensor links, then by Lemma 4.7 one of these tensor links is a splitting tensor. That is, a tensor link such that removing it and its conclusion yields two disjoint proof nets.
Not every tensor link is a splitting tensor, because removing an arbitrary tensor link can result in a single, disconnected proof structure. For example, in the case shown in Figure 4.5 removing the leftmost tensor link will generate an incorrect proof structure, because the path from \( a \) to \( b \) through \( a^\perp \otimes b^\perp \) will disappear, making both correction graphs disconnected. However, Lemma 4.7 guarantees the existence of a splitting tensor and in the case above, the other tensor link is indeed splitting.

Given that there is a splitting tensor link, we are schematically in the situation shown in Figure 4.6.

Induction hypothesis gives us a proof \( \mathcal{D}_1 \) of \( \vdash \Gamma; A \) and a proof \( \mathcal{D}_2 \) of \( \vdash B; \Delta \). We can combine these proofs as follows to produce a proof of \( \vdash \Gamma; A \otimes B; \Delta \).

\[
\vdash \Gamma; A \quad \vdash B; \Delta \\
\vdash \Gamma; A \otimes B; \Delta \quad [\otimes]
\]

**Splitting Par** If there are par links in the proof net, Lemma 4.8 guarantees that way are schematically in the case shown in Figure 4.7 on the left,
and removing the par link results in the structure shown in Figure 4.7 on the right.

A problem with the structure on the right is that it is not even a proof structure! This is because the formula $A \& B$ is not the conclusion of any link, violating condition 2 of Definition 4.3. However, we can modify the definitions of proof structures and proof nets to also allow formulas which are not the conclusion of any link. We will call these the hypotheses of the proof structure. We will explore the idea of proof nets with hypotheses further in Chapter 7, where we introduce two sided proof structures and proof nets. A proof structure with hypotheses $\Gamma'$ and conclusions $\Delta$ will correspond to a two sided sequent $\Gamma' \vdash \Delta$.

Given that our proof nets are allowed to have hypotheses, the situation we are in is shown in Figure 4.8.

Now, induction hypothesis gives us a proof $\mathcal{D}_1$ of $\Gamma' \vdash \Gamma, A, B$ and a proof $\mathcal{D}_2$ of $\Delta', A \& B \vdash \Delta$, which we can combine as follows.
If the proof net has no par links, then acyclicity and connectedness guarantee every terminal tensor link is splitting. So we are in the situation shown in Figure 4.9.

Induction hypothesis gives us a proof $D_1$ of $\Gamma' \vdash \Gamma, A$ and a proof $D_2$ of $\Delta' \vdash B, \Delta$. We can combine these proofs as follows to produce a proof of $\Gamma', \Delta' \vdash \Gamma \otimes B, \Delta$.

\[
\frac{\Gamma' \vdash \Gamma, A, B \quad [\otimes]}{\Gamma' \vdash \Gamma, A \otimes B, \Delta} \quad \text{(Cut)}
\]

\[
\frac{\Gamma' \vdash \Gamma, A \quad \Delta' \vdash B, \Delta}{\Gamma', \Delta' \vdash \Gamma \otimes B, \Delta} \quad [\otimes]
\]

\[\Box\]

### 4.4 Cut Elimination

Because we have factored out all rule permutations which were present in the sequent calculus, proving cut elimination for the proof net calculus turns out to be quite simple, as shown by Girard (1987).

Before we prove cut elimination, we show in the following lemma we can restrict ourselves to proof nets with axiom links on atomic formulas without loss of generality.

**Lemma 4.10** For each proof net $S$ there exists a proof net $S'$ with the same conclusions as $S$ where all axiom links are atomic. We call this proof net eta expanded.

**Proof** By induction on the total number of connectives of formulas which are the conclusions of non-atomic axiom links.

A proof net $S$ with at least one complex axiom link has to be of the form.
We can replace this proof net by a proof net with axiom links on the direct subformulas of the conclusion of the axiom link as follows

\[ A \otimes B \rightarrow A \land B \]

where we have reduced the total number of connectives of conclusions of axiom links by two.

We can see that any switching of the new par link will produce a path from the formula \( A \otimes B \) to the formula \( A \land B \), so every correction graph of the new net will be acyclic and connected. \( \square \)

**Theorem 4.11** For every proof net \( S \) there exists a proof net \( S' \) with the same conclusions as \( S \) which does not contain cut links.

**Proof** First, we replace \( S \) by \( S'' \) which is its eta expanded counterpart according to Lemma 4.10. We proceed by induction on the total number \( n \) of connectives of formulas which are premisses to cut links.

\[ n = 0 \] We can successively remove all remaining cut links, which must be of the following form.

\[ A \]

\[ A \otimes B \]

\[ A \land B \]

\[ S_1 \]

\[ S_2 \]
It is easy to see we can replace this proof net by the following, while maintaining acyclicity and connectedness.

\[ S_1 \quad S_2 \]

\[ A \]

\[ S_1 \quad S_2 \quad S_3 \]

\[ A \]

\[ B \]

\[ A \Uparrow \otimes B \Downarrow \]

\[ A \Uparrow B \Downarrow \]

\[ A \Uparrow B \Downarrow \]

\[ n > 0 \] When we have a complex cut, our proof net has to be of the following form. The possibility that \( A \Uparrow \otimes B \Downarrow \) or \( A \Uparrow B \Downarrow \) is the conclusion of an axiom link is excluded because our proof net is eta expanded.

No correction graph of this proof net can have connections between \( S_1 \) and \( S_2 \), other than through \( A \Uparrow \otimes B \Downarrow \), otherwise the proof structure would be cyclic. Similarly for \( S_1 \) and \( S_3 \) and for \( S_2 \) and \( S_3 \). We also know that every correction graph of this proof net needs to have a path from \( A \) to \( B \) which is completely inside \( S_3 \).

Now, when we replace the cut link by the following two cut links, the total complexity decreases by 2.

We show that this new proof structure is also a proof net, that is that for every correction graph and any two vertices in the correction graph
there is a unique path which connects them. Take an arbitrary correction graph and two arbitrary vertices \( v_1 \) and \( v_2 \) in it.

If both are in the same component, they are connected by virtue of being connected for every correction graph of the original proof net.

If \( v_1 \) is in \( S_1 \) and \( v_2 \) in \( S_3 \), then we know there is a unique path from \( v_1 \) to \( A^\perp \) and a unique path from \( v_2 \) to \( A \). We can combine these two paths with the cut link to produce a unique path from \( v_1 \) to \( v_2 \). Similarly for \( v_1 \) in \( S_2 \) and \( v_2 \) in \( S_3 \).

In the final case \( v_1 \) is in \( S_1 \) and \( v_2 \) in \( S_2 \). Now we know there is a unique path from \( v_1 \) to \( A^\perp \), which we extend by the cut link to \( A \). We also know there is a unique path from \( A \) to \( B \) through \( S_3 \), which we extend again with the cut link to \( B^\perp \). Finally we know there is a unique path from \( B^\perp \) to \( v_2 \). Combining these paths gives us the unique path from \( v_1 \) to \( v_2 \) we need. Notice that in the original proof net this path went directly through the formula \( A^\perp \otimes B^\perp \).

\[ \square \]

### 4.5 Contractions

For a proof net with \( p \) par links, there will be \( 2^p \) different correction graphs, so naive application of the acyclicity and connectedness criterion will give us an exponential algorithm. Danos (1990) gives us a computationally more attractive method for checking whether a proof structure is acyclic and connected. In Chapter 7 we will adapt this criterion for the multimodal Lambek calculus.

Starting with the graph of the proof structure, we apply the contractions shown in Figure 4.10, with the following conditions.

1. Only if \( x \neq y \)

2. Only if the two edges come from the same link.

It is important to note the edges of a par link are paired, as suggested by the arc connecting them. This means that when multiple par links have the
same vertex as a base, which can happen after applying some contractions, we keep track of which pairs belong together.

It is immediate that whenever it is possible to apply more than one contraction to a graph, the results will converge as the conflicting contractions will either 1) produce isomorphic graphs immediately, or 2) the two conversions are locally confluent, that is, applying conversion \( a \) after \( b \) and applying conversion \( b \) after \( a \) produces the same result.

Contraction of a proof net will result in a single vertex.

Example 4.12 An example of the contraction of a proof net is shown in Figure 4.11.

Reduction of a proof structure which is not a proof net will result in a graph which is not a single vertex. The incorrect proof structure of Figure 4.2 on page 53 will reduce in 4 steps to

which can be further reduced by a single 1 reduction after which no reductions are possible. This means we can never apply the 2 reduction to eliminate the par link and contract to a single vertex.

Theorem 4.13 A proof structure \( S \) is a proof net iff its graph reduces to a single vertex by applying reductions 1 and 2 above.

Proof
4.6 The Intuitionistic Fragment

We prove that for all proof nets its graph can be reduced to a single vertex. We do this by following the inductive proof net definition.

(Axiom) The proof net consists just of an axiom link, which we can reduce by a 1 reduction to a single vertex.

(Unary) We add a unary link to a proof net which reduces to a vertex. To the resulting graph we can apply a 1 reduction, resulting in a single vertex.

(Times) We have two proof nets reducing to a vertex. After adding the times link we can just apply the 1 reduction two times, and the resulting graph consists of a single vertex.

(Par) We have a proof net which reduces to a single vertex. Adding the par link will give us a 2 redex. Applying a 2 reduction followed by a 1 reduction gives us a single vertex.

(Cut) We have two proof nets reducing to a vertex. We can apply a 1 reduction to the cut link, which gives us a single vertex.

It is easy to see that whenever we apply one of the reductions to reduce a proof structure \( S \) to a proof structure \( S' \) then \( S \) is acyclic and connected iff \( S' \) is. A single vertex is acyclic and connected, so if \( S \) reduces in a number of steps to a single vertex it is acyclic and connected. Application of theorem 4.9 gives us that \( S \) is also a proof net.

Beginning with a proof structure with \( n \) par links and \( m \) times links, we can first reduce all times links in \( O(m) \) time. Then we can find a par link to which reduction 2 and 1 can be applied in at most \( O(n) \) time. If such a par link cannot be found, we fail because the proof structure is disconnected. Reducing all par links will then take \( n + (n - 1) + \ldots + 1 = \frac{1}{2}n(n + 1) \) time. The maximum time for determining a proof structure is a proof net will then be \( O(\frac{1}{2}n(n + 1) + m) = O(n^2) \).

Two recent algorithms, proposed by Guerrini (1999) and by Murawski & Ong (2000), check correctness for multiplicative proof nets in linear time.

4.6 The Intuitionistic Fragment

It is relatively simple to adapt the classical proof nets we’ve seen so far to the intuitionistic fragment of multiplicative linear logic. This basically amounts to a restriction on the formulas of classical linear logic.

Definition 4.14 A one sided intuitionistic sequent is of the form \( \vdash N_0, \ldots, N_n, P \), where the \( N \) and \( P \) formulas are defined over a set of atomic formulas \( A \) as follows.

\[
N ::= A^\perp \\
| P \otimes N \\
| N \otimes N
\]
\[ \mathcal{P} ::= A \\
| \mathcal{N} \& \mathcal{P} \\
| \mathcal{P} \otimes \mathcal{P} \]

We will call the formulas of \( \mathcal{N} \) negative formulas and the formulas of \( \mathcal{P} \) positive formulas.

With this definition in hand, we can translate a multiplicative intuitionistic sequent

\[ A_1, \ldots, A_n \vdash B \]

into a one sided sequent

\[ \vdash \| A_1 \|^-, \ldots, \| A_n \|^-, \| B \|^+ \]

as follows.

\[
\begin{align*}
\| a \|^+ &= a \\
\| A \& B \|^+ &= \| A \|^\& \| B \|^+ \\
\| A \otimes B \|^+ &= \| A \|^\otimes \| B \|^+ \\
\| a \|^- &= a^+ \\
\| A \& B \|^- &= \| A \|^\& \| B \|^- \\
\| A \otimes B \|^- &= \| A \|^\otimes \| B \|^-
\end{align*}
\]

Note that the translation function \( \| \cdot \|^+ \) produces only formulas of \( \mathcal{P} \) and that the translation function \( \| \cdot \|^\- \) produces only formulas of \( \mathcal{N} \).

Instead of translating an intuitionistic sequent first into a classical sequent and then constructing the proof net, we can also construct an intuitionistic proof net directly, using polarity labels to indicate if we are operating on a positive or a negative formula. The links for an intuitionistic proof structure are shown in Table 4.3.

Using these links, the proof nets we are generating differ from classical proof nets built with intuitionistic formulas only in the formulas.

**Example 4.15** For the intuitionistic sequent

\[ a, a \& (a \& b), (a \& b) \& b \vdash b \]

we can construct the following proof net
Table 4.3: Links for MILL

Translating the sequent above to a classical sequent would produce

$$\vdash a^\perp, a \otimes (a \otimes b^\perp), (a^\perp \multimap b) \otimes b^\perp, b$$

and the corresponding proof net would be the following.

4.7 Non-commutative Proof Nets

As already noted by Girard (1987), it is possible to give a graph theoretic characterization of proof nets for non-commutative multiplicative linear logic as well. Non-commutative, or cyclic linear logic is obtained by replacing the commutativity rule
\[ \vdash \Gamma, B, A, \Delta \quad [P] \]

by the cyclic permutation rule.

\[ \vdash \Gamma, A \quad [Cyc] \]

Cyclic permutations allow us to move formulas to the front or the back of the sequent, but the cyclic ordering of the formulas will remain the same.

For a one sided sequent calculus for cyclic linear logic, we have to realize that negation, when it distributes over a multiplicative connective due to the de Morgan laws, reverses the order of the subformulas, as follows.

\[
\begin{align*}
A &\perp = A

(A \otimes B) &\perp = B \perp \neg A

(A \leadsto B) &\perp = B \perp \otimes A
\end{align*}
\]

Non-commutative proof nets, then, are those for which the axioms links are planar.

**Example 4.16** We can derive the sequent \( \vdash b^\perp, a \leadsto (a^\perp \otimes b) \) in cyclic linear logic as follows.

\[
\begin{align*}
\vdash a, a^\perp &\quad [Ax] \\
\vdash b, b^\perp &\quad [Ax] \\
\vdash a, a^\perp \otimes b, b^\perp &\quad [\otimes] \\
\vdash b^\perp, a, a^\perp \otimes b &\quad [Cyc] \\
\vdash b^\perp, a \leadsto (a^\perp \otimes b) &\quad [\Rightarrow]
\end{align*}
\]

The proof net corresponding this sequent is the following.

\[ 
\begin{array}{c}
\vdash b^\perp, a \leadsto (a^\perp \otimes b) \\
\end{array}
\]

We can translate the formulas of \( L_e \), which is the Lambek calculus which allows for empty antecedent derivations, into formulas of cyclic linear logic as follows.
4.7 Non-commutative Proof Nets

Table 4.4: Links for $L_e$

\[
\begin{align*}
\| A \cdot B \| &= \| A \| \otimes \| B \| \\
\| A/B \| &= \| A \| \uplus (\| B \|)^\perp \\
\| B\backslash A \| &= (\| B \|)^\perp \uplus \| A \| 
\end{align*}
\]

When we look at the non-commutative, intuitionistic fragment of linear logic, we extend the non-commutative links above with polarity marking, just as we did for multiplicative intuitionistic linear logic in the previous section. The links for $L_e$ are shown in Table 4.4.

**Example 4.17** One characteristic non-theorem of $L$ is $A \cdot B \not\vdash B \cdot A$. There is only one proof structure for this sequent, which is shown below and which is not planar.

**Theorem 4.18 (Roorda (1991))** A proof net is valid in $L_e$ iff

- all its correction graphs are acyclic and connected,
- all its axiom links are planar.
See also Lamarche & Retoré (1996) for a good overview of proof nets for the Lambek calculus. Abrusci & Ruet (1999) present a combined commutative/non-commutative version of multiplicative linear logic, essentially a classical version of the intuitionistic calculus of de Groote (1996). They give a proof net calculus for their logic with a correctness criterion based on Girard’s (1987) long trip condition. However, it is unclear if we can extend this methodology to non-associative or multimodal logics.

4.8 Conclusions

We have seen how proof nets function as a sort of parallel proof theory for MLL and how proof nets for MILL can be obtained as a natural fragment of this calculus. We have also seen a natural way of dealing with non-commutativity. In the next chapters we will adapt these proof nets to several different settings. In Chapter 5, we will add the first order quantifiers to the proof net calculus and use them to encode string positions and locality domains. In Chapter 6, we will use labeling as a way of enforcing word order and structural constraints. Finally, in Chapter 7 we will look at a version of the contraction criterion discussed in Section 4.5 which is sound and complete for the multimodal Lambek calculus.