Proof Nets and the Identity of Proofs

Lutz Strassburger

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Proof Nets and the Identity of Proofs

Lutz Straßburger

N° 6013

October 2006

Thème SYM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6013--FR+ENG

Proof Nets and the Identity of Proofs

Lutz Straßburger

Thème SYM — Systèmes symboliques
Projet Parsifal

Rapport de recherche n° 6013 — October 2006 — 74 pages

Abstract: These are the notes for a 5-lecture-course given at ESSLLI 2006 in Malaga, Spain. The URL of the school is `http://esslli2006.lcc.uma.es/`. This version slightly differs from the one which has been distributed at the school because typos have been removed and comments and suggestions by students have been worked in.

The course is intended to be introductory. That means no prior knowledge of proof nets is required. However, the student should be familiar with the basics of propositional logic, and should have seen formal proofs in some formal deductive system (e.g., sequent calculus, natural deduction, resolution, tableaux, calculus of structures, Frege-Hilbert-systems, etc.). It is probably helpful if the student knows already what cut elimination is, but this is not strictly necessary.

In these notes, I will introduce the concept of "proof nets" from the viewpoint of the problem of the identity of proofs. I will proceed in a rather informal way. The focus will be more on presenting ideas than on presenting technical details. The goal of the course is to give the student an overview of the theory of proof nets and make the vast amount of literature on the topic easier accessible to the beginner.

For introducing the basic concepts of the theory, I will in the first part of the course stick to the unit-free multiplicative fragment of linear logic because of its rather simple notion of proof nets. In the second part of the course we will see proof nets for more sophisticated logics.

This is a basic introduction into proof nets from the perspective of the identity of proofs. We discuss how deductive proofs can be translated into proof nets and what a correctness criterion is.

Key-words: Proof nets, cut elimination, identity of proofs, correctness criterion, multiplicative linear logic, pomset logic, classical logic

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Reseaux de démonstration et l'identité des démonstrations

Résumé : Pas de résumé

Mots-clés : Reseaux de démonstration, élimination des coupures, identité des démonstrations, logique lineaire, logique classique

Contents

1 Introduction

1.1 The problem of the identity of proofs

Whenever we study mathematical objects within a certain mathematical theory, we normally know when two of these objects are considered to be the same, i.e., are indistinguishable within the theory. For example in group theory two groups are indistinguishable if they are isomorphic, in topology two spaces are considered the same if they are homeomorphic, and in graph theory we have the notion of graph isomorphism. However, in proof theory the situation is different. Although we are able to manipulate and transform proofs in various ways, we have no satisfactory notion telling us when two proofs are the same, in the sense that they use the same argument. The main reason is the lack of understanding of the essence of a proof, which in turn is caused by the bureaucracy involved in the syntactic presentation of proofs. It is therefore essential to find new presentations of proofs that are "syntax-free", in the sense that they avoid "unnecessary bureaucracy".

Finding such presentations is, of course, of utter importance for logic and proof theory in its own sake. We can only speak of a real *theory of proofs*, if we are able to identify its objects. Apart from that, the problem is of relevance not only for philosophy and mathematics, but also for computer science, because many logical systems permit a close correspondence between proofs and programs. In the view of this so-called *Curry-Howard correspondence*, the question of the identity of proofs becomes the question of the identity of programs, i.e., when are two programs just different presentations of the same algorithm. In other words, the fundamental proof theoretical question on the nature of proofs is closely related to the fundamental question on the nature of algorithms. In both cases the problem is finding the right presentation that is able to avoid unnecessary syntactic bureaucracy.

1.2 Historical overview

Interestingly, the problem of the identity of proofs can be considered older than proof theory itself. Already in 1900, when Hilbert was preparing his celebrated lecture, he considered to add a 24th problem, asking to develop a theory of proofs that allows to compare proofs and provide criteria for judging which is the simplest proof of a theorem.[1] But only in the early 1920s, when Hilbert launched his famous program to give a formalization of mathematics in an axiomatic form, together with a proof that this axiomatization is consistent, formal proof theory as we know it today came into existence.

It was Gödel [Göd31], who first considered formal proofs as first-class citizens of the logical world, by assigning a unique number to each of them. Even though this work destroyed Hilbert's program, the idea of treating proofs as mathematical objects—in the very same way as it is done with formulas—led eventually to our modern understanding of formal proofs. Only a few years later, Gentzen [Gen34] provided the first structural analysis of formal proofs and introduced methods of transforming them. His concept of cut elimination is still the most central target of investigation in structural proof theory. But even after Gentzen's work, the natural question of asking for a notion of identity between proofs seemed silly because there are only two trivial answers: two proofs are the same if they prove the same formula, or, two proofs are the same if they are syntactically equal.

In [Pra71], Prawitz proposed the notion of normalization in natural deduction for determining the identity of proofs: two proofs are the same (in the sense that they stand for the same argument) if and only if they have the same normal form. The normalization process in natural deduction corresponds to Gentzen's cut elimination in the sequent calculus: All auxiliary lemmas are removed from the proof, which then uses only material that does already appear in the formula to be proved. However, normalization does not respect any complexity issues because it is hyper-exponential. This means in particular that all so-called speed-up theorems are ignored. In fact, it can happen that a proof with cuts that fits a page is identified with a cut-free proof that exhausts the size of the universe [Boo84]. Furthermore, it is probably quite difficult to convince a working mathematician of the idea that a cunningly short proof using three clever lemmas should be the same as an extraordinarily long proof that does not use these lemmas, even if it can be obtained from the first one via cut elimination. After all, the main part of the mathematician's work consists of finding the right auxiliary lemmas in the first place.

From the viewpoint of computer science the situation looks similar. Through the Curry-Howard correspondence, formulas become types and proofs become programs. The normalization of the proof corresponds to the computation of the program. Translating Prawitz' idea into this setting means that two programs are the same if and only if they have the same input-output-behavior, which completely disregards any reasonable complexity property.

Independently, Lambek [Lam68, Lam69] proposed an idea for identifying proofs that is based on commuting diagrams in categories seen as deductive systems: two proofs are the same, if they constitute the same morphism in the category. For propositional intuitionistic logic on the one side, and Cartesian closed categories on the other side, the two notions coincide. Similarly, by using *-autonomous categories, one can make the two notions coincide for linear logic (see Section 2.8). But for classical logic, the logic of our every day reasoning, neither notion has a commonly agreed definition (see Section 5).[2]

Unfortunately, the problem of identifying proofs has not received much attention since the work by Prawitz and Lambek. Probably one of the reasons is that the fundamental problem of the bureaucracy involved in deductive systems (in which formal proofs are carried out) seemed to be an insurmountable obstacle. In fact, the problem seems so difficult, that it is widely considered to be "only philosophical". However, behind the undeniable philosophical aspects, the problem clearly is a mathematical one and deserves a rigorous mathematical treatment. The developments in logic, proof theory, and related fields within the last two decades suggest that it is worthwhile to give it a new attack. In these notes we will see some ideas in that direction.

1.3 Proof nets

The term "proof net" has been coined by Girard [Gir87] for his "bureaucracy-free" presentation of proofs in linear logic. He used the term "bureaucracy" for the phenomenon of "trivial rule permutations" in the sequent calculus that do not change the essence of a proof.

In these notes, we will use the term "proof net" is a broader sense: A proof net is a graph theoretical or geometric presentation that captures the essence of a proof and is free of any syntactic bureaucracy. Of course,

[1] This has been discovered by the historian Rüdiger Thiele while studying the original notebooks of Hilbert [Thi03]. The history of proof theory might have taken a different development if Hilbert had included his 24th problem into the lecture.

[2] See also [Doš03] for a comparison of the two notions.

for making this precise, it is necessary to say what is meant by "the essence of a proof" and by "syntactic bureaucracy". This is far from clear and is the target of most of today's research efforts on the problem of the identity of proofs.

Although we use Girard's terminology, some of the ideas and technical breakthroughs that we are going to present are much older. The proof nets for unit-free multiplicative logic (that we use as playground for introducing the theory) are essentially the *coherence graphs* of Eilenberg, Kelly, and Mac Lane [EK66, KM71]. For the case of classical logic, the same idea has been rediscovered under the name of *logical flow-graph* by Buss [Bus91]. A very simplified version of proof nets for classical logic is based on Andrews' *matings* [And76] and Bibel's *connections* [Bib81]. If the proof presentations by Andrews and Bibel (which are identical) are restricted to "a linear version" we can (by using the right notion of correctness) get back Girard's *proof nets*. But these linear proof nets have a much better proof theoretical behavior than the nonlinear (i.e., classical) version. This is the reason why we start our survey with proof nets for linear logic.

2 Unit-free multiplicative linear logic

Unit-free multiplicative linear logic (MLL^-) is a very simple logic, that has nonetheless a well-developed theory of proof nets.[3] For this reason I will use MLL^- to introduce the concept of proof nets.

2.1 Sequent calculus for MLL^-

When we define a logic in terms of a deductive system, we have to do two things. First, we have to define the set of well-formed formulas, and second, we have to define the subset of derivable (or provable) formulas, which is done via a set of inference rules[4]. Here is the necessary data for MLL^-: The set of formulas is defined via

$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{A}^\perp \mid \mathscr{F} \parr \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F}$$

where $\mathscr{A} = \{a, b, c, \ldots\}$ is a countable set of propositional variables, and $\mathscr{A}^\perp = \{a^\perp, b^\perp, c^\perp, \ldots\}$ are their duals. In the following, we will call the elements of the set $\mathscr{A} \cup \mathscr{A}^\perp$ *atoms*.

The (*linear*) *negation* of a formula is defined inductively via

$$a^{\perp\perp} = a \qquad (A \otimes B)^\perp = B^\perp \parr A^\perp \qquad (A \parr B)^\perp = B^\perp \otimes A^\perp$$

Note that we invert the order of the arguments when we take the negation of a binary connective. This is not strictly necessary (since for the time being we stay in the commutative world) but will simplify our life when it comes to drawing pictures of proof nets in later sections.

Here is a set of inference rules for MLL^- given in the formalism of the *sequent calculus*:

$$\mathsf{id}\,\frac{}{\vdash A^\perp, A} \qquad \mathsf{exch}\,\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta} \qquad \parr\,\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \parr B, \Delta}$$
$$\otimes\,\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta} \qquad \mathsf{cut}\,\frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta} \tag{1}$$

Note that the sequent calculus needs (apart from the concept of formula) another kind of syntactic entity, called *sequent*. Very often these are just sets or multisets of formulas. But depending on the logic in question, sequents can be more sophisticated structures like lists or partial orders (or whatever) of formulas. For us, throughout these lecture notes, sequents will be finite lists of formulas, separated by a comma, and written with a $\vdash$ at the beginning. Usually they are denoted by Γ or Δ.

2.1.1 Example If A and B are two different formulas, then

$$\vdash A, B \qquad \vdash A, B, A \qquad \vdash A, A, B$$

are three different sequents.

We say a sequent $\vdash \Gamma$ is *derivable* (or *provable*) if there is a *derivation* (or *proof tree*) with $\vdash \Gamma$ as conclusion. Defining this formally precise tends to be messy. Since the basic concept should be familiar for the reader, we content ourselves here by giving some examples.

[3] One could even say the best developed theory of proof nets among all logics that are out there...

[4] To be precise, one should say *axioms and inference rules*. But we consider here axioms as special kinds of inference rules, namely, those without premises.

2.1.2 Example The two sequents $\vdash a^\perp, a\otimes b^\perp, b\otimes c^\perp, c$ and $\vdash ((a \parr a^\perp)\otimes b)\parr b^\perp$ are provable:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash b^\perp, b$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash c^\perp, c$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash b^\perp, b\otimes c^\perp, c$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp, a\otimes b^\perp, b\otimes c^\perp, c$}
\end{prooftree}
\qquad
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a\parr a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash b^\perp, b$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash (a\parr a^\perp)\otimes b, b^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash ((a\parr a^\perp)\otimes b)\parr b^\perp$}
\end{prooftree}
\tag{2}
$$

Of course it can happen that a sequent or a formula has more than one proof. This is where things get interesting. At least for these course notes. Here are four different proofs of the sequent $\vdash a^\perp \parr(a\otimes a), a\parr(a^\perp\otimes a^\perp)$, three of them do not contain the cut-rule:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a, a^\perp$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp, a\otimes a, a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp\parr(a\otimes a), a^\perp\otimes a^\perp, a$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a, a^\perp\otimes a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a\parr(a^\perp\otimes a^\perp)$}
\end{prooftree}
\tag{3}
$$

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a, a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a, a^\perp\otimes a^\perp, a$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a, a, a^\perp\otimes a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a, a\parr(a^\perp\otimes a^\perp)$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp, a\otimes a, a\parr(a^\perp\otimes a^\perp)$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a\parr(a^\perp\otimes a^\perp)$}
\end{prooftree}
\tag{4}
$$

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a, a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a, a^\perp$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp, a\otimes a, a^\perp$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a^\perp, a^\perp, a\otimes a$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp, a^\perp\parr(a\otimes a)$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a, a^\perp\otimes a^\perp, a^\perp\parr(a\otimes a)$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a\parr(a^\perp\otimes a^\perp), a^\perp\parr(a\otimes a)$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a\parr(a^\perp\otimes a^\perp)$}
\end{prooftree}
\tag{5}
$$

and one does contain the cut-rule:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), (a^\perp\parr a^\perp)\otimes a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp\parr(a\otimes a), ((a^\perp\parr a^\perp)\otimes a)\otimes a^\perp, a$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a, ((a^\perp\parr a^\perp)\otimes a)\otimes a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a, a^\perp$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$\vdash a^\perp, a$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a, a^\perp\otimes a^\perp, a$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a, a, a^\perp\otimes a^\perp$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash a^\perp, a\otimes a, a, a^\perp\otimes a^\perp$}
\LeftLabel{cut}
\BinaryInfC{$\vdash a^\perp, a\otimes a, a, a^\perp\otimes a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a, a^\perp\otimes a^\perp$}
\LeftLabel{exch}
\UnaryInfC{$\vdash a, a^\perp\parr(a\otimes a), a^\perp\otimes a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a\parr(a^\perp\parr(a\otimes a)), a^\perp\otimes a^\perp$}
\LeftLabel{cut}
\BinaryInfC{$\vdash a^\perp\parr(a\otimes a), a, a^\perp\otimes a^\perp$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash a^\perp\parr(a\otimes a), a\parr(a^\perp\otimes a^\perp)$}
\end{prooftree}
\tag{6}
$$

Are these proofs really different? Or are they just different ways of writing down the same proof, i.e., they only seem different because of the syntactic bureaucracy that the sequent calculus forces upon us? In the following, we will try to give a sensible answer to this question, and proof nets are a way to do so.

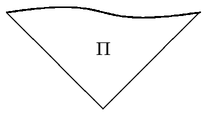
2.1.3 Exercise Give at least two more sequent calculus proofs for the sequent

$$\vdash a^\perp \parr (a \otimes a), a \parr (a^\perp \otimes a^\perp) \quad .$$

2.2 From sequent calculus to proof nets, 1st way (sequent calculus rule based)

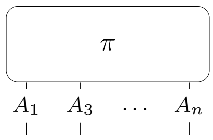
Although, morally, the concept of proof net should stand independently from any deductive formalism, the proof nets introduced by Girard very much depend on the sequent calculus. The ideology is the following:

2.2.1 Ideology *A proof net is a graph in which every vertex represents an inference rule application in the corresponding sequent calculus proof, and every edge of the graph stands for a formula appearing in the proof. A sequent calculus proof with conclusion* $\vdash A_1, A_2, \ldots, A_n$ *, written as*



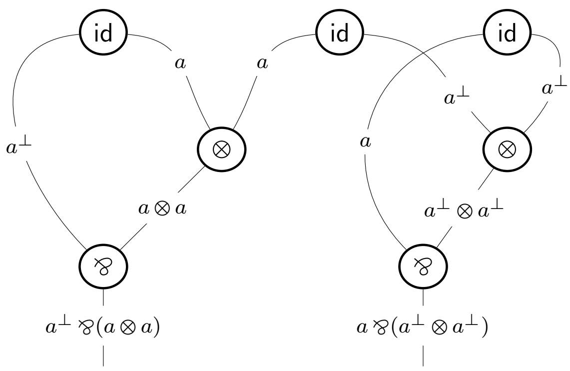
$$\vdash A_1, A_2, \ldots, A_n$$

is translated into a proof net with conclusions $A_1, A_2, \ldots, A_n$*, written as*



This is done inductively, rule instance by rule instance, as shown in Figure 1. Note that the exch-rule does not exactly follow the ideology.

Let us see, what happens if we apply this translation to our four different proofs (3)–(6): The first two, i.e., (3) and (4), both yield the same proof net, namely



(7)

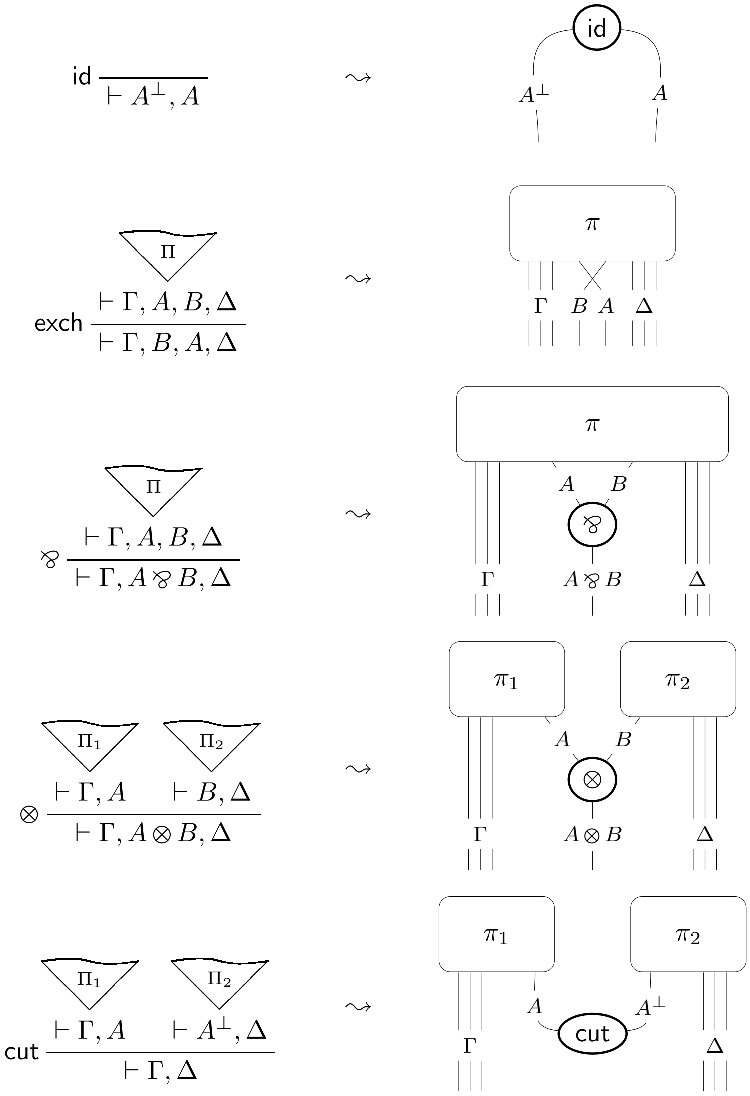
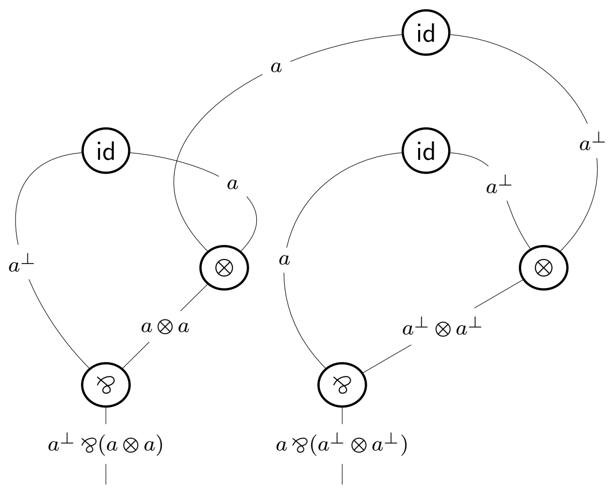
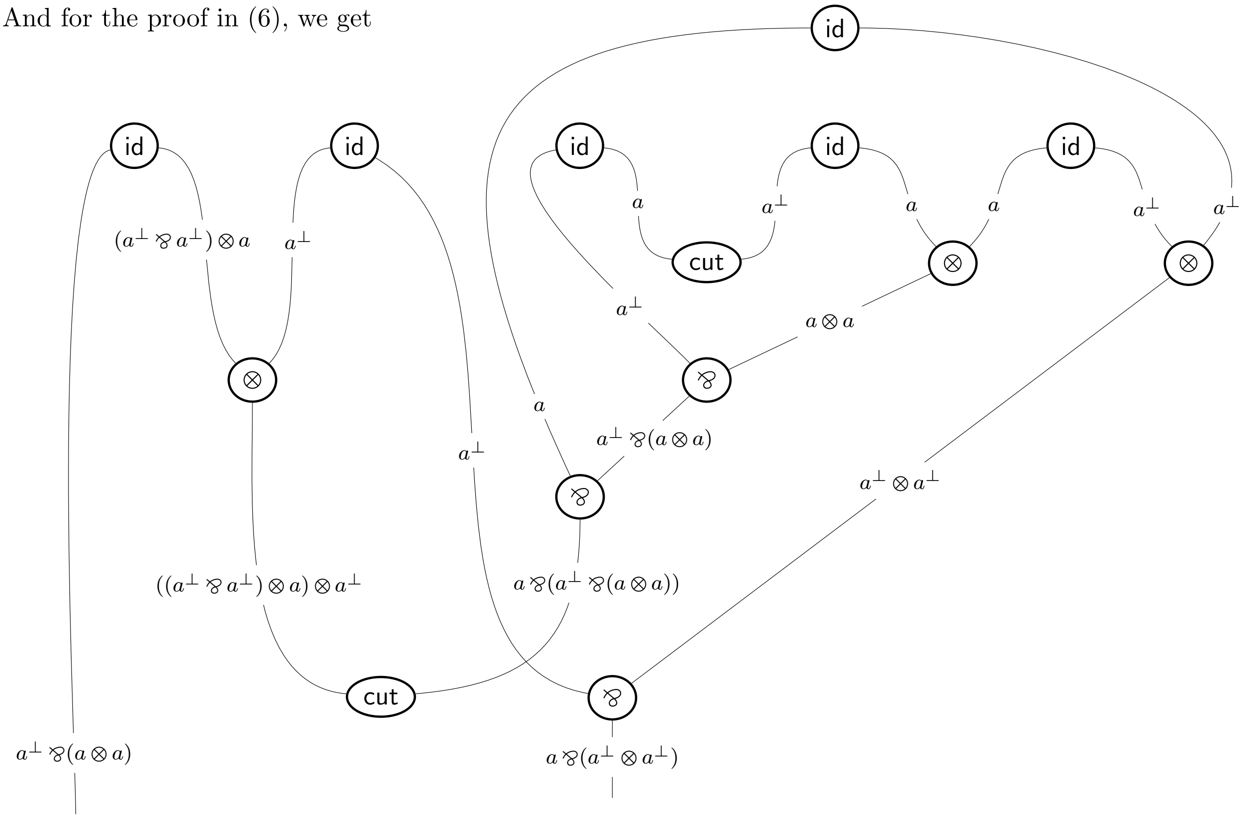


Figure 1: From sequent calculus to proof nets (sequent calculus rule driven)

The proof in (5) yields a different proof net:

 (8)

And for the proof in (6), we get



(9)

The proof nets for (3) and (4) are the same, and the ones for (5) and (6) are different. The big question is now:

2.2.2 Big Question: Is it reasonable to conclude that the two proofs in (3) and (4) are the same, while the one in (5) is different, and the one in (6) is even more different?

For finding an answer, consider the following two (partial) sequent calculus proofs.

$$\parr\,\dfrac{\otimes\,\dfrac{\begin{matrix}\Pi_1\\ \vdash \Gamma, A\end{matrix} \qquad \begin{matrix}\Pi_2\\ \vdash B, \Delta, C, D, \Sigma\end{matrix}}{\vdash \Gamma, A \otimes B, \Delta, C, D, \Sigma}}{\vdash \Gamma, A \otimes B, \Delta, C \parr D, \Sigma} \qquad \text{vs.} \qquad \otimes\,\dfrac{\begin{matrix}\Pi_1\\ \vdash \Gamma, A\end{matrix} \qquad \parr\,\dfrac{\begin{matrix}\Pi_2\\ \vdash B, \Delta, C, D, \Sigma\end{matrix}}{\vdash B, \Delta, C \parr D, \Sigma}}{\vdash \Gamma, A \otimes B, \Delta, C \parr D, \Sigma} \tag{10}$$

It is certainly reasonable to say that the two proofs in (10) are essentially the same—whether we apply first the $\otimes$-rule or first the $\parr$-rule does not matter in this situation.

The phenomenon shown in (10) is called a *trivial rule permutation*. There are clearly more of them, and we will not give a complete list here. The point to note here is that these trivial rule permutation are a type of bureaucracy, which is typical for the sequent calculus, and which was one of Girard's original motivations for the introduction of proof nets. In fact, there is the following theorem:

2.2.3 Theorem *Two sequent calculus proofs using the rules in* (1) *translate into the same proof net if and only if they can be transformed into each other via trivial rule permutations.*

We will not give a proof here because it won't give any new insights. But to illustrate the main point, consider the following three derivations

$$\mathsf{exch}\,\dfrac{\parr\,\dfrac{\vdash B, C, B}{\vdash B \parr C, B}}{\vdash B, B \parr C} \qquad \parr\,\dfrac{\mathsf{exch}\,\dfrac{\vdash B, C, B}{\vdash B, B, C}}{\vdash B, B \parr C} \qquad \parr\,\dfrac{\mathsf{exch}\,\dfrac{\mathsf{exch}\,\dfrac{\vdash B, C, B}{\vdash B, B, C}}{\vdash B, B, C}}{\vdash B, B \parr C}$$

where B and C are arbitrary formulae, and the second instance of exch in the rightmost derivation exchanges the two occurrences of B. Note that the leftmost and the middle derivation both consist of one instance of the $\bindnasrepma$-rule and one instance of exch, but are **not** a trivial rule permutation. When we "trivialy" permute the $\bindnasrepma$-rule and the exch-rule in the leftmost example, then we obtain the rightmost derivation, which contains one $\bindnasrepma$-rule and two exchanges.

Now the reader is invited to do the following exercise:

2.2.4 Exercise Find a sequence of trivial rule permutations that transforms the proof in (3) into the one in (4). Convince yourselves that it is impossible to find a series of rule permutations that converts the proof in (4) into the one in (5).

2.2.5 Exercise Give the proof nets for the two sequent proofs in (2).

2.2.6 Exercise Give the proof nets for the sequent proofs that you found in Exercise 2.1.3, and compare them with (7), (8), and (9).

2.3 From sequent calculus to proof nets, 2nd way (coherence graph based)

Let us now discuss a second method for obtaining a proof net from a sequent calculus proof. Here the ideology is:

2.3.1 Ideology *A proof net consists of the formula tree/sequent forest of the conclusion of the proof, together with some additional graph structure capturing the "essence" of the proof.*

It turns out that for MLL^- the "essence" of a proof is captured by the axiom links. More precisely, the proof net is obtained by drawing the "flow-graph" (or "coherence-graph") through the sequent calculus proof. This means that we trace all atom occurrences through the proof. The idea is quite simple, but again, the formal definitions tend to be messy. For these lecture notes, I decided not to give the detailed definitions but to show the idea via examples. Figure 2 shows how it is done for the examples (3),(4), and (5).

2.3.2 Trivial Observation The flow-graphs drawn inside the proofs have crossings exactly at those places where the exchange rule is applied.

Note that if the id-rule is applied only to atoms, and there is no cut-rule present, then we get (up to some trivial change of notation) *exactly the same proof nets* as with the first method. The reason is that in MLL^- there is a one-to-one correspondence between the binary connectives $\bindnasrepma$ and $\otimes$ appearing in the sequent and the instances of the inference rules for $\bindnasrepma$ and $\otimes$ appearing in the proof. Further, every atom occurrence in the sequent is killed by an instance of the id-rule.

2.3.3 Exercise Convince yourselves that indeed the two methods always yield the same result for cut-free proofs with only atomic instances of identity. Draw the flow graphs for the examples in (2) and the sequent calculus proofs that you found in Exercise 2.1.3.

In Figure 3 we convert the example in (6) into a proof net via the flow-graph method. Note that this time the result is different from the proof net in (9). There are two reasons. First, the non-atomic instance of the id-rule, and second, the presence of the cut-rule.

The non-atomic identity rule is not a problem because an important fact about the sequent calculus system in (1) is that the id-rule can be replaced by its atomic version without changing derivability:

$$\mathsf{id}\ \frac{}{\vdash A^\perp, A} \qquad \leadsto \qquad \mathsf{atomic\ id}\ \frac{}{\vdash a^\perp, a}$$

This is done inductively by systematically replacing

$$\mathsf{id}\ \frac{}{\vdash A^\perp \bindnasrepma B^\perp, B \otimes A} \qquad \text{by} \qquad \bindnasrepma\ \frac{\mathsf{exch}\ \frac{\mathsf{exch}\ \frac{\otimes\ \frac{\mathsf{id}\ \frac{}{\vdash B^\perp, B} \qquad \mathsf{id}\ \frac{}{\vdash A, A^\perp}}{\vdash B^\perp, B \otimes A, A^\perp}}{\vdash B^\perp, A^\perp, B \otimes A}}{\vdash A^\perp, B^\perp, B \otimes A}}{\vdash A^\perp \bindnasrepma B^\perp, B \otimes A} \tag{11}$$

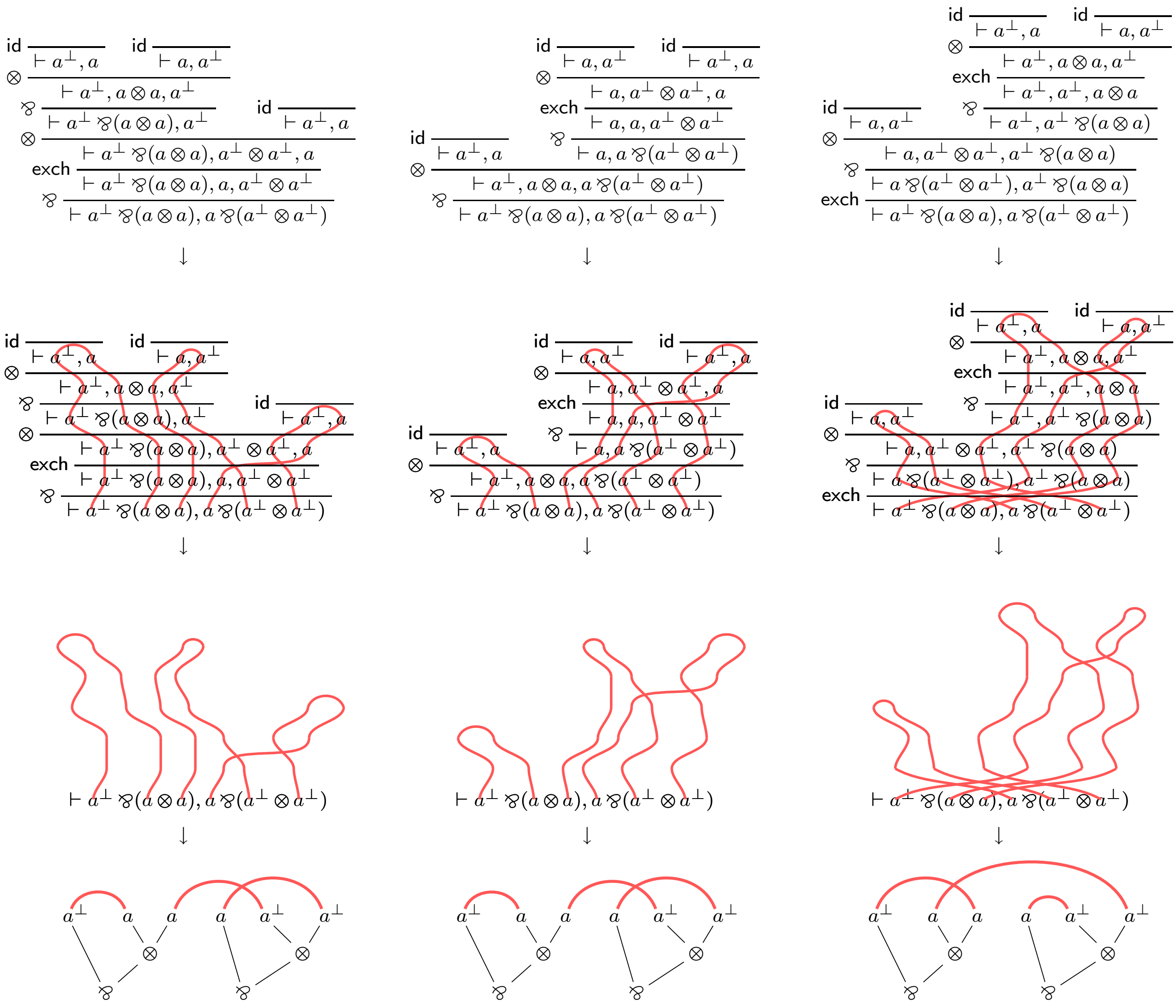


Figure 2: From sequent calculus to proof nets via coherence graphs

2.3.4 Exercise Reduce in (6) the leftmost instance of id to atomic version. And draw the proof net according to the method in Figure 1. What does change compared to the net in (9)?

For dealing with cuts (without forgetting them!), we can prevent the flow-graph from flowing through the cut, i.e., by keeping the information that there is a cut. What is meant by this is shown in Figure 4.

2.3.5 Exercise Compare the net obtained in Figure 4 with your result of Exercise 2.3.4.

Now, we indeed get the same result with both methods, and it might seem foolish to emphasize the different nature of the two methods if they yield the same notion of proof net. The point to make here is that this is the case only for MLL^-, which is a very fortunate coincidence. For any other logic, which is more sophisticated, like classical logic or larger fragments of linear logic, the two methods yield different notions of proof nets. We will come back to this in later sections when we discuss these logics.

2.4 From deep inference to proof nets

The flow graph method has the advantage of being independent from the formalism that is used for describing the deductive system for the logic. We will now repeat exactly the same exercise we did for the sequent calculus

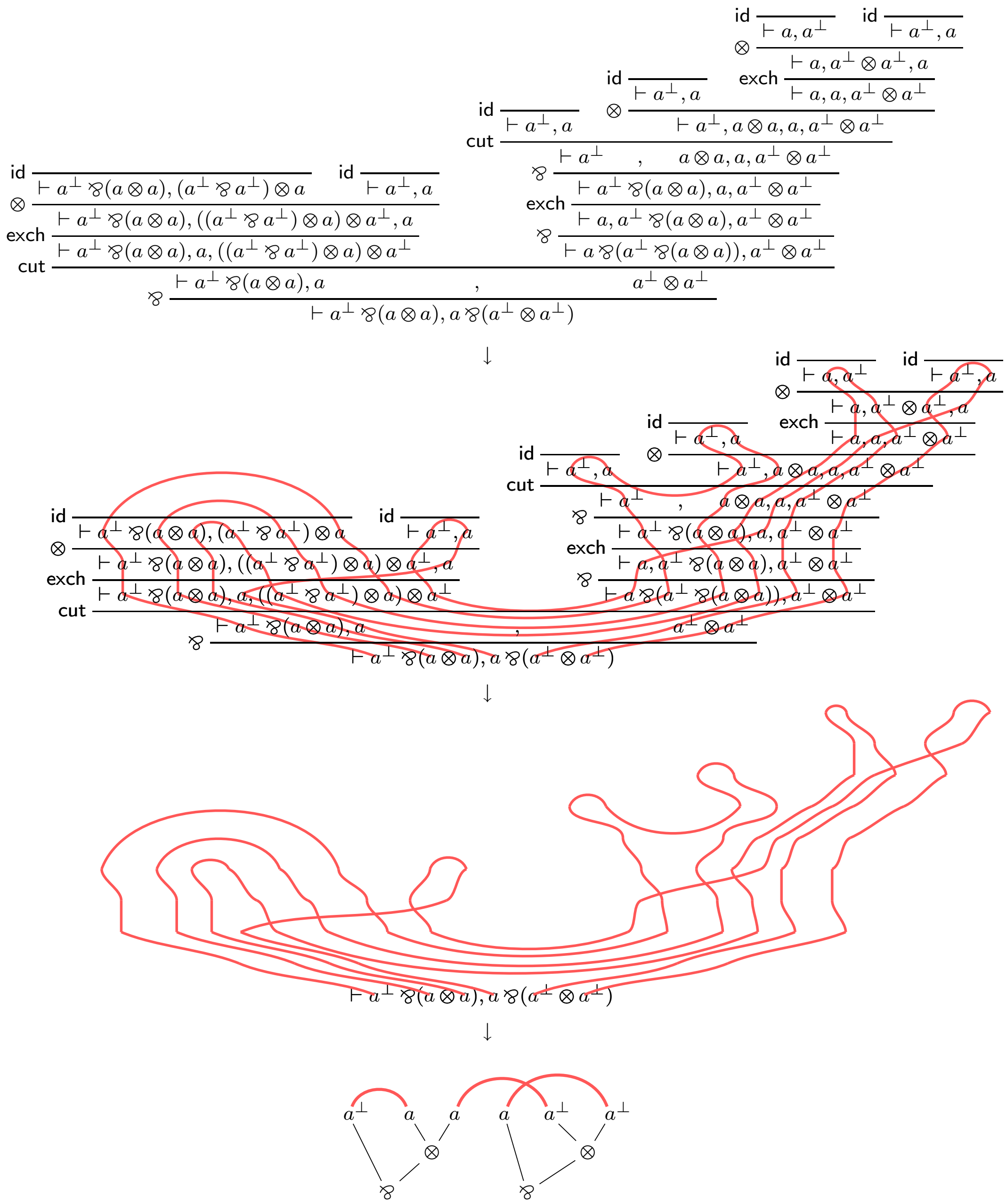


Figure 3: From sequent calculus to proof nets via coherence graphs

system for MLL^- in the previous section. But this time we start from a different deductive system for MLL^-, which is given in the formalism of calculus of structures. It is shown in Figure 5, where $S\{\ \}$ stands for an arbitrary (positive) formula context. Because of the possibility of applying inference rules deep inside any context, the name "deep inference" is used.

Because of this possibility and because we do not have units in the system, we need two variants of the $\mathsf{i}{\downarrow}$-rule (doing the same job as the id-rule in the sequent calculus). Similarly, the cut, i.e., the rule $\mathsf{i}{\uparrow}$ comes in two versions. This also means, that in principle a derivation could have nothing as conclusion. Then we do not have a proof, but a refutation. This leads to an important property of the system in Figure 5, namely its up-down symmetry. For every rule there is a dual co-rule, which is obtained by negating and exchanging premise and conclusion. This flipping around can then be done also for whole derivations. A derivation from A to B becomes a derivation from $B^\perp$ to $A^\perp$, and a proof of a formula A becomes a refutation of $A^\perp$.

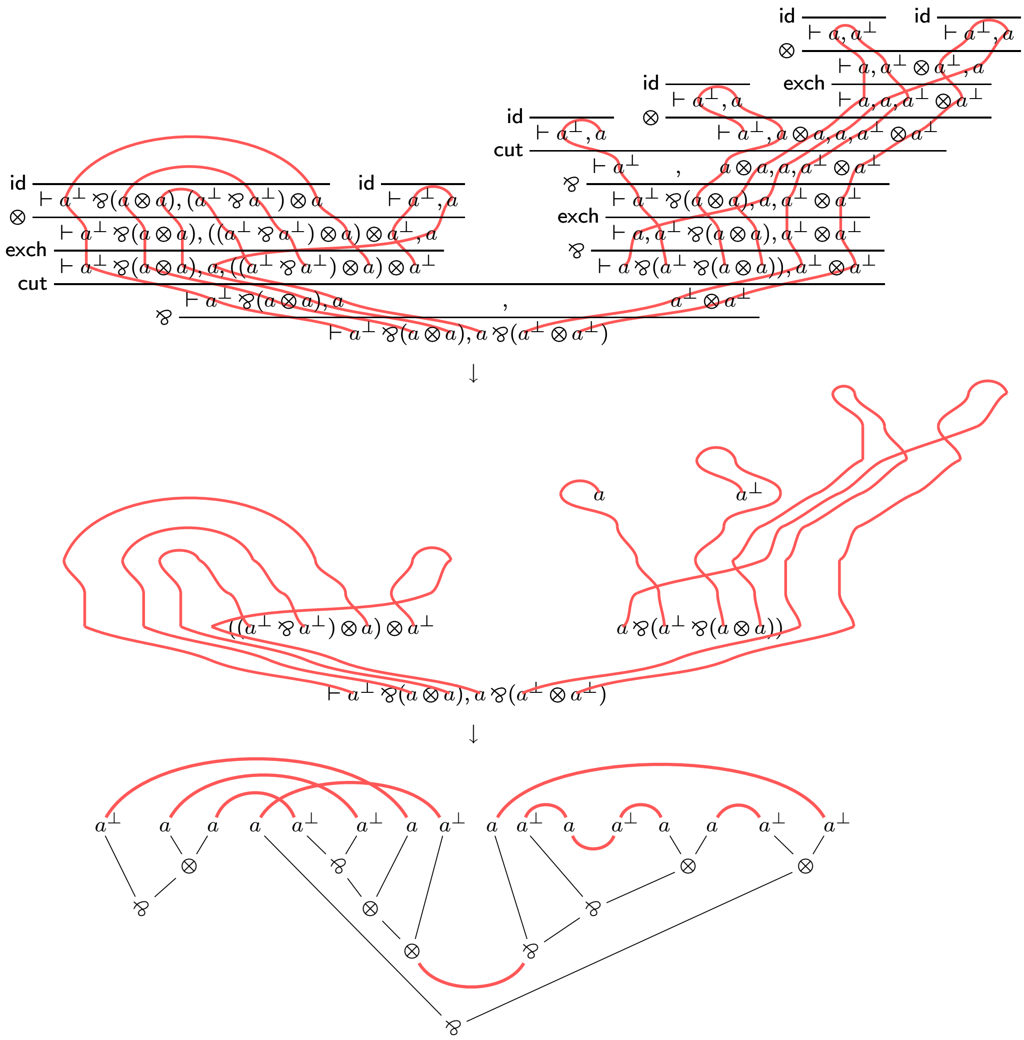


Figure 4: From sequent calculus to proof nets with cuts via coherence graphs

The s-rule (called *switch*) is dual to itself. With the help of commutativity and associativity, i.e., the rule $\sigma\!\downarrow$, $\sigma\!\uparrow$, $\alpha\!\downarrow$, and $\alpha\!\uparrow$ we can obtain the following variants that we also call *switch*:

$$\mathsf{s}\,\frac{S\{(A \parr B)\otimes C\}}{S\{A \parr (B\otimes C)\}} \qquad \mathsf{s}\,\frac{S\{B\otimes (A \parr C)\}}{S\{A \parr (B\otimes C)\}} \qquad \mathsf{s}\,\frac{S\{(A \parr C)\otimes B\}}{S\{(A\otimes B) \parr C\}} \tag{12}$$

Similarly, with the help of $\sigma\!\downarrow$ and $\sigma\!\uparrow$, we can get two other versions if $\mathsf{i}\!\downarrow$ and $\mathsf{i}\!\uparrow$:

$$\mathsf{i}\!\downarrow\frac{S\{B\}}{S\{B\otimes(A^\perp \parr A)\}} \qquad \text{and} \qquad \mathsf{i}\!\uparrow\frac{S\{(A\otimes A^\perp) \parr B\}}{S\{B\}} \tag{13}$$

Here are three examples of derivations in the calculus of structures:

$$\mathsf{i}\!\downarrow\dfrac{\mathsf{i}\!\downarrow\dfrac{}{b \parr b^\perp}}{((a \parr a^\perp)\otimes b) \parr b^\perp} \qquad \mathsf{s}\,\dfrac{\mathsf{i}\!\downarrow\dfrac{\mathsf{i}\!\uparrow\dfrac{\mathsf{s}\,\dfrac{(b^\perp \parr c^\perp)\otimes c}{b^\perp \parr (c^\perp\otimes c)}}{b^\perp}}{b^\perp\otimes(a \parr a^\perp)}}{a \parr (b^\perp\otimes a^\perp)} \qquad \mathsf{s}\,\dfrac{\mathsf{i}\!\downarrow\dfrac{\mathsf{i}\!\uparrow\dfrac{\mathsf{s}\,\dfrac{(a \parr b)\otimes a^\perp}{(a\otimes a^\perp) \parr b}}{b}}{(c^\perp \parr c)\otimes b}}{c^\perp \parr (c\otimes b)} \tag{14}$$

$$
\mathsf{i}{\downarrow}\,\frac{}{A^{\perp}\parr A}
\qquad
\mathsf{i}{\uparrow}\,\frac{A\otimes A^{\perp}}{}
$$

$$
\mathsf{i}{\downarrow}\,\frac{S\{B\}}{S\{(A^{\perp}\parr A)\otimes B\}}
\qquad
\mathsf{i}{\uparrow}\,\frac{S\{B\parr(A\otimes A^{\perp})\}}{S\{B\}}
$$

$$
\sigma{\downarrow}\,\frac{S\{A\parr B\}}{S\{B\parr A\}}
\qquad
\sigma{\uparrow}\,\frac{S\{A\otimes B\}}{S\{B\otimes A\}}
$$

$$
\alpha{\downarrow}\,\frac{S\{A\parr(B\parr C)\}}{S\{(A\parr B)\parr C\}}
\qquad
\alpha{\uparrow}\,\frac{S\{A\otimes(B\otimes C)\}}{S\{(A\otimes B)\otimes C\}}
$$

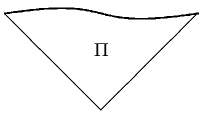
$$
\mathsf{s}\,\frac{S\{A\otimes(B\parr C)\}}{S\{(A\otimes B)\parr C\}}
$$

Figure 5: A system for MLL^{-} in the calculus of structures

The last two are dual to each other.

In order to go on, we have to make sure that the deductive system shown in Figure 5 speaks about the same logic, as the one shown in (1). This does the following theorem:

2.4.1 Theorem *Let A be an MLL^{-} formula. There is a sequent calculus proof of A, denoted by*



$$\vdash A$$

in the system shown in (1)*, if and only if there is a proof of A in the calculus of structures denoted by*

$$
\begin{array}{c}
- \\
\Big\| \;\tilde{\Pi} \\
A
\end{array}
$$

in the system shown in Figure 5.

Proof: For going from sequent calculus to calculus of structures, we show that every rule in (1) can be simulated by the rules in Figure 5. The only interesting cases are $\otimes$ and cut, which are simulated as follows:

$$
\mathsf{s}\,\frac{\mathsf{s}\,\frac{(\Gamma\parr A)\otimes(B\parr\Delta)}{\Gamma\parr(A\otimes(B\parr\Delta))}}{\Gamma\parr(A\otimes B)\parr\Delta}
\qquad\text{and}\qquad
\mathsf{i}{\uparrow}\,\frac{\mathsf{s}\,\frac{\mathsf{s}\,\frac{(\Gamma\parr A)\otimes(A^{\perp}\parr\Delta)}{\Gamma\parr(A\otimes(A^{\perp}\parr\Delta))}}{\Gamma\parr(A\otimes A^{\perp})\parr\Delta}}{\Gamma\parr\Delta}
$$

For the rules $\parr$ and exch it is trivial. Now we can mimic the sequent calculus proof such that different branches in the proof tree, say

$$\vdash\Gamma_1 \qquad \vdash\Gamma_2 \qquad \ldots \qquad \vdash\Gamma_n$$

are kept together in a single formula

$$\Gamma_1\otimes\Gamma_2\otimes\cdots\otimes\Gamma_n$$

In the end, we have for every instance of id in the sequent proof tree and instance of $\mathsf{i}{\downarrow}$ in the calculus of structures proof.

For the other direction, we first have to show for every rule

$$\mathsf{r}\,\frac{S\{A\}}{S\{B\}}$$

in Figure 5, the sequent $\vdash A^{\perp}, B$ is provable with the rules in (1). Here we do it only for the switch rule:

$$\mathord{\parr}\,\dfrac{\mathsf{exch}\,\dfrac{\mathsf{exch}\,\dfrac{\mathord{\parr}\,\dfrac{\otimes\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash C, C^{\perp}} \qquad \mathsf{exch}\,\dfrac{\mathsf{exch}\,\dfrac{\otimes\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash A^{\perp}, A} \qquad \mathsf{id}\,\dfrac{}{\vdash B, B^{\perp}}}{\vdash A^{\perp}, A \otimes B, B^{\perp}}}{\vdash A^{\perp}, B^{\perp}, A \otimes B}}{\vdash B^{\perp}, A^{\perp}, A \otimes B}}{\vdash C, C^{\perp} \otimes B^{\perp}, A^{\perp}, A \otimes B}}{\vdash C, (C^{\perp} \otimes B^{\perp}) \parr A^{\perp}, A \otimes B}}{\vdash (C^{\perp} \otimes B^{\perp}) \parr A^{\perp}, C, A \otimes B}}{\vdash (C^{\perp} \otimes B^{\perp}) \parr A^{\perp}, A \otimes B, C}}{\vdash (C^{\perp} \otimes B^{\perp}) \parr A^{\perp}, (A \otimes B) \parr C}$$

for the other rules it is similar. Now we show that in fact the sequent $\vdash S\{A\}^{\perp}, S\{B\}$ (for every positive context $S\{\ \}$) can be proved with the rules in (1). This is done by structural induction on $S\{\ \}$. If $S\{\ \} = \{\ \}$, we are done. If $S\{\ \} = C \otimes S'\{\ \}$ for some formula C and some (smaller) context $S'\{\ \}$, then we have

$$\mathord{\parr}\,\dfrac{\mathsf{exch}\,\dfrac{\mathsf{exch}\,\dfrac{\otimes\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash C^{\perp}, C} \qquad \begin{array}{c}\Pi'\\ \vdash S'\{B\}, S'\{A\}^{\perp}\end{array}}{\vdash C^{\perp}, C \otimes S'\{B\}, S'\{A\}^{\perp}}}{\vdash C^{\perp}, S'\{A\}^{\perp}, C \otimes S'\{B\}}}{\vdash S'\{A\}^{\perp}, C^{\perp}, C \otimes S'\{B\}}}{\vdash S'\{A\}^{\perp} \parr C^{\perp}, C \otimes S'\{B\}}$$

where Π' exists by induction hypothesis. The other cases, i.e., $S\{\ \} = C \parr S'\{\ \}$, $S\{\ \} = S'\{\ \} \otimes C$, and $S\{\ \} = S'\{\ \} \parr C$, are similar. Now we are ready to simulate the whole derivation. We proceed by induction om the length of $\tilde{\Pi}$ (i.e., the number of rule instances). If the length is 1, then it must be an instance of $\mathsf{i}\!\downarrow$, and we get a sequent calculus proof with id. Now assume the length of $\tilde{\Pi}$ is > 1, i.e., $\tilde{\Pi}$ is of shape

$$\begin{array}{c} \| \,\tilde{\Pi}' \\ \mathsf{r}\,\dfrac{S\{A\}}{S\{B\}} \end{array}$$

Then we can build

$$\mathsf{cut}\,\dfrac{\begin{array}{c}\Pi'\\ \vdash S\{A\}\end{array} \qquad \begin{array}{c}\Pi''\\ \vdash S\{A\}^{\perp}, S\{B\}\end{array}}{\vdash S\{B\}}$$

where Π' exists by induction hypothesis and Π'' by what has been said above. □

2.4.2 Corollary *Let $A_1, A_2, \ldots, A_n$ be a list of* MLL^{-} *formulas. There is a sequent calculus derivation*

$$\begin{array}{c}\Pi\\ \vdash A_1, A_2, \ldots, A_n\end{array}$$

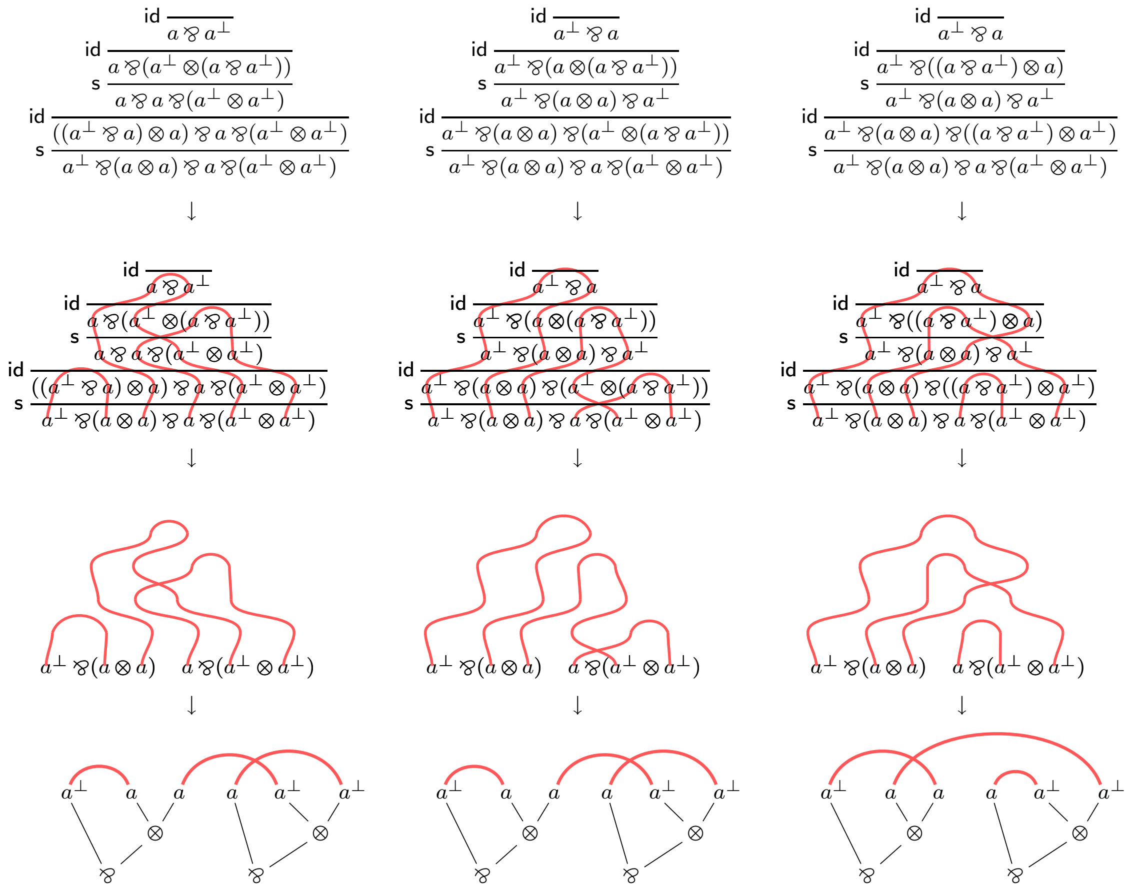


Figure 6: From calculus of structures to proof nets

if and only if there is a derivation

$$\begin{array}{c} \overline{\Big\Vert}\, \tilde{\Pi} \\ A_1 \parr A_2 \parr \cdots \parr A_n \end{array}$$

in the calculus of structures.

Let us now see how derivations in the calculus of structures are translated into proof nets. The method is exactly the same as in the previous section: We simply trace the atoms through the derivation. Figures 6–8 show the calculus of structures version of Figures 2–4.

2.4.3 Trivial Observation We get the same proof nets as before.

2.4.4 Subtle Observation In Figures 6 and 8 the flow-graphs drawn inside the proofs have the same crossings as the resulting proof nets, while in Figures 2 and 4, the flow-graphs drawn inside the proofs have more (seemingly unnecessary) crossings.

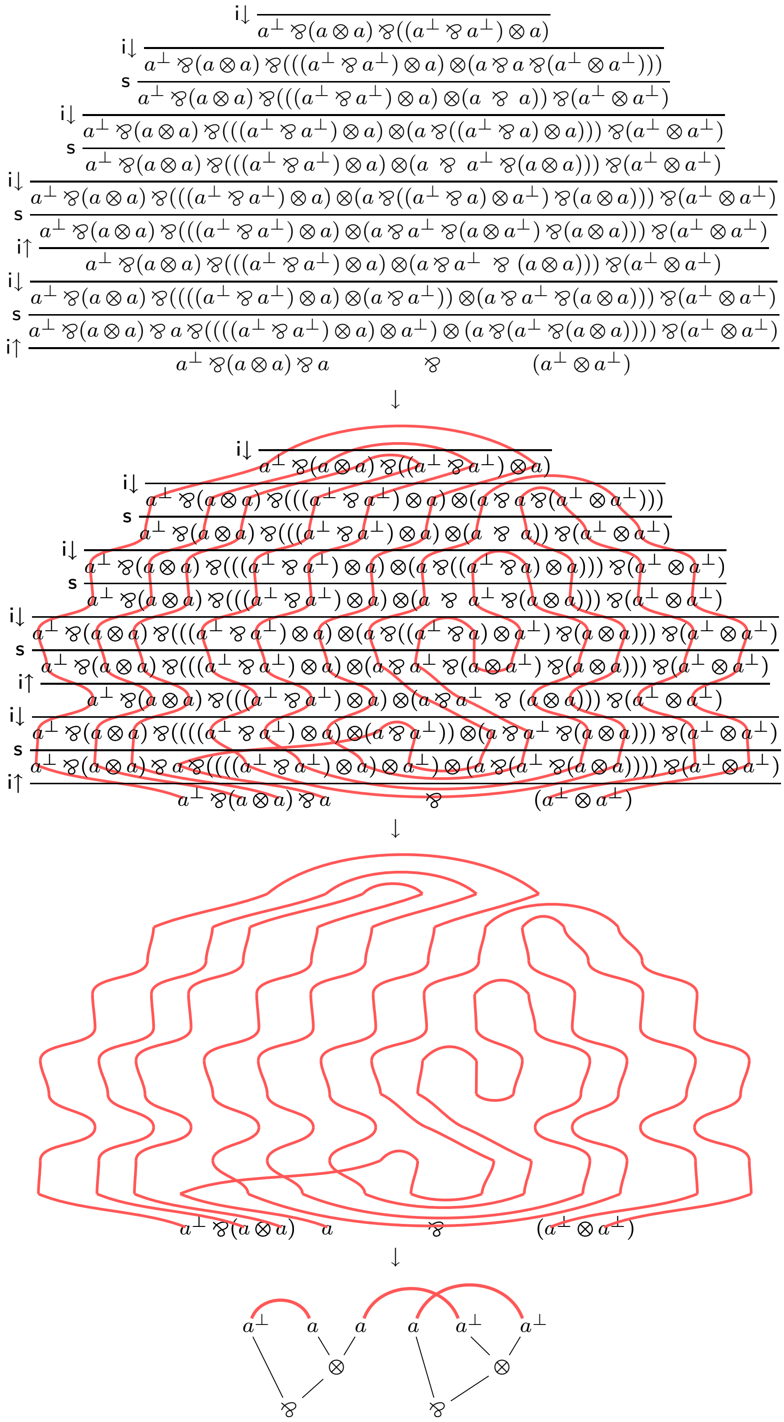


Figure 7: From calculus of structures to proof nets

2.5 Correctness criteria

We have seen how we can obtain a proof net out of a formal proof in some deductive system. But what about the other way around? Suppose we have such a graph that looks like a proof net. Can we decide whether it really comes from a proof, and if so, can we recover this proof? Of course the answer is trivially yes because the graph is finite and we just need to check all proofs of that size. The interesting question is therefore, whether we can do it efficiently.

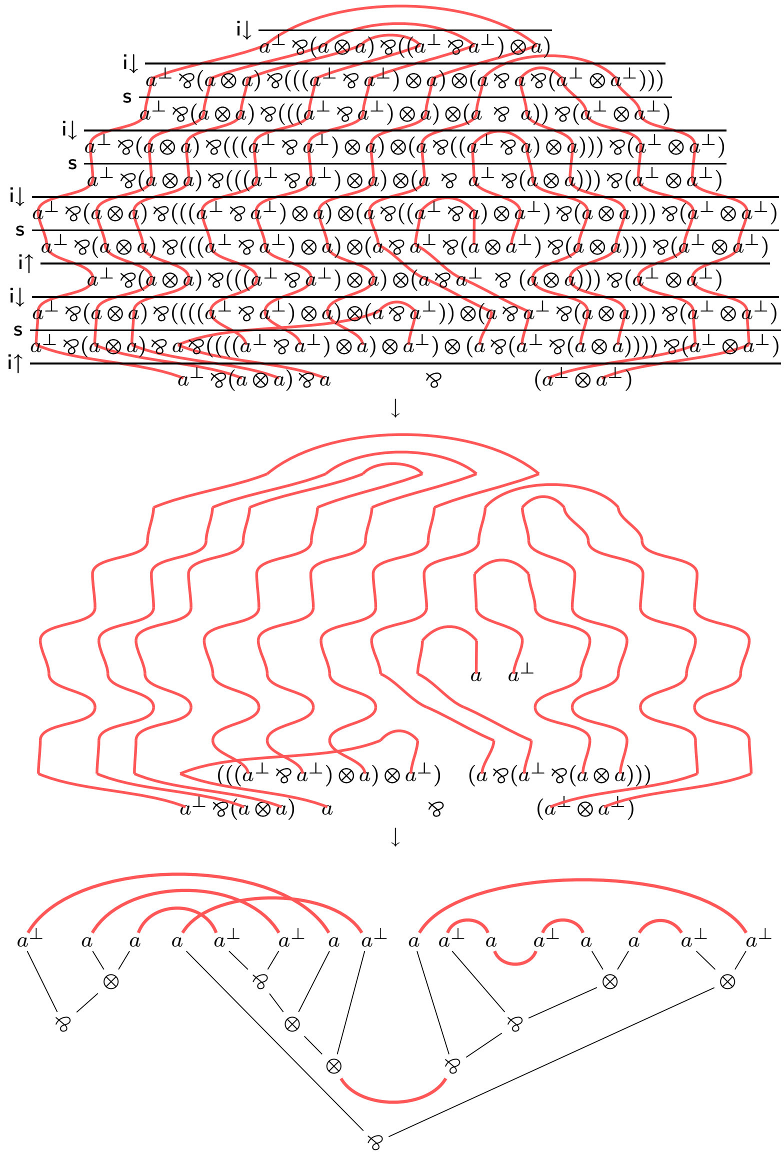
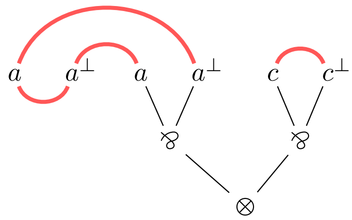
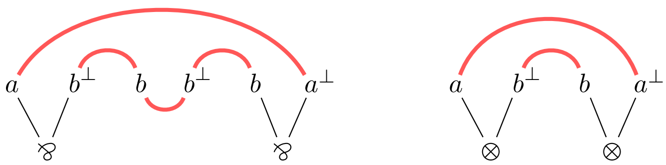


Figure 8: From calculus of structures to proof nets

The answer is still yes, and it is done via so-called *correctness criteria*. For introducing the idea, we take the following graphs as running examples



(15)

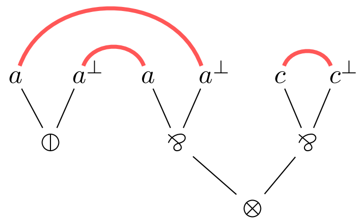


(16)

By playing around, you will notice that it is quite easy to find a proof (in sequent calculus or calculus of structures) that translates into the net in (15), but it seems impossible to find such proofs for the two examples in (16). We are now going to show that this is indeed impossible. For doing so, we need some formal definitions.

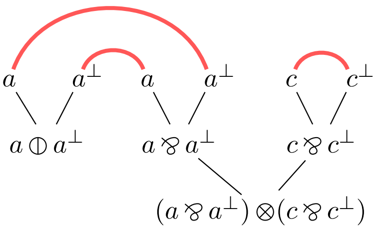
2.5.1 Definition A *pre-proof net* is a sequent forest Γ, possibly with cuts, together with a perfect matching of the set of leaves (i.e., the set of occurrences of propositional variables and their duals), such that only dual pairs are matched.

In this context, a cut must be seen as a special kind of formula $A \varobar A^\perp$, where $\varobar$ is a special connective which may occur only at the root of a formula tree in which the two direct subformulas are dual to each other. For example, (15) should be read as



(15')

Clearly, the examples in (15) and (16) are all pre-proof nets. In the following, we will think of an inner node (i.e., a non-leaf node) of the sequent forest labeled not only by the connective but by the whole subformula rooted by that connective. Our favorite example (15) should then be read as



(15")

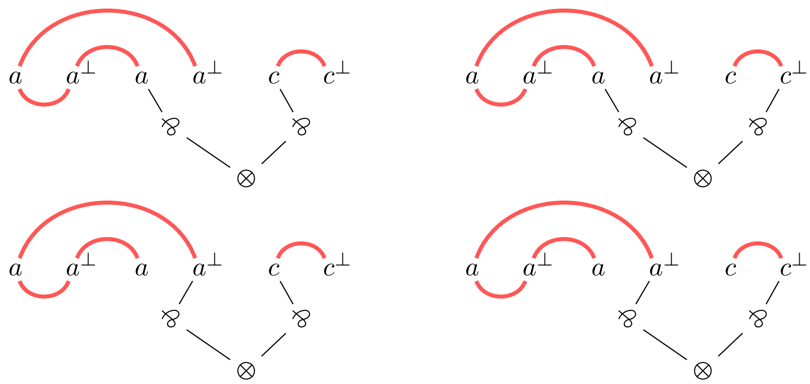
Although sometimes we think of pre-proof nets to be written as in (15"), we will keep writing them as in (15) for better readability.

2.5.2 Definition A pre-proof net π is called *sequentializable* iff there is a proof in the sequent calculus or in the calculus of structures that translates into π.

Originally, the term "sequentializable" was motivated by the name "sequent calculus". But we use it here also if the "sequentialization" is done in the calculus of structures.

2.5.3 Definition Let π be a pre-proof net. A *switching* for π is a graph obtained from π by removing for every $\parr$-node one of the two edges connecting it to its children.

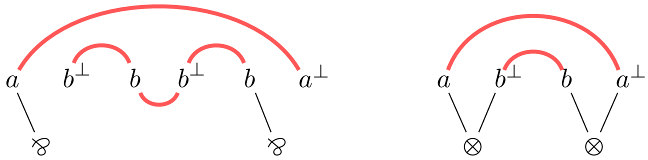
Clearly, if a pre-proof net contains n $\parr$-nodes, then there are 2^n switchings. Here are all 4 switchings for the example in (15):



(17)

2.5.4 Definition A pre-proof net *obeys the switching criterion* (or, shortly, is *correct*) iff all its switchings are connected and acyclic.

As (17) shows, the pre-proof net in (15) is correct. The two pre-proof nets in (16) are not, as the following switchings show:



(18)

The first is not connected, and the second is cyclic.

In the following, we use the term *proof net* for those pre-proof nets which are correct, i.e., obey the switching criterion. The following theorem says that the proof nets are exactly those pre-proof nets that represent an actual proof.

2.5.5 Theorem *A pre-proof net is correct if and only if it is sequentializable.*

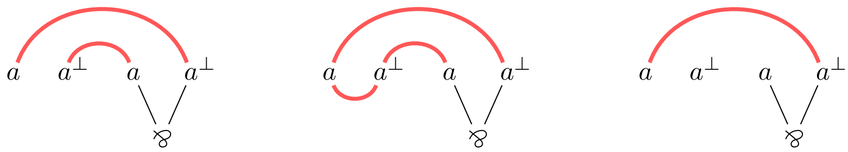
We will give two proofs of this theorem. The first uses the sequent calculus, and the second the calculus of structures. For the first proof, we need the following lemma:

2.5.6 Lemma *Let π be a proof net with conclusions $A_1, \ldots, A_n$. If all A_i have a $\otimes$ or a cut as root, then one of them is splitting, i.e., by removing that $\otimes$ (or $\oplus$), the net becomes disconnected.*

For proving this lemma, we need some more concepts.

2.5.7 Definition Let σ and π be pre-proof nets. We say σ is a *subprenet* of π, written as $\sigma \subseteq \pi$ if all formulas/cuts appearing in σ are subformulas of the formulas/cuts appearing in π, and the linking of σ is the restriction of the linking of π to the formulas/cuts in σ. We say σ is a *subnet* of π if $\sigma \subseteq \pi$, and σ and π are both correct. A *door* of σ is any formula that appears as conclusion of σ.

2.5.8 Example Consider the following three graphs:



The first two are subprenets of (15), the third one is not (because a link is missing). The second one is in fact a subnet of (15), but the first one is not (because it is not correct). The doors of the first example are a, $a^\perp$, and $a \mathbin{⅋} a^\perp$. The doors of the second example are $a \oplus a^\perp$ and $a \mathbin{⅋} a^\perp$.

2.5.9 Lemma *Let σ and ρ be subnets of some proof net π.*

(i) *The subprenet $\sigma \cup \rho$ is a subnet of π if and only if $\sigma \cap \rho \neq \emptyset$.*

(ii) *If $\sigma \cap \rho \neq \emptyset$ then $\sigma \cap \rho$ is a subnet of π.*

Proof: Intersection and union in the statement of that lemma have to be understood in the canonical sense: An edge/node/link appears in in $\sigma \cap \rho$ (resp. $\sigma \cup \rho$) if it appears in both, σ and ρ (resp. in at least one of σ or ρ). For giving the proof, let us first note that because in π every switching is acyclic, also in every subprenet of π every switching is acyclic, in particular also in $\sigma \cup \rho$ and $\sigma \cap \rho$. Therefore, we need only to consider the connectedness condition.

(i) If $\sigma \cap \rho = \emptyset$ then every switching of $\sigma \cup \rho$ must be disconnected. Conversely, if $\sigma \cap \rho \neq \emptyset$, then every switching of $\sigma \cup \rho$ must be connected (in every switching of $\sigma \cup \rho$ every node in $\sigma \cap \rho$ must be connected to every node in σ and to every node in ρ, because σ and ρ are both correct).

(ii) Let $\sigma \cap \rho \neq \emptyset$ and let s be a switching for $\sigma \cup \rho$. Then s is connected and acyclic by (i). Let s_σ, s_ρ, and $s_{\sigma\cap\rho}$, be the restrictions of s to σ, ρ, and $\sigma \cap \rho$, respectively. Now let A and B be two vertices in $\sigma \cap \rho$. Then A and B are connected by a path in s_σ because σ is correct, and by a path in s_ρ because ρ is correct. Since s is acyclic, the two paths must be the same and therefore be contained in $s_{\sigma\cap\rho}$. □

2.5.10 Lemma *Let π be a proof net, and let A be a subformula of some formula/cut appearing in π. Then there is a subnet σ of π, that has A as a door.*

Proof: For proving this lemma, we need the following notation. Let π be a proof net, let A be some formula occurrence in π, and let s be a switching for π. Then we write $s(\pi, A)$ for the graph obtained as follows:

- If A is an immediate subformula of a formula occurrence B in π, and there is an edge from B to A in s, then remove that edge and let $s(\pi, A)$ be the connected component of (the remainder of) s that contains A.
- Otherwise let $s(\pi, A)$ be just s.

Now let

$$\sigma = \bigcap_s s(\pi, A)$$

where s ranges over all possible switchings of π. (Note that it could happen that formally σ is not a subprenet because some edges in the formula trees might be missing. We graciously add these missing edges to σ such that it becomes a subprenet.) Clearly, A is in σ. We are now going to show that A is a door of σ. By way of contradiction, assume it is not. This means there is ancestor B of A that is in $\bigcap_s s(\pi, A)$. Now choose a switching $\hat{s}$ such that whenever there is a $\parr$ node between A and B, i.e.,

$$\begin{array}{ccc} & & A \\ & & \vdots \\ C_1 & & C_2 \\ & \diagdown\ \diagup & \\ & C_1 \parr C_2 & \\ & \vdots & \\ & B & \end{array} \qquad \text{or} \qquad \begin{array}{ccc} A & & \\ \vdots & & \\ C_2 & & C_1 \\ & \diagdown\ \diagup & \\ & C_2 \parr C_1 & \\ & \vdots & \\ & B & \end{array}$$

then $\hat{s}$ chooses C_1 (i.e., removes the edge between C_2 and its parent). Then there must be a $\otimes$ between A and B:

$$\begin{array}{ccc} & & A \\ & & \vdots \\ D_1 & & D_2 \\ & \diagdown\ \diagup & \\ & D_1 \otimes D_2 & \\ & \vdots & \\ & B & \end{array} \qquad \text{or} \qquad \begin{array}{ccc} A & & \\ \vdots & & \\ D_2 & & D_1 \\ & \diagdown\ \diagup & \\ & D_2 \otimes D_1 & \\ & \vdots & \\ & B & \end{array}$$

Otherwise B would not be in σ. Now suppose we have chosen the uppermost such $\otimes$. Then the path connecting A and D_1 in $\hat{s}(\pi, A)$ cannot pass through D_2 (by the definition of $\hat{s}(\pi, A)$). But this means that in $\hat{s}$ there are two distinct paths connecting A and D_1, which contradicts the acyclicity of $\hat{s}$.

Now we have to show that σ is a subnet. Let s be a switching for σ. Since σ is a subprenet of π, we have that s is acyclic. Now let $\tilde{s}$ be an extension of s to π. Then s is the restriction of $\tilde{s}(\pi, A)$ to σ, and hence connected. $\square$

2.5.11 Definition Let π be a proof net, and let A be a subformula of some formula/cut appearing in π. The *kingdom of A in π*, denoted by kA, is the smallest subnet of π, that has A as a door. Similarly, the *empire of A in π*, denoted by eA, is the largest subnet of π, that has A as a door. We define $A \ll B$ iff $A \in kB$, where A and B can be any (sub)formula/cut occurrences in π.

An immediate consequence of Lemmas 2.5.9 and 2.5.10 is that kingdom and empire always exist.

2.5.12 Exercise Why?

2.5.13 Remark The subnet σ constructed in the proof of Lemma 2.5.10 is in fact the empire of A. But we will not need this fact later and will not prove it here.

2.5.14 Lemma *Let π be a proof net, and let A, A', B, and B' be subformula occurrences appearing in π, such that A and B are distinct, A' is immediate subformula of A, and B' is immediate subformula of B. Now suppose that $B' \in eA'$. Then we have that $B \notin eA'$ if and only if $A \in kB$.*

Proof: We have $B' \in eA' \cap kB$. Hence, $\sigma = eA' \cap kB$ and $\rho = eA' \cup kB$ are subnets of π. By way of contradiction, let $B \notin eA'$ and $A \notin kB$. Then ρ has A' as door and is larger than eA' because it contains B. This contradicts the definition of eA'. On the other hand, if $B \in eA'$ and $A \in kB$ then σ has B as door and is smaller than kB because it does not contain A. This contradicts the definition of kB. □

2.5.15 Lemma *Let π be a proof net, and let A and B be subformulas appearing in π. If $A \ll B$ and $B \ll A$, then either A and B are the same occurrence or they are dual atoms connected via an identity link.*

Proof: If a and $a^\perp$ are two dual atom occurrences connected by a link, then clearly $ka = ka^\perp$. Now let A and B be two distinct non-atomic formula occurrences in π with $A \in kB$ and $B \in kA$. Then $kA \cap kB$ is a subnet and hence $kA = kA \cap kB = kB$. We have two cases:

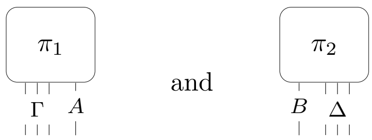
- If $A = A' \parr A''$ then the result of removing A from kB is still a subnet, contradicting the minimality of kB.
- If $A = A' \otimes A''$ then $kA = kA' \cup kA'' \cup \{A' \otimes A''\}$. Hence $B \in kA'$ or $B \in kA''$. This contradicts Lemma 2.5.14, which says that $B \notin eA'$ and $B \notin eA''$. □

From Lemma 2.5.15 it immediately follows that $\ll$ is a partial order on the non-atomic subformula occurrences in π. We make crucial use of this fact in the following:

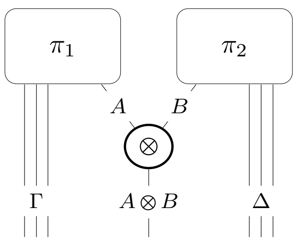
Proof of Lemma 2.5.6: Choose among the conclusions $A_1, \ldots, A_n$ (including the cuts) of π one which is maximal w.r.t. $\ll$. Without loss of generality, assume it is $A_i = A'_i \otimes A''_i$. We will now show that it is splitting, i.e., $\pi = \{A'_i \otimes A''_i\} \cup eA'_i \cup eA''_i$. By way of contradiction, assume $A'_i \otimes A''_i$ is not splitting. This means we have somewhere in π a formula occurrence B with immediate subformula B' such that (without loss of generality) $B' \in eA'_i$ and $B \notin eA'_i$. We also know that B must occur at or above some other conclusion, say $A_j = A'_j \otimes A''_j$. Hence $B \in kA_j$ and therefore $kB \subseteq kA_j$. But by Lemma 2.5.14 we have $A_i \in kB$ and therefore $A_i \in kA_j$, which contradicts the maximality of A_i w.r.t. $\ll$. □

Finally, we can prove Theorem 2.5.5.

First Proof of Theorem 2.5.5: Let us first show that the (in the sequent calculus) sequentializable pre-proof nets are indeed correct. This is done by verifying that the id-rule yields correct nets (which is obvious) and that all other inference rules preserve correctness. For the exch-rule this is obvious. Let us now consider the $\otimes$-rule. By way of contradiction, assume that

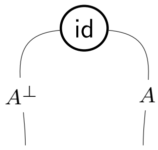


are correct, but

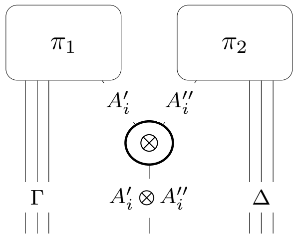


is not correct. This means there is a switching that is either disconnected or contains a cycle. Since a $\otimes$-node does not affect switchings, we conclude that the property of being disconnected or cyclic must hold for the same switching in one of π_1 or π_2. But this is a contradiction to the correctness of π_1 and π_2. For the the $\parr$-rule and the cut-rule we proceed similarly.

Conversely, let π be a correct pre-proof net. We proceed by induction on the size of π, i.e., the number n of $⅋$-, $\otimes$-, and cut-nodes in π, to construct a sequent calculus proof Π, that translates into π. If n is 0, then π must be of the shape



and we can apply the id-rule. Now let $n > 0$. If one of the conclusion formulas of π has a $⅋$-root, we can apply the $⅋$-rule and proceed by induction hypothesis. Now suppose all roots are $\otimes$ or cuts. Then we apply Lemma 2.5.6, which tells us, that there is one of them which splits the net. Assume, without loss of generality, that it is a $\otimes$-root, say $A_i = A_i' \otimes A_i''$. This means, the net is of the shape



and we can apply the $\otimes$-rule and proceed by induction hypothesis for π_1 and π_2. If the splitting root is a cut, we apply the cut-rule instead. $\square$

Let us now see the second proof. For this, we need the following lemma:

2.5.16 Lemma *Let π be a proof net with conclusion*

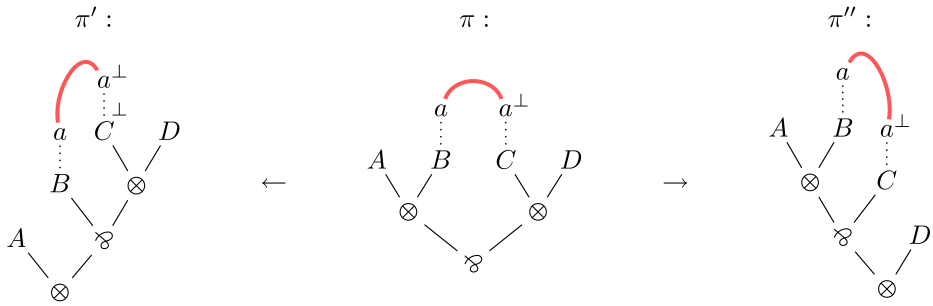
$$S\{(A \otimes B\{a\}) ⅋ (C\{a^\perp\} \otimes D)\} \quad ,$$

such that the a and the $a^\perp$ are paired up in the linking. Let π' and π'' be pre-proof nets with conclusions

$$S\{A \otimes (B\{a\} ⅋ (C\{a^\perp\} \otimes D))\} \qquad \text{and} \qquad S\{((A \otimes B\{a\}) ⅋ C\{a^\perp\}) \otimes D\}$$

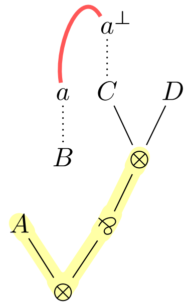
respectively, such that the linkings of π' and π'' (i.e., the pairing of dual atoms) are the same as the linking of π. Then at least one of π' and π'' is also correct.

Proof: Let us visualize the information we have about π, π', and π'' as follows:

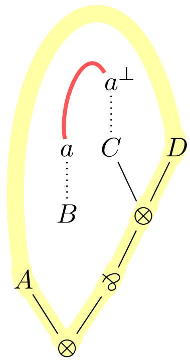


We proceed by way of contradiction, and assume that π is correct and that π' and π'' are both incorrect. If there is a switching s for π' (or π'') that is disconnected, then the same switching is also disconnected in π. Hence, we need to consider only the acyclicity condition. Suppose that there is a switching s' for π' that is cyclic. Then, in s' the $⅋$ below B must be switched to the right, and the cycle must pass through A, the root

$\otimes$ and the $\mathbin{\rotatebox[origin=c]{180}{\&}}$ as follows:

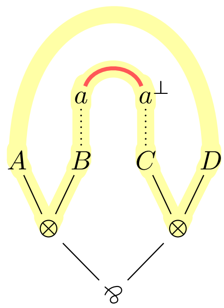


Otherwise we could construct a switching with the same cycle in π. If our cycle continues through D, i.e.,

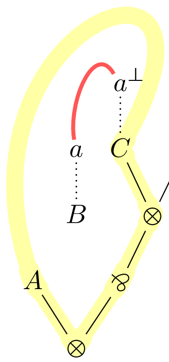


(19)

then we can use the path from A to D (that does not go through B or C, see Exercise 2.5.17) to construct a cyclic switching s in π as follows:

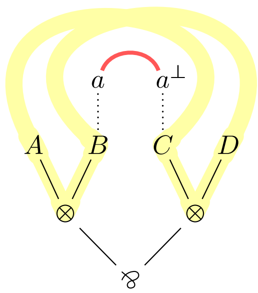


Hence, the cycle in s' goes through C, giving us a path from A to C, not passing through B (see Exercise 2.5.17):



(20)

By the same argumentation we get a switching s'' in π'' with a path from B to D, not going through C. From s' and s'', we can now construct a switching s for π with a cycle as follows:



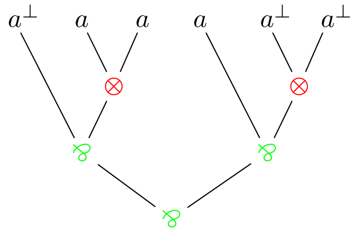
which contradicts the correctness of π. $\square$

2.5.17 Exercise Explain why we can in (19) assume that the cycle does not go through B or C, and in (20) not through B.

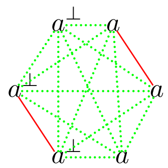
In our second proof of Theorem 2.5.5 we will also need the following concept:

2.5.18 Definition Let A be a formula. The *relation web* of A is the complete graph, whose vertices are the atom occurrences in A. An edge between two atom occurrences a and b is colored red, if the first common ancestor of them in the formula tree is a $\otimes$, and green if it is a $\parr$.

2.5.19 Example Consider the formula $(a^\perp \parr (a \otimes a)) \parr (a \parr (a^\perp \otimes a^\perp))$. Its formula tree is the following:



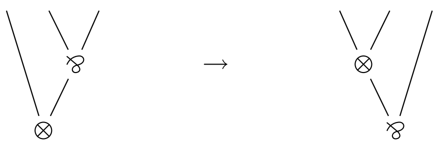
The relation web is therefore



where we use regular edges for red and dotted edges for green.

2.5.20 Definition The *degree of freedom* of a formula A, is the number of green edges in its relation web.

Second Proof of Theorem 2.5.5: Again, we start by showing that all rules preserve correctness. Here, the only interesting case is the switch rule (all others being trivial), which does the following transformation somewhere inside the net:


(21)

By way of contradiction, assume the net on the left is correct, and the one on the right is not. First, suppose there is a switching for the second net that is cyclic. If that cycle does not contain the $\otimes$-node shown on the right in (21), then this cycle is also present in the net on the left in (21). If our cycle contains the $\otimes$-node, then we can make the same cycle be present in the net on the left by switching the $\parr$-node to the left (i.e., removing the edge to the right). Now assume we have a disconnected switching for the net on the right. Then the same switching also disconnects the net on the left. Contradiction.

Conversely, assume we have a correct net π with conclusion F. For the time being, assume that π is cut-free. We proceed by induction on the degree of freedom of F. Pick inside F any pair of atoms that are linked together, say a and $a^\perp$. Then $F = S\{S_1\{a\} \parr S_2\{a^\perp\}\}$. Without loss of generality, we can assume that $S_1\{\ \}$ and $S_2\{\ \}$ are not $\parr$-contexts. We have the following cases:

- If $S_1\{\ \} = S_2\{\ \} = \{\ \}$, we can apply the rule $\mathsf{i}\!\downarrow$, and proceed by induction hypothesis.
- If $S_1\{\ \} \neq \{\ \}$ and $S_2\{\ \} = \{\ \}$, then $F = S\{(A \otimes B\{a\}) \parr a^\perp\}$ for some A and $B\{\ \}$. We can apply the switch rule to get $S\{A \otimes (B\{a\} \parr a^\perp)\}$, which is still correct (with the same linking as for F), but has smaller degree of freedom than F. The case where $S_1\{\ \} = \{\ \}$ and $S_2\{\ \} \neq \{\ \}$ is similar.
- If $S_1\{\ \} \neq \{\ \}$ and $S_2\{\ \} \neq \{\ \}$, then, without loss of generality, $F = S\{(A \otimes B\{a\}) \parr (C\{a^\perp\} \otimes D)\}$, for some A, $B\{\ \}$, $C\{\ \}$, D. By Lemma 2.5.16, we can apply the switch rule, since one of

  $$S\{A \otimes (B\{a\} \parr (C\{a^\perp\} \otimes D))\} \quad \text{and} \quad S\{((A \otimes B\{a\}) \parr C\{a^\perp\}) \otimes D\}$$

  is still correct. Since both of them have smaller degree of freedom than F, we can proceed by induction hypothesis.

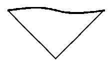
If π contains cuts, we can replace inside π all cuts with $\otimes$, to get a formula F' such that there is a derivation

$$\begin{array}{c} F' \\ \mathsf{i}\!\uparrow \Big\| \\ F \end{array}$$

Then π becomes a cut-free net with conclusion F', and we can proceed as above. $\square$

Note that the two different proofs of Theorem 2.5.5 yield a stronger version of Theorem 2.4.1.

2.5.21 Theorem *For every sequent calculus proof*



$$\vdash A_1, A_2, \dots, A_n$$

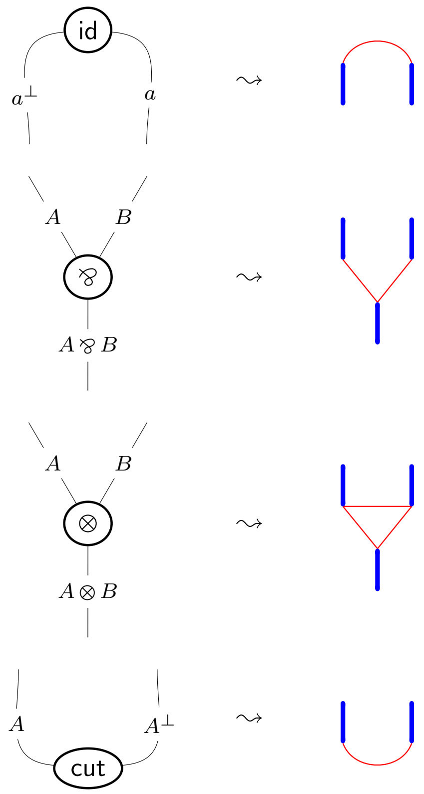
there is a proof in the calculus of structures

$$\begin{array}{c} \overline{} \\ \Big\| \\ A_1 \parr A_2 \parr \dots \parr A_n \end{array}$$

yielding the same proof net, and vice versa.

A geometric or graph-theoretic criterion like the one in Definition 2.5.4 and Theorem 2.5.5 is called a *correctness criterion.* The desired properties are soundness and completeness, as stated in Theorem 2.5.5. For MLL^-, the literature contains quite a lot of such criteria, and it would go far beyond the scope of this lecture notes to attempt to give a complete survey. But nonetheless, we will show here two other correctness criteria.

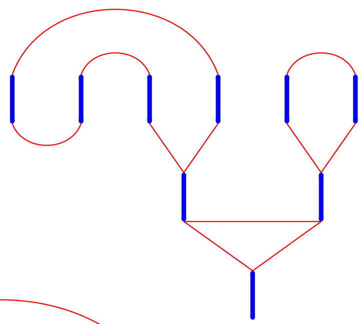
For the next one, we write the pre-proof nets in a different way:



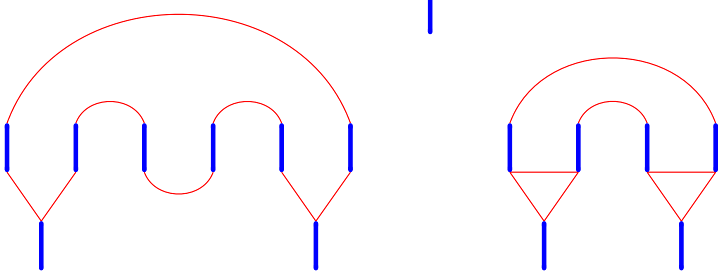
(22)

We call the resulting graphs *RB-graphs*. The R and B stand for Regular/Red and Bold/Blue. The main property of these graphs is that the blue/bold edges (in the following called *B-edges*) provide a bipartition of the set of vertices, i.e., every vertex in the RB-graph is connected to exactly one other vertex via a B-edge. The red/regular edges are in the following called *R-edges*.

Here are the examples from (15) and (16) written as RB-graphs:



(23)



(24)

2.5.22 Definition Let G be an RB-graph. An *Æ-path* in G is a path whose edges are alternating R- and B-edges, and that does not touch any vertex more than once. An *Æ-cycle* in G is a Æ-path from a vertex to itself, starting with a B-edge and ending with an R-edge.

The A and E stand for "alternating" and "elementary". The meaning of "alternating" should be clear, and the meaning of "elementary" is that the path or cycle must not cross itself.

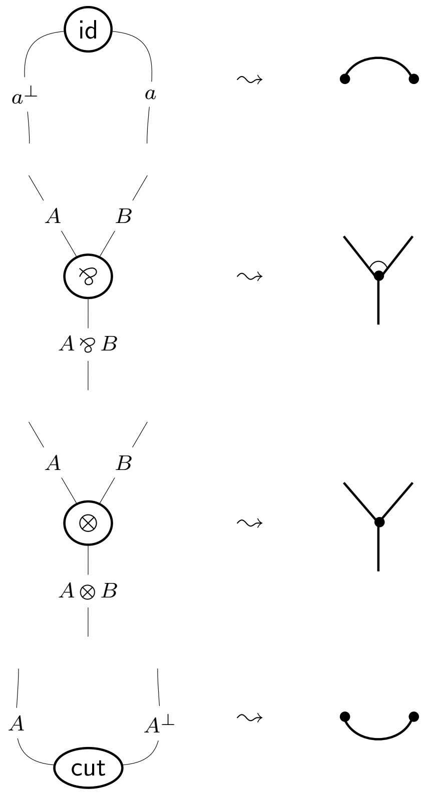
2.5.23 Definition A pre-proof net π *obeys the RB-criterion* (or shortly, is *RB-correct*) iff its RB-graph G_π contains no Æ-cycle and every pair of vertices in G_π is connected via an Æ-path.

2.5.24 Theorem *A pre-proof net is RB-correct if and only if it is a proof net.*

Proof: We show that a pre-proof net is RB-correct iff it obeys the switching criterion, which is easy: If there are two vertices in the RB-graph not connected by an Æ-path, then there is a switching yielding a disconnected graph, and vice versa. Similarly, the RB-graph contains an Æ-cycle if and only if we can provide a switching with a cycle. $\square$

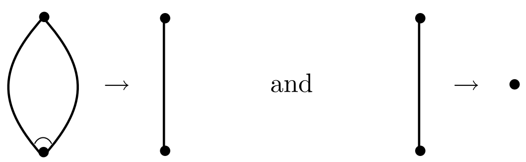
2.5.25 Exercise Work out the details of the previous proof.

For the third correctness criterion, we write our nets in yet another way:



(25)

Now consider the following two rewriting rules on these graphs:



(26)

It is important to note that in the first rule the two edges are between the same pair of vertices and are connected by an arc at exactly one of the two vertices. The second rule only applies if the two vertices on the lefthand side are distinct, and the edge is not connected to another edge by an arc.

2.5.26 Theorem *The reduction relation induced by the rules in* (26) *is terminating and confluent.*

Proof: Termination is obvious because at each step the size of the graph is reduced. Hence, it suffices to show local confluence to get confluence. But this is easy since there are no (proper) critical pairs. $\square$

This means that for each pre-proof net we get a uniquely defined reduced graph, and the question is now how this graph looks like.

2.5.27 Exercise Apply the reduction relation defined in (26) to the nets in (15) and (16).

2.5.28 Definition A pre-proof net *obeys the contraction criterion* if its normal form according to the reduction relation defined in (26) is

$$\bullet$$

i.e., a single vertex without edges.

At this point rather unsurprisingly, we get:

2.5.29 Theorem *A pre-proof net obeys the contraction criterion if and only if it is a proof net.*

Proof: As before, we show this by showing the equivalence of the switching criterion and the contraction criterion. This is easy to see since both reductions in (26) preserve and reflect correctness according to the switching criterion. □

Before we leave the topic of correctness criteria, let us make some important observations on their complexity. The naive implementation of checking the switching criterion needs exponential time: if there are n par-links in the net, then there are 2^n switchings to check. However, checking the RB-criterion needs only quadratic runtime. To verify this is an easy graph-theoretic exercise. It is also easy to see that checking the contraction criterion can be done in quadratic time. But it is rather surprising that it can be done in linear time in the size of the net.[5] This means that (in the case of MLL^-) when we go from a formal proof in a deductive system like the sequent calculus or the calculus of structures (whose correctness can trivially be checked in linear lime in the size of the proof) to the proof net, we do not lose any information. The proof net contains the *essence* of the proof, including the "deductive information". Unfortunately, MLL^- is (so far) the only logic (except some variants of it), for which this ideal of proof nets is reached. We come back to this in Sections 3 and 5.

2.6 Two-sided proof nets

The proof nets we have seen are also called *one-sided proof nets.* This implies that there is another kind: the so called *two-sided proof nets.* Their existence is justified by the fact, that systems in the sequent calculus can also come in a two-sided version. Here is a two-sided sequent calculus system for MLL^-:

$$\begin{array}{cc}
\mathsf{id}\,\dfrac{}{A \vdash A} & \mathsf{cut}\,\dfrac{\Gamma \vdash \Delta, A \quad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \\[2ex]
\mathsf{exchL}\,\dfrac{\Gamma, B, A, \Gamma' \vdash \Delta}{\Gamma, A, B, \Gamma' \vdash \Delta} & \mathsf{exchR}\,\dfrac{\Gamma \vdash \Delta, B, A, \Delta'}{\Gamma \vdash \Delta, A, B, \Delta'} \\[2ex]
\parr\mathsf{L}\,\dfrac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'} & \parr\mathsf{R}\,\dfrac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \parr B, \Delta} \\[2ex]
\otimes\mathsf{L}\,\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta} & \otimes\mathsf{R}\,\dfrac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} \\[2ex]
\multimap\mathsf{L}\,\dfrac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'} & \multimap\mathsf{R}\,\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta} \\[2ex]
.^{\perp}\mathsf{L}\,\dfrac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta} & .^{\perp}\mathsf{R}\,\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}
\end{array} \tag{27}$$

Now sequents are not just lists of formulas, but pairs of lists of formulas, and these pairs of lists are separated by a $\vdash$. One reason for using two-sided sequent systems is that one can treat negation as a proper connective (i.e., it is not pushed down to the atoms as in the one-sided version) and that one can have implication as primitive (i.e., there are rules for implication, instead of just defining $A \multimap B = A^{\perp} \parr B$). Of course, the disadvantage is that we heavily increase the number of rules. Here is an example of a two-sided derivation using the rules

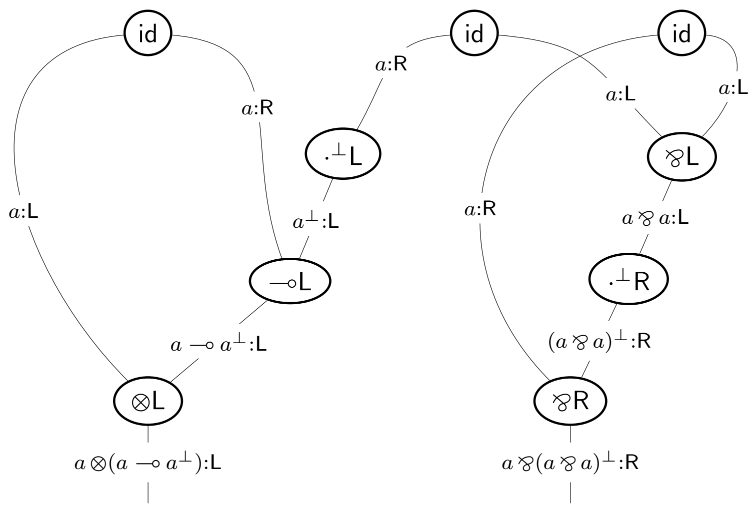
[5] For references, see Section 2.9.

in (27):

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{}
\LeftLabel{id}
\UnaryInfC{$a \vdash a$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$a \vdash a$}
\LeftLabel{$\cdot^{\perp}\mathsf{L}$}
\UnaryInfC{$a^{\perp}, a \vdash$}
\LeftLabel{$\multimap\mathsf{L}$}
\BinaryInfC{$a, a \multimap a^{\perp}, a \vdash$}
\AxiomC{}
\LeftLabel{id}
\UnaryInfC{$a \vdash a$}
\LeftLabel{$\parr\mathsf{L}$}
\BinaryInfC{$a, a \multimap a^{\perp}, a \parr a \vdash a$}
\LeftLabel{$\cdot^{\perp}\mathsf{R}$}
\UnaryInfC{$a, a \multimap a^{\perp} \vdash a, (a \parr a)^{\perp}$}
\LeftLabel{$\otimes\mathsf{L}$}
\UnaryInfC{$a \otimes (a \multimap a^{\perp}) \vdash a, (a \parr a)^{\perp}$}
\LeftLabel{$\parr\mathsf{R}$}
\UnaryInfC{$a \otimes (a \multimap a^{\perp}) \vdash a \parr (a \parr a)^{\perp}$}
\end{prooftree}
\tag{28}
$$

We omitted the instances of the exchange rule in this example.

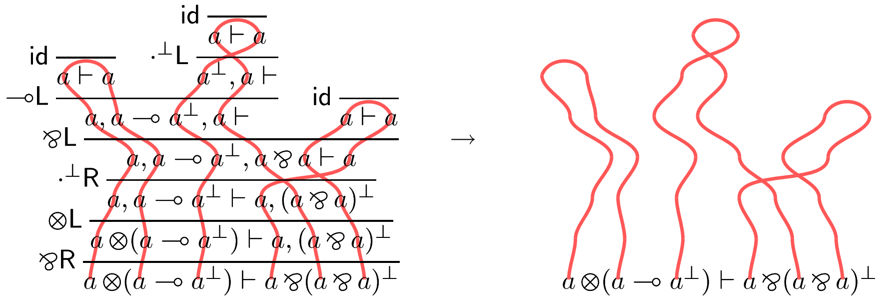
We can follow ideology 2.2.1 to translate two-sided sequent proofs into proof nets. But we have to assign to each rule and each formula appearing in the net a label which indicated whether the rule/formula comes from the left-hand side or the right-hand side of the sequent. We use here the letters L and R. Figure 9 shows how the sequent rules in (27) are translated into proof nets. We omitted the rules for exch (compare with Figure 1) and $\cdot^{\perp}$. Here is the result of translating (28):



(29)

Note that the nets in (7) and (29) are almost identical from the graph-theoretical viewpoint. The only differences are that the labels are different and that there are additional $\cdot^{\perp}\mathsf{L}$ and $\cdot^{\perp}\mathsf{R}$ nodes in (29).

Of course, we can also use the flow-graph method to obtain a proof net from (28):



(30)

However, observe that from the flow-graph we do not get *a priori* the information what is left and what is right. This means, for a proper definition of pre-proof net, we need to find another way to determine when we may allow an identity link between a pair of atoms. To do so, we use *polarities*: we assign to each subformula appearing in the sequent a unique polarity, that is, an element of the set $\{\bullet, \circ\}$, where $\bullet$ can be read as left/negative/up and $\circ$ as right/positive/down. Let $\Gamma \vdash \Delta$ be given. Then

(i) all formulas in Γ get polarity $\bullet$, and all formulas in Δ get polarity $\circ$;

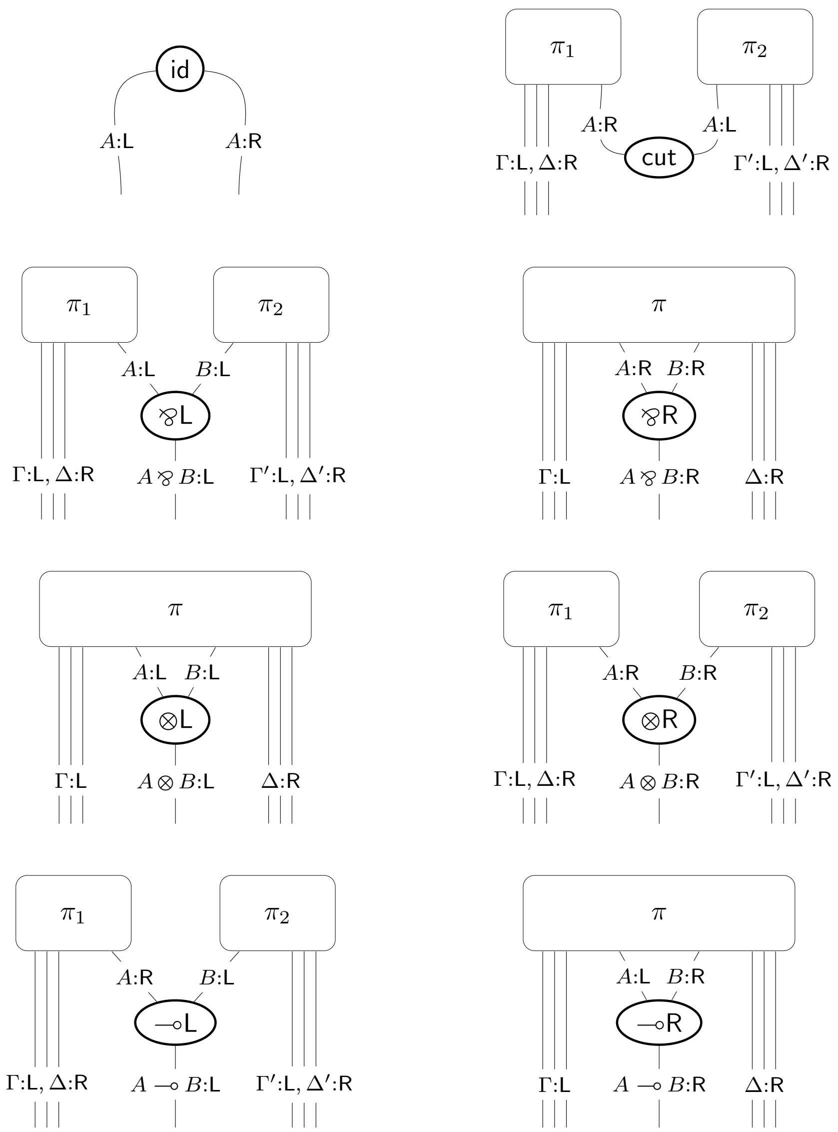


Figure 9: From two-sided sequent calculus to proof nets (sequent calculus rule driven)

(ii) whenever $A \otimes B$ has polarity $\bullet$ (resp. $\circ$), then A and B also get polarity $\bullet$ (resp. $\circ$);

(iii) whenever $A \parr B$ has polarity $\bullet$ (resp. $\circ$), then A and B also get polarity $\bullet$ (resp. $\circ$);

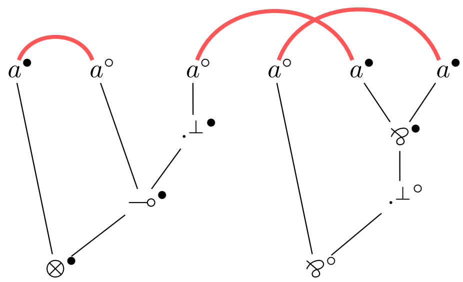
(iv) whenever $A \multimap B$ has polarity $\bullet$ (resp. $\circ$), then A gets polarity $\circ$ (resp. $\bullet$) and B gets polarity $\bullet$ (resp. $\circ$);

(v) whenever $A^\perp$ has polarity $\bullet$ (resp. $\circ$), then A gets polarity $\circ$ (resp. $\bullet$).

Note that only for negation and implication there is a change of polarity. We can now give the following definition:

2.6.1 Definition A *two-sided pre-proof net* consists of a list of formula trees, which is correctly polarized according to (ii)-(v) above, together with a perfect matching of the set of leaves, such that two leaves are matched only if they are labeled by the same atom and have different polarity. If there are cuts in the net, then the two cut formulas have to be identical and must have different polarity.

As an example we give here the net corresponding to (30):

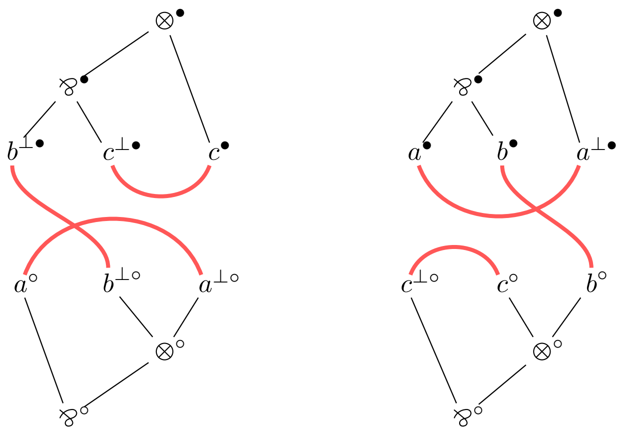


(31)

where we put the polarity as superscript to nodes in the trees. It should not come at a surprise that we get (again, up to some trivial change in the notation) the same proof nets as with the first method.

The correctness criteria are exactly the same as we discussed them for one-sided proof nets in Section 2.5. The only thing to note is that the $\otimes^\bullet$, $\parr^\circ$, and $\multimap^\circ$ behave as the $\parr$ in Section 2.5, i.e., remove one edge to a child to get a switching, and the $\otimes^\circ$, $\parr^\bullet$, and $\multimap^\bullet$ behave as the $\otimes$ in Section 2.5. Clearly, the example in (31) is correct.

Let us now for the time being forget the $\multimap$ and $\cdot^\perp$ nodes that flip the polarity. Then we can draw the two-sided proof net in a two-sided way. Things with polarity $\bullet$ are drawn on the top and things with polarity $\circ$ are drawn at the bottom. In between we put the identity links. Since we now have again negated atoms, we have to change the condition on the identity links: *Two atoms occurrences may be linked, if they are the same and have different polarity, or if they are dual to each other and have the same polarity.* Here are two examples:



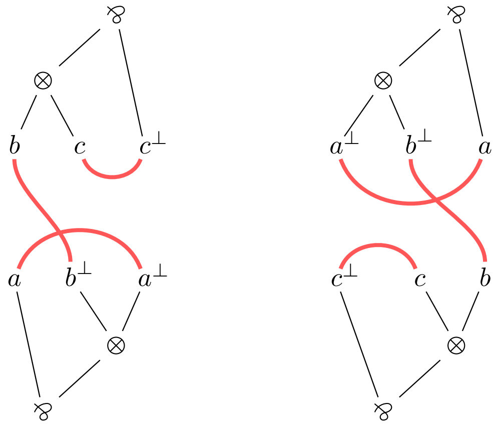
(32)

Note that these are exactly the nets that are obtained via the flow-graph method from the calculus of structures derivations in (14):

$$\begin{array}{r|c}
 & (b^\perp \parr c^\perp) \otimes c \\ \hline
\mathsf{s} & b^\perp \parr (c^\perp \otimes c) \\ \hline
\mathsf{i}\!\uparrow & b^\perp \\ \hline
\mathsf{i}\!\downarrow & b^\perp \otimes (a \parr a^\perp) \\ \hline
\mathsf{s} & a \parr (b^\perp \otimes a^\perp)
\end{array}
\qquad\qquad
\begin{array}{r|c}
 & (a \parr b) \otimes a^\perp \\ \hline
\mathsf{s} & (a \otimes a^\perp) \parr b \\ \hline
\mathsf{i}\!\uparrow & b \\ \hline
\mathsf{i}\!\downarrow & (c^\perp \parr c) \otimes b \\ \hline
\mathsf{s} & c^\perp \parr (c \otimes b)
\end{array}
\qquad (33)$$

This is the second justification for the existence of two-sided proof nets. They are directly obtained via flow-graphs from derivations in the calculus of structures. As before, the correctness criterion does not change.

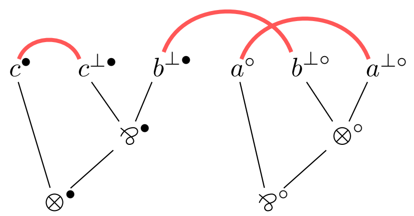
Note that from the graph-theoretical point of view, there is now not much difference between $\otimes^\circ$ and $\parr^\bullet$, between $\otimes^\bullet$ and $\parr^\circ$, between a° and $a^{\perp\bullet}$, and between $a^\bullet$ and $a^{\perp\circ}$ (at least for MLL^-). For this reason, the nets in (32) could also be written as



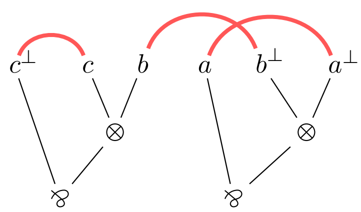
(34)

Now they are in fact the same net just turned upside down. This is no surprise since the two derivations in (33) are dual to each other.

Let us emphasize that it does not matter in which way the graph is drawn on the paper. The net on the left in (32,34) could also be drawn as



or



(35)

In the case of MLL^- it makes no difference whether one prefers to put a proof net in a "one-sided" (35) or "two-sided" (32,34) way on the paper, and whether one prefers to use the labels $\{\bullet, \circ\}$ for left/right (or up/down) as in (32) or to live without them as in (34).

This rather trivial observation can be nailed down in the following theorem which strengthens Theorems 2.4.1 and 2.5.21.

2.6.2 Theorem *Let $n, m \leq 1$ and $A_1, \ldots, A_n, B_1, \ldots, B_m$ be any MLL^- formulas. Then the following are equivalent:*

(i) *There is a sequent calculus proof*

$$\vdash A_1^\perp, \ldots, A_n^\perp, B_1, \ldots, B_m \tag{36}$$

(ii) *There is a calculus of structures proof*

$$\begin{array}{c} - \\ \Vert \\ A_1^\perp \parr \cdots \parr A_n^\perp \parr B_1 \parr \ldots \parr B_m \end{array} \tag{37}$$

(iii) *There is a (two-sided) sequent calculus proof*

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_m \tag{38}$$

(iv) *There is a calculus of structures proof*

$$\begin{array}{c} A_1 \otimes \cdots \otimes A_n \\ \Vert \\ B_1 \parr \ldots \parr B_m \end{array} \tag{39}$$

Furthermore all of (36)–(39) *can be constructed such that they all yield the same proof net.*

This section can be summarized a follows: Two-sided proof nets are a special kind of one-sided proof nets, and one-sided proof nets are a special kind of two-sided proof nets. Whether one prefers the one-sided or the two-sided version is a matter of personal taste and also a matter of economics in notation.

This observation is *not* restricted to the special case of MLL^-. Every theory of two-sided proof nets can trivially transformed into a theory of one-sided proof nets, and vice versa. By using polarities as described above, the one-sided version can usually be presented in a more compact way than the two-sided version. For this reason, we will in the following stay in the one-sided world.

2.7 Cut elimination

Let us now briefly discuss an issue that usually excites proof theorists: In a well-designed deduction system, every formula/sequent that is derivable, can also be derived without using the cut-rule. Proving this fact is usually a highly nontrivial task and involves (depending on the logic in question) very sophisticated proof theory.

But here we do not have time and space to go into the details of cut elimination and its important consequences. We discuss here only a very simple logic, namely MLL^-, for which cut elimination is easy. For the sequent calculus it is stated as follows:

2.7.1 Theorem *For every proof*

$$\begin{array}{c}\Pi\\ \vdash \Gamma\end{array}$$

using the rules in (1) *there is a proof*

$$\begin{array}{c}\Pi'\\ \vdash \Gamma\end{array}$$

of the same conclusion, that does not use the cut*-rule.*

Proof (Sketch): We do only sketch the direct proof within the sequent calculus, because, even in the simple case of MLL^-, carrying out all the details is quite tiresome and boring. But the basic idea is simple: The cut rule is permuted up in the proof until it disappears. Most cases are trivial rule permutations similar to the one in (10). But there are two so called *key cases* which are not trivial. Here is the first one:

$$\mathsf{cut}\,\dfrac{\parr\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A, B, \Delta\end{array}}{\vdash \Gamma, A \parr B, \Delta} \qquad \otimes\,\dfrac{\begin{array}{c}\Pi_2\\ \vdash \Gamma', B^\perp\end{array} \quad \begin{array}{c}\Pi_3\\ \vdash A^\perp, \Delta'\end{array}}{\vdash \Gamma', B^\perp \otimes A^\perp, \Delta'}}{\vdash \Gamma, \Delta, \Gamma', \Delta'} \quad \rightsquigarrow \quad \mathsf{cut}\,\dfrac{\mathsf{cut}\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A, B, \Delta\end{array} \quad \begin{array}{c}\Pi_2\\ \vdash \Gamma', B^\perp\end{array}}{\vdash \Gamma, A, \Delta, \Gamma'} \qquad \begin{array}{c}\Pi_3\\ \vdash A^\perp, \Delta'\end{array}}{\vdash \Gamma, \Delta, \Gamma', \Delta'}$$

where some instances of the exch-rule have been omitted. The second key case lets the cut disappear when it meets an identity rule at the top of the proof:

$$\mathsf{cut}\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash A^\perp, A} \qquad \begin{array}{c}\Pi_1\\ \vdash A^\perp, \Gamma\end{array}}{\vdash A^\perp, \Gamma} \quad \rightsquigarrow \quad \begin{array}{c}\Pi_1\\ \vdash A^\perp, \Gamma\end{array}$$

The difficult part of a cut elimination proof is usually to show termination of this "permuting the cut up" business, which is done by coming up with a clever induction measure. We leave that to the reader. □

2.7.2 Exercise Complete the proof of Theorem 2.7.1. The following things still have to be done:

1. Find a correct way of dealing with the exch-rule. (Hint: Design a "super-cut-rule", that has exchange built in.)
2. Show the termination of the process of permuting up the cut, i.e., find the right induction measure.
3. Show that after the termination, the resulting proof is indeed cut-free.

If you have seen a cut elimination argument before, then this exercise should not be too hard for you. If you have never before seen some cut elimination, you will learn a lot about it by doing this exercise. In any case, this (rather painful) exercise will help you to admire the beauty of proof nets.

Let us now turn to cut elimination in the calculus of structures, where it means that not only the cut (i.e., the rule $\mathsf{i}\!\uparrow$), but the whole up-fragment (i.e., all rules with the $\uparrow$ in the name) are not needed for provability.

2.7.3 Theorem *If there is a proof*

$$\{\mathsf{i}\!\downarrow, \mathsf{i}\!\uparrow, \sigma\!\downarrow, \sigma\!\uparrow, \alpha\!\downarrow, \alpha\!\uparrow, \mathsf{s}\} \left\| \begin{array}{c} \overline{} \\ \\ A \end{array} \right. \Pi$$

using all rules in Figure 5, then there is a proof

$$\{\mathsf{i}\!\downarrow, \sigma\!\downarrow, \alpha\!\downarrow, \mathsf{s}\} \left\| \begin{array}{c} \overline{} \\ \\ A \end{array} \right. \Pi$$

that has the same conclusion A and that does not use any up-rule.

Proof (Sketch): The easy trick of permuting the cut up does not work in the presence of deep inference. For this reason completely different techniques have to be used to eliminate the up-fragment. But as in the sequent calculus, carrying out all the details is technical and boring. For this reason we will give here only a sketch. The first observation to make is that every up-rule

$$\mathsf{r}\!\uparrow \frac{S\{B^\perp\}}{S\{A^\perp\}}$$

can be replaced by a derivation using only $\mathsf{i}\!\downarrow, \mathsf{i}\!\uparrow, \mathsf{s}, \mathsf{r}\!\downarrow$:

$$\mathsf{i}\!\uparrow \dfrac{\mathsf{s}\dfrac{\mathsf{r}\!\downarrow \dfrac{\mathsf{i}\!\downarrow \dfrac{S\{B^\perp\}}{S\{(A^\perp \parr A) \otimes B^\perp\}}}{S\{(A^\perp \parr B) \otimes B^\perp\}}}{S\{A^\perp \parr (B \otimes B^\perp)\}}}{S\{A^\perp\}} \tag{40}$$

This means that only the rule $\mathsf{i}\!\uparrow$ needs to be eliminated. The second observation is, that this rule can be reduced to its atomic version:

$$\mathsf{i}\!\uparrow \frac{A \otimes A^\perp}{} \quad \rightsquigarrow \quad \mathsf{ai}\!\uparrow \frac{a \otimes a^\perp}{}$$

$$\mathsf{i}\!\uparrow \frac{S\{B \parr (A \otimes A^\perp)\}}{S\{B\}} \quad \rightsquigarrow \quad \mathsf{ai}\!\uparrow \frac{S\{B \parr (a \otimes a^\perp)\}}{S\{B\}}$$

This[6] is done by systematically replacing

$$\mathsf{i}\!\uparrow \frac{S\{B \parr ((A \parr C) \otimes (C^\perp \otimes A^\perp))\}}{S\{B\}} \quad \text{by} \quad \mathsf{i}\!\uparrow \dfrac{\mathsf{i}\!\uparrow \dfrac{\mathsf{s}\dfrac{\alpha\!\uparrow \dfrac{S\{B \parr ((A \parr C) \otimes (C^\perp \otimes A^\perp))\}}{S\{B \parr (((A \parr C) \otimes C^\perp) \otimes A^\perp)\}}}{S\{B \parr ((A \parr (C \otimes C^\perp)) \otimes A^\perp)\}}}{S\{B \parr (A \otimes A^\perp)\}}}{S\{B\}} \tag{41}$$

Now the rules to eliminate are $\mathsf{ai}\!\uparrow$, $\alpha\!\uparrow$, and $\sigma\!\uparrow$ (see Exercise 2.7.4).[7] For this, we use Guglielmi's powerful *splitting lemma*, which says that whenever there is a proof

$$\{\mathsf{i}\!\downarrow, \sigma\!\downarrow, \alpha\!\downarrow, \mathsf{s}\} \left\| \begin{array}{c} \overline{} \\ \\ S\{C \parr (A \otimes B)\} \end{array} \right. \Pi \tag{42}$$

[6] By duality, we can do the same trick to the rule $\mathsf{i}\!\downarrow$. Compare with what we did in (11) to the id-rule in the sequent calculus.

[7] It might seem silly to first explain that only the rule $\mathsf{i}\!\uparrow$ needs to be eliminated, and then say that also the other up-rules need to be taken care of. The reason is that with the trick in (40) *all* up-rules can be removed, but by reducing the rule $\mathsf{i}\!\uparrow$ to it atomic version, only *some* of them are reintroduced. This is crucial for more sophisticated logics than MLL^-.

then there are formulas K_A and K_B, such that for every formula C, we have derivations

$$\begin{array}{c} K_A \parr K_B \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_1 \\ S\{C\} \end{array} \quad , \quad \begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_2 \\ K_A \parr A \end{array} \quad , \quad \begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_3 \\ K_B \parr B \end{array} \tag{43}$$

Additionally, we get a derivation

$$\begin{array}{c} K_A \parr K_B \parr X \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \tilde{\Pi}_1 \\ S\{C \parr X\} \end{array} \tag{44}$$

for every formula X. The crucial point of this lemma is that it speaks only about the down-fragment. Furthermore we need a variant of the splitting lemma, which says that whenever we have

$$\begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi \\ C \parr a \end{array} \quad , \quad \text{then we also have} \quad \begin{array}{c} a^\perp \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi' \\ C \end{array} \quad .$$

Now we can remove the rule $\mathsf{ai}\uparrow$ starting with the topmost instance as follows. What we have is

$$\begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi \\ \mathsf{ai}\uparrow \dfrac{S\{B \parr (a \otimes a^\perp)\}}{S\{B\}} \end{array}$$

Via the splitting lemma, we can get

$$\begin{array}{c} K_a \parr K_{a^\perp} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_1 \\ S\{B\} \end{array} \quad , \quad \begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_2 \\ K_a \parr a \end{array} \quad , \quad \begin{array}{c} \overline{} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_3 \\ K_{a^\perp} \parr a^\perp \end{array}$$

Now we can get from Π_2 and Π_3 the following:

$$\begin{array}{c} a^\perp \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi'_2 \\ K_a \end{array} \qquad \text{and} \qquad \begin{array}{c} a \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi'_3 \\ K_{a^\perp} \end{array}$$

now we can build:

$$\begin{array}{c} \mathsf{ai}\downarrow \dfrac{}{a^\perp \parr a} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi'_2 \parr \Pi'_3 \\ K_a \parr K_{a^\perp} \\ \{\mathsf{i}\downarrow,\sigma\downarrow,\alpha\downarrow,\mathsf{s}\} \Big\Vert\, \Pi_1 \\ S\{B\} \end{array}$$

which gives us a proof for $S\{B\}$ that does not use any up-rules. □

2.7.4 Exercise Explain, why not only the rules $\mathsf{ai}\uparrow$ and $\alpha\uparrow$, but also the rule $\sigma\uparrow$ needs to be eliminated after doing (41).

2.7.5 Exercise Complete the proof of Theorem 2.7.3. Things that remain to be done:

1. Show how the rules $\alpha\uparrow$ and $\sigma\uparrow$ can be eliminated in a similar way as $\mathsf{ai}\uparrow$. Hint: Use $\tilde{\Pi}_1$ instead of Π_1. For $\sigma\uparrow$ plug in $B \otimes A$ for X. For $\alpha\uparrow$, you need to apply splitting twice.
2. Prove the splitting lemma. This is the really hard part. Hint: First show the lemma for $S\{\ \} = \{\ \}$ (shallow splitting). To do so, proceed by induction on the length of Π and the size of $C \parr (A \otimes B)$. Then show the lemma for arbitrary context $S\{\ \}$. For this, proceed by induction on $S\{\ \}$.

3. Show that the rule

$$\mathsf{i}{\uparrow}\,\frac{A \otimes A^{\perp}}{\quad}$$

is not needed for provability, i.e., it is impossible to prove "nothing". Hint: You need a variant of splitting, saying, if there is a proof

$$\begin{array}{r}\{\mathsf{i}{\downarrow}, \sigma{\downarrow}, \alpha{\downarrow}, \mathsf{s}\}\ \overline{\Big\Vert}\ \Pi \\ A \otimes B\end{array}$$

then there are proofs

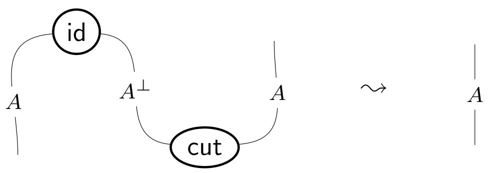
$$\begin{array}{r}\{\mathsf{i}{\downarrow}, \sigma{\downarrow}, \alpha{\downarrow}, \mathsf{s}\}\ \overline{\Big\Vert}\ \Pi_2 \\ A\end{array} \qquad \text{and} \qquad \begin{array}{r}\{\mathsf{i}{\downarrow}, \sigma{\downarrow}, \alpha{\downarrow}, \mathsf{s}\}\ \overline{\Big\Vert}\ \Pi_3 \\ B\end{array}$$

i.e., $S\{\ \}$, C, K_A, and K_B are all "empty".

Proving cut elimination via splitting in the calculus of structures is in the unit-free case a little more messy than in the case with units because many cases need to be considered separately that could be treated together in the presence of units.

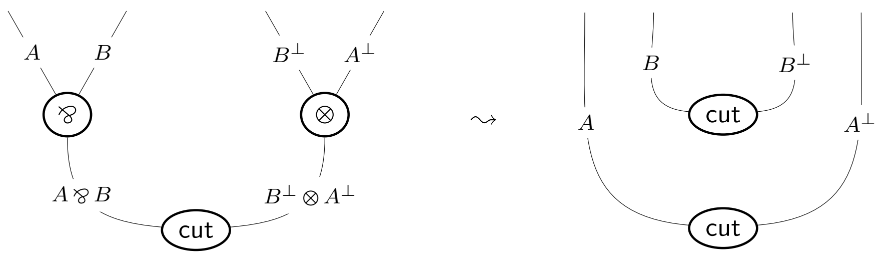
Let us now see how proof nets deal with the problem of cut elimination. Of course, the main point to make here is that cut elimination will become considerably simpler:

Consider the following reduction rules on pre-proof nets with cuts:



(45)

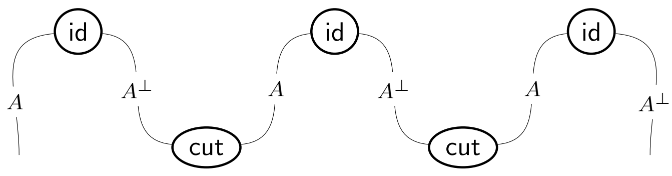
and



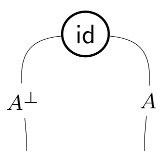
(46)

2.7.6 Theorem *The cut reduction relation defined by* (45) *and* (46) *terminates and is confluent.*

Proof: Showing termination is trivial because in every reduction step the size of the net decreases. For showing confluence, note that the only possibility for making a critical pair is when two cuts want to reduce with the same identity link. Then the situation must be of the shape:

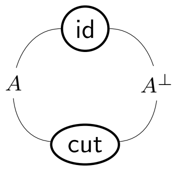


But no matter in which order and with which identity we reduce the cuts, the final result will always be



Hence we also have confluence. $\square$

However, it could happen, that we end up in a situation like



where we cannot reduce any further. That something like this cannot happen if we start out with a correct net is ensured by the following theorem, which says that the cut reduction preserves correctness.

2.7.7 Theorem *Let π and π' be pre-proof nets such that π reduces to π' via the reductions* (45) *and* (46)*. If π is correct, then so is π'.*

Proof: For proving this, let us use the RB-correctness criterion. Written in terms of RB-graphs, the two reduction rules look as follows:

 (47)

and

 (48)

That the first rule preserves RB-correctness is obvious because it just shortens an existing path. For the second rule, we proceed by way of contradiction. First, assume that the graph on the right contains an Æ-cycle, while the one on the left does not. There are three possibilities:

1. The Æ-cycle does not contain one of the new B-R-B-paths. Then the same cycle is also present on the left. Contradiction.

2. The Æ-cycle contains exactly one of the new B-R-B-paths. Then, as before, the same cycle is also present on the left. Contradiction.

3. The Æ-cycle contains both of the new B-R-B-paths. Then we can construct an Æ-cycle on the left that comes in at the upper left corner, goes down through the $\otimes$-link, and goes out at the lower left corner. Again, we get a contradiction.

That Æ-path connectedness is preserved is shown in a similar way. □

2.7.8 Exercise Complete the proof of Theorem 2.7.7, i.e., show that if we apply (48) to an RB-correct net, then in the result every pair of vertices is connected by an Æ-path. Hint 1: Note that the two rightmost vertices in (48) must be connected by an Æ-path that does not touch the new B-R-B-paths (why?). Hint 2: You will need the fact that the first net is also Æ-cycle free.

The important point of Theorem 2.7.7 is that it allows us to give short proofs of Theorems 2.7.1 and 2.7.3: Let Π be a proof with cuts in MLL^- given in the sequent calculus or the calculus of structures. We can translate Π into a proof net π, as described in Sections 2.2–2.4 and remove the cuts from the proof net as described above. This gives us a proof net π', which we can translate back to the sequent calculus or the calculus of structures. This works because removing the cuts from the proof net preserves the property of being correct (i.e., being a proof net), and translating back does not introduce any new cuts.

This raises an important question: Suppose we start out with a proof Π with cuts in MLL^- (given in sequent calculus or the calculus of structures). Now we could first remove the cuts as sketched out in the proofs of Theorems 2.7.1 and 2.7.3, and then translate the resulting cut-free proof Π' into a proof net π'_1. Alternatively, we could first translate Π into a proof net π, and then remove the cuts from π, to obtain the cut-free proof net π'_2. Do we get the same result? Is $\pi'_1 = \pi'_2$?

The answer is of course **yes**. To see this, note that the cut reduction steps in the sequent calculus either preserve the proof net (if the cut is just permuted up via a trivial rule permutation) or do exactly the same as the cut reduction steps for proof nets.

The same is true for the calculus of structures. The proof of the splitting lemma is designed such that it preserves the net. To make this formally precise would go beyond the scope of these lecture notes, but by comparing Figures 7 and 8 the reader should get an idea.

We can summarize this by the following commuting diagram:

$$\begin{array}{ccc}
\begin{array}{c}\text{proof with cuts}\\ \text{(in SC or CoS)}\end{array} & \longrightarrow & \text{proof net with cuts} \\
\begin{array}{c}\text{cut elimination}\\ \text{(in SC or CoS)}\end{array} \downarrow & & \downarrow \begin{array}{c}\text{cut elimination}\\ \text{(in proof nets)}\end{array} \\
\begin{array}{c}\text{cut-free proof}\\ \text{(in SC or CoS)}\end{array} & \longrightarrow & \text{cut-free proof net}
\end{array} \qquad (49)$$

Our basic introduction into the theory of proof nets for MLL^- is now finished. However, a very important and fundamental question has not yet been mentioned:

2.7.9 Big Question Let π and π' be two proof nets such that π' is obtained from π by applying some cut reduction steps. Do π and π' represent the *same* proof?

By comparing (7) and (9), one might be tempted to say no. But by looking at Figures 3 and 7, one is tempted to say yes. Furthermore, from the viewpoint of proof nets it makes no difference, whether we eliminate the cut from

$$\mathsf{cut}\,\frac{\overset{\Pi_1}{\vdash A^\perp, B} \qquad \overset{\Pi_2}{\vdash B^\perp, C}}{\vdash A^\perp, C}$$

in the sequent calculus or whether we perform the the composition

$$\begin{array}{c} A \\ \Pi_1 \Big\| \\ B \\ \Pi_2 \Big\| \\ C \end{array} \quad \to \quad \begin{array}{c} A \\ \Big\| \\ C \end{array}$$

in the calculus of structures (no matter whether we use one-sided or two-sided proof nets, cf. Section 2.6).

In the following section we are going to give another justification for the "yes", which is independent from proof nets, sequent calculus, calculus of structures, or any other way of presenting proofs.

2.8 *-Autonomous categories (without units)

In this section we will introduce the concept of *-autonomous categories. We do not presuppose any knowledge of category theory. We introduce what we need on the way along. It is not much anyway. The basic idea is to give an abstract algebraic theory of proofs, which is based on the following postulates about proofs:

(i) for every proof f of conclusion B from hypothesis A (denoted by $f\colon A \to B$) and every proof g of conclusion C from hypothesis B (denoted by $g\colon B \to C$) there is a uniquely defined composite proof $g \circ f$ of conclusion C from hypothesis A (denoted by $g \circ f\colon A \to C$),

(ii) this composition of proofs is associative,

(iii) for each formula A there is an identity proof $1_A\colon A \to A$ such that for $f\colon A \to B$ we have $f \circ 1_A = f = 1_B \circ f$, i.e, it behaves as identity w.r.t. composition.

These axioms say no more and no less than that the proofs are the arrows in a category whose objects are the formulas. Let us now add more axioms that are specific to logic and do not hold in general in categories:

(iv) Whenever we have a formula A and formula B, then $A \otimes B$ is another formula. For two proofs $f\colon A \to C$ and $g\colon B \to D$ we have a uniquely defined proof $f \otimes g\colon A \otimes B \to C \otimes D$, such that for all $h\colon C \to E$ and $k\colon D \to F$, we have

$$(h \otimes k) \circ (f \otimes g) = (h \circ f) \otimes (h \circ g)\colon A \otimes B \to E \otimes F \quad . \qquad (50)$$

Using category theoretical language, Axiom (iv) just says that $\otimes$ is a bifunctor. What does this mean? Consider the following two derivations (using the notation from the calculus of structures):

$$\begin{array}{rc} & A\otimes B \\ f\otimes B & \Vert \\ & C\otimes B \\ C\otimes g & \Vert \\ & C\otimes D \end{array} \qquad \text{and} \qquad \begin{array}{rc} & A\otimes B \\ A\otimes g & \Vert \\ & A\otimes D \\ f\otimes D & \Vert \\ & C\otimes D \end{array} \tag{51}$$

In the left one we use first f to go from A to C, and do nothing to B,[8] and then use g to go from B to D (and do nothing to C). In the right derivation, we first use g to go from B to D, and then f to go from A to C. Equation (50) says that the two derivations with premise $A\otimes B$ and conclusion $C\otimes D$ in (51) represent the same proof, denoted by $f\otimes g$. Mathematicians came up with a very clever way of writing an equation between objects as in (51), namely, via *commuting diagrams*. Instead of writing the two derivations in (51) and saying they are equal, we write:

$$\text{The diagram}\quad \begin{tikzcd} A\otimes B \arrow[r, "A\otimes g"] \arrow[d, "f\otimes B"'] & A\otimes D \arrow[d, "f\otimes D"] \\ C\otimes B \arrow[r, "C\otimes g"'] & C\otimes D \end{tikzcd} \quad\text{commutes}\ .$$

From the proof theoretical viewpoint, this equation is indeed wanted. The difference between the two derivations in (51) is an artefact of syntactic bureaucracy. The kind of bureaucracy in exhibited in (51) is called *bureaucracy of type A*. This implies that there must also be a *bureaucracy of type B*. Consider the following two derivations:

$$\begin{array}{rc} & A\otimes(B\parr C) \\ f\otimes(B\parr C) & \Vert \\ \mathsf{s} & \dfrac{A'\otimes(B\parr C)}{(A'\otimes B)\parr C} \end{array} \qquad \text{and} \qquad \begin{array}{rc} \mathsf{s} & \dfrac{A\otimes(B\parr C)}{(A\otimes B)\parr C} \\ (f\otimes B)\parr C & \Vert \\ & (A'\otimes B)\parr C \end{array} \tag{52}$$

In the left one we first use the proof f, taking us from A to A' (and doing nothing to B and C), and then we apply the switch rule. In the derivation on the right we first apply the switch rule, and then do f. Clearly the two are essentially the same and should be identified eventually. Let us write this as commuting diagram:

$$\begin{tikzcd} A\otimes(B\parr C) \arrow[r, "\mathsf{s}_{A,B,C}"] \arrow[d, "f\otimes(B\parr C)"'] & (A\otimes B)\parr C \arrow[d, "(f\otimes B)\parr C"] \\ A'\otimes(B\parr C) \arrow[r, "\mathsf{s}_{A',B,C}"'] & (A'\otimes B)\parr C \end{tikzcd} \tag{53}$$

Using category theoretical language, equation (53) says precisely that the morphism $\mathsf{s}_{A,B,C}\colon A\otimes(B\parr C)\to(A\otimes B)\parr C$ is *natural in A*. Of course, in the end, we should have that switch is natural in all three arguments.

Before we can continue with our list of axioms, we need another category theoretical concept. Suppose we have two formulas A and B and proofs $f\colon A\to B$ and $g\colon B\to A$. If we have for some reason that $f\circ g = 1_B$ and $g\circ f = 1_A$, then we say that A and B are *isomorphic*. In this case f and g are *isomorphisms*. The following axiom shows two examples:

(v) For all formulas A, B, and C, we postulate the existence of proofs

$$\begin{aligned} \alpha_{A,B,C}&\colon A\otimes(B\otimes C)\to(A\otimes B)\otimes C \\ \sigma_{A,B}&\colon A\otimes B\to B\otimes A \end{aligned} \tag{54}$$

[8] More precisely, it is the identity 1_B taking us from B to B.

which are isomorphisms, and which are natural in all arguments,[9] and which obey the following equations:

$$
\begin{tikzcd}
A \otimes (B \otimes (C \otimes D)) \arrow[rr, "A \otimes \alpha_{B,C,D}"] \arrow[d, "\alpha_{A,B,C \otimes D}"'] & & A \otimes ((B \otimes C) \otimes D) \arrow[d, "\alpha_{A,B \otimes C,D}"] \\
(A \otimes B) \otimes (C \otimes D) \arrow[dr, "\alpha_{A \otimes B,C,D}"'] & & (A \otimes (B \otimes C)) \otimes D \arrow[dl, "\alpha_{A,B,C} \otimes D"] \\
& ((A \otimes B) \otimes C) \otimes D &
\end{tikzcd}
\tag{55}
$$

$$
\begin{tikzcd}
A \otimes (B \otimes C) \arrow[r, "A \otimes \sigma_{B,C}"] \arrow[d, "\alpha_{A,B,C}"'] & A \otimes (C \otimes B) \arrow[d, "\alpha_{A,C,B}"] \\
(A \otimes B) \otimes C \arrow[d, "\sigma_{A \otimes B,C}"'] & (A \otimes C) \otimes B \arrow[d, "\sigma_{A,C} \otimes B"] \\
C \otimes (A \otimes B) \arrow[r, "\alpha_{C,A,B}"'] & (C \otimes A) \otimes B
\end{tikzcd}
\tag{56}
$$

$$
\begin{tikzcd}
A \otimes B \arrow[rr, "\sigma_{A,B}"] \arrow[dr, "1_{A \otimes B}"'] & & B \otimes A \arrow[dl, "\sigma_{B,A}"] \\
& A \otimes B &
\end{tikzcd}
\tag{57}
$$

What we have defined so far, could be called a *symmetric monoidal category without unit*. This terminology is not standard, because the notion has not much been used in mathematics. What is standard is the notion of *monoidal category* and *symmetric monoidal category* (the first one being without the σ), which additionally have a distinguished *unit object* $\mathbf{1}$ and natural isomorphisms $\lambda_A \colon \mathbf{1} \otimes A \to A$ and $\varrho_A \colon A \otimes \mathbf{1} \to A$ obeying the equations

$$
\begin{tikzcd}
A \otimes (\mathbf{1} \otimes B) \arrow[rr, "\alpha_{A,\mathbf{1},B}"] \arrow[dr, "A \otimes \lambda_B"'] & & (A \otimes \mathbf{1}) \otimes B \arrow[dl, "\varrho_A \otimes B"] \\
& A \otimes B &
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
\mathbf{1} \otimes A \arrow[rr, "\sigma_{\mathbf{1},A}"] \arrow[dr, "\lambda_A"'] & & A \otimes \mathbf{1} \arrow[dl, "\varrho_A"] \\
& A &
\end{tikzcd}
\tag{58}
$$

An important property of monoidal categories is MacLane's *coherence theorem*. Stated in terms of proofs, it says the following:

2.8.1 Theorem *Let $n \geq 1$ and $A_1, \ldots, A_n$ be formulas. Now let B and C be formulas built from $A_1, \ldots, A_n$ by using $\otimes$ such that every A_i appears exactly once in B and C. If Axioms* (i)–(v) *hold, then all proofs from B to C formed with the available data are equal. This proof always exists, is an isomorphism, and is natural in all n arguments.*

We will not give a proof here.

For being able to really speak about logic and proofs, we need negation, which is introduced by the following axioms:

[9]At this point you should start to see why it makes sense to use the category theoretical language. Without it, we would have, for example, to postulate for all formulas A, B, and C another proof $\alpha^{-1}_{A,B,C}$ such that the two derivations

$$
\begin{array}{c}
A \otimes (B \otimes C) \\
\alpha_{A,B,C} \, \Vert \\
(A \otimes B) \otimes C \\
\alpha^{-1}_{A,B,C} \, \Vert \\
A \otimes (B \otimes C)
\end{array}
\qquad \text{and} \qquad
\begin{array}{c}
(A \otimes B) \otimes C \\
\alpha^{-1}_{A,B,C} \, \Vert \\
A \otimes (B \otimes C) \\
\alpha_{A,B,C} \, \Vert \\
(A \otimes B) \otimes C
\end{array}
$$

are both doing nothing (i.e., are equal to the identity proof). Furthermore, we would need a lot of equations in the form of (52), in order to express the naturality.

(vi) For every formula A there is another formula $A^\perp$, and for every proof $f\colon A \to B$, there is another proof $f^\perp\colon B^\perp \to A^\perp$ such that $1_A^\perp = 1_{A^\perp}\colon A^\perp \to A^\perp$ and such that $(g \circ f)^\perp = f^\perp \circ g^\perp\colon C^\perp \to A^\perp$ for every $f\colon A \to B$ and $g\colon B \to C$.

(vii) For every formula A and proof $f\colon A \to B$ we have that $A^{\perp\perp} = A$ and $f^{\perp\perp} = f$. (More precisely, the mapping $A^{\perp\perp} \to A$ is the identity on A).

Spoken in category theoretical terms, Axiom (vi) says that $(-)^\perp$ is a *contravariant endofunctor*. With this, we can define the $\parr$ via $A \parr B = (A^\perp \otimes B^\perp)^\perp$. Axiom (vii) says that if we flip around a derivation twice, we get back where we started from.[10] It also allows us to conclude that the $\parr$ that we just defined has the same properties as postulated for the $\otimes$ in (iv) and (v), i.e., it is a bifunctor and carries a monoidal structure (without unit).

2.8.2 Exercise Formulate the statements of Axioms (iv) and (v) for the $\parr$ defined via $A \parr B = (A^\perp \otimes B^\perp)^\perp$, and show that they follow from (i)–(vii).

Before stating our final postulates about proofs, let us introduce the following notation. For two formulas A and B, we write $\mathrm{Hom}(A, B)$ for the set of proofs from A to B, and we write $h^{\mathbb{1}}(B)$ for the set of proofs of B that have no premise.[11]

(viii) For all formulas A, B, and C, there is a bijection

$$\varphi\colon \mathrm{Hom}(A \otimes B, C) \to \mathrm{Hom}(A, B^\perp \parr C) \tag{59}$$

which is natural in all three arguments.

(ix) For all formulas A and B, we have a bijection

$$\varphi\colon h^{\mathbb{1}}(A^\perp \parr B) \to \mathrm{Hom}(A, B) \tag{60}$$

which is natural in both arguments and respects the monoidal structure.

In the case with units, Axiom (viii) would complete the definition of a **-autonomous category*. It essentially says that we are allowed to do *currying* and *uncurrying*. To see this, note that linear logic knows the connective $\multimap$, standing for *linear implication*, defined via $A \multimap B = A^\perp \parr B$.[12] Equation (59) now says that we can jump freely back and forth between proofs $A \otimes B \to C$ and $A \to B \multimap C$.[13]

Since we do not have units, we also need (ix), which says that the proofs of $A \multimap B$ are the same as the proofs $A \to B$. To be precise, we need to give additional equation saying that $h^{\mathbb{1}}$ is a functor, i.e., every proof $f\colon A \to B$ is mapped to a function $h^{\mathbb{1}}(f)\colon h^{\mathbb{1}}(A) \to h^{\mathbb{1}}(B)$ such that composition and identity are preserved. Furthermore, the $h^{\mathbb{1}}$ needs to go well along with the monoidal structure, to say what that means exactly would take us too far astray. But to give you an idea of the problem, let us figure out how we could construct a proof $B \to (A \parr A^\perp) \otimes B$, corresponding to the rule $\mathsf{i}{\downarrow}$ in Figure 5, by using the axioms (i)–(ix). If we had a unit $\mathbf{1}$ together with the equations (58), then it would be easy: we could start out with $\lambda_A\colon \mathbf{1} \otimes A \to A$, apply (59) to get

$$\hat{\lambda}_A = \varphi(\lambda_A)\colon \mathbf{1} \to A \parr A^\perp$$

By (iv), we can form a proof $\hat{\lambda}_A \otimes 1_B\colon \mathbf{1} \otimes B \to (A \parr A^\perp) \otimes B$. We can precompose this with $\lambda_B^{-1}\colon B \to \mathbf{1} \otimes B$, to get

$$(\hat{\lambda}_A \otimes 1_B) \circ \lambda_B^{-1}\colon B \to (A \parr A^\perp) \otimes B$$

Constructing this map without using the unit requires heavy category theoretical machinery that we are not going to show here. See Section 2.9 for references.

[10] What we impose here is also called *strictness*, and does usually not hold. For example, the double dual of a vector space is usually not the space itself. Even in the finite dimensional case we only have a natural isomorphism between A and $A^{\perp\perp}$.

[11] The reason for this notation is the following: $\mathrm{Hom}(A, B)$ is in fact the value of the functor $\mathrm{Hom}(-, -)$ in two arguments. The functor $\mathrm{Hom}(A, -)$ in one argument is also written as h^A. If there is a proper unit $\mathbf{1}$ then the proofs of A are the elements of the set $\mathrm{Hom}(\mathbf{1}, A)$, i.e., $h^{\mathbf{1}}$ is a functor mapping every formula to its set of proofs. In $h^{\mathbb{1}}$, the unit is *virtual*.

[12] As in classical logic, "A implies B" is the same as "not A or B".

[13] If you have never seen currying, think of a function f in two arguments, denoted by $f\colon A \times B \to C$. This is essentially the same as a function $f'\colon A \to B \to C$, taking and argument from the set A and returning a function $B \to C$ which asks for an element of B to finally return the result in C.

2.8.3 Exercise We mentioned switch in (52) and (53) but we did not postulate it in (i)–(ix). In this exercise you are asked to construct $\mathsf{s}_{A,B,C}\colon A\otimes(B \mathbin{⅋} C) \to (A \mathbin{⅋} B)\otimes C$, corresponding to the switch rule in Figure 5, by using (i)–(viii). Hint: Start with the identity $B\otimes C \to B\otimes C$ and apply (59). You will also need the associativity of ⅋ that you have constructed in Exercise 2.8.2.

The wonderful point of Axioms (i)–(ix) is that they precisely describe the mathematical structure spanned by cut-free proof nets for MLL^-. This means two things:

First, the proof nets for MLL^- that we discussed in the previous sections form a category: The objects are the formulas and the maps $A \to B$ are the cut-free proof nets with conclusion $\vdash A^\perp, B$,[14] and the composition $g \circ f$ of two maps $f\colon A \to B$ and $g\colon B \to C$ is defined by eliminating the cut from

$$\mathsf{cut}\,\frac{\overset{f}{\vdash A^\perp, B} \qquad \overset{g}{\vdash B^\perp, C}}{\vdash A^\perp, C}$$

In the calculus of structures, this corresponds to performing the composition

$$\begin{array}{c} A \\ \| \\ B \\ \| \\ C \end{array} \quad \to \quad \begin{array}{c} A \\ \| \\ C \end{array}$$

It is easy to verify that this category, denoted by **PN**, obeys (i)–(ix) (where $h^{\mathbb{1}}(A)$ is just the set of cut-free proof nets with conclusion A).

Second, the the category **PN** is the free category with this property. This means that whenever there is a category $\mathscr{C}$, obeying (i)–(ix), then there is a uniquely defined functor (i.e., map that preserves all the structure defined in (i)–(ix)) from **PN** to $\mathscr{C}$.

2.8.4 Theorem *The category of cut-free proof nets for* MLL^-*, with arrow composition defined by cut elimination, forms the free *-autonomous category without units* (*generated from the set* $\mathscr{A}$ *of propositional variables*)*.*

Another way of seeing this is that we can trivially translate proofs in the sequent calculus or the calculus of structures into the free *-autonomous category (without units), by simply following the syntax. Theorem 2.8.4 says, that if we do this to two proofs Π_1 and Π_2, we get the same map in the category, if and only if Π_1 and Π_2 yield the same proof net after cut elimination.

In other words, if you have no objections against any of the Axioms (i)–(ix), you must answer the Big Question 2.7.9 with **yes**.

But there is also a **but**: Let us emphasize that this yes is valid only for MLL^-. What we have said in this section does **not** allow us to draw any conclusions about any other logic.

2.9 Notes

As already mentioned in the introduction, the terminology of "proof nets" and "bureaucracy" is due to Girard. He introduced proof nets along with sequent calculus presentation for linear logic in [Gir87]. He essentially followed Ideology 2.2.1 for obtaining his proof nets. The terminology "coherence" is due to MacLane. In [Mac63] he proves the "coherence theorem" for symmetric monoidal categories. See also [Mac71]. The concept of coherence graph is based in the work of Eilenberg, Kelly, and MacLane [EK66, KM71], who also provided the acyclicity condition and observed that it is preserved by composition, i.e., cut elimination. The observation that cut elimination is composition in a category is due to Lambek [Lam68, Lam69]. The terminology "flow-graph" is due to Buss [Bus91].[15]

The calculus of structures has been discovered by Guglielmi [Gug02], who initiated the systematic proof theoretic investigation of the concept of deep inference. For more details on this see [GS01, BT01, Str03a, Brü03].

[14] Or, equivalently, the two-sided proof nets where A has polarity $\bullet$ and B has polarity $\circ$.

[15] Strictly speaking, coherence graphs and flow graphs are not the same thing. But in the simple case of MLL^-, the two notions coincide.

The notion of "correctness criterion" is also due to Girard. In [Gir87] he gave the "long-trip-criterion" that we did not present here. The splitting tensor theorem (our Lemma 2.5.6) also first appeared in [Gir87]. The proof given in Section 2.5 follows the presentation of Bellin and van de Wiele in [BvdW95], who also give a proof of Theorem 2.2.3 and discuss in more detail the relation between proof nets and trivial rule permutations. Another well-written short discussion on this issue can be found in [Laf95]. Our second proof of Theorem 2.5.5 (i.e., the one using the calculus of structures) follows the presentation in [Str03a]. However, the result is already implicit present in the work of [DHPP99] and [Ret97]. A different way of proving Theorem 2.5.5 via the calculus of structures is presented in [Joi06].

The switching criterion (Definition 2.5.4 and Theorem 2.5.5) is due to Danos and Regnier [DR89]. For this reason the switching criterion is in the literature also called Danos-Regnier-criterion or DR-criterion. However, the contraction criterion is also due to Danos and Regnier[16] and should therefore also be called DR-criterion. See [Moo02, Pui01] for a more recent investigation of the contraction criterion. That (a version of) the contraction criterion can be checked in linear time in the size of the net has been discovered by Guerrini [Gue99]. The RB-correctness criterion has been found by Retoré [Ret93, Ret99a, Ret03], who provided a detailed analysis of proof nets using RB-graphs in various forms.

The concept of two-sided proof nets must be considered as folklore. In several early papers on proof nets the possibility of a two-sided version is mentioned, but the details are never carried out because the one-sided version is more economic. In the literature, two-sided proof nets are used when for some reason the authors want to avoid the use of negation (e.g., [BCST96, FP04]). The concept of polarities has to be attributed to Lamarche (e.g., [Lam95]). In [Lam01] he develops the algebraic theory of polarities and structural contexts in full detail.

The notion of cut elimination has been developed by Gentzen [Gen34, Gen35]. For a variant of linear logic it has first been proved by Lambek [Lam58]. For full linear logic (sequent calculus and proof nets) it has been proved by Girard [Gir87]. For linear logic presented in the calculus of structures, the first direct proof of cut elimination was also based on rule permutation (similar to the sequent calculus) [Str03a, Str03b]. The idea of using splitting as sketched in the proof of Theorem 2.7.3 is due to Guglielmi [Gug02].

*-Autonomous categories have been discovered by Barr [Bar79]. That there is a relation to linear logic was discovered immediately after the introduction of linear logic (see, e.g., [Laf88, See89]). Blute [Blu93] was the first to note that the category of proof nets is actually the free *-autonomous category without units. However, no complete proof was given; there was no proper definition of a *-autonomous category without units. That there is in fact a non-trivial mathematical problem to give such a definition was observed only 12 years later, but then by three research groups independently at the same time [LS05a, DP05, HHS05] (see also Sections 3.5 and 5.1). The most in-depth treatment is [HHS05]. We used here the notation of [LS05a].

The terminology of "Formalism A" and "Formalism B" is due to Guglielmi [Gug04a, Gug04b]. See also [Hug04, McK05, Str05a, Str05b] for the relation between deep inference and category theory.

3 Other fragments of linear logic

In this section we will very briefly inspect how proof nets for larger fragments of linear logic look like. We will first look at the so-called exponentials (which are modalities) and the additives (which are a second pair of conjunction/disjunction). Then we will go back to the purely multiplicative fragment and play with variations of it.

3.1 Multiplicative exponential linear logic (without units)

The formulas of unit-free multiplicative exponential linear logic (MELL^-) are generated by the syntax

$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{A}^\perp \mid \mathscr{F} \parr \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F} \mid !\mathscr{F} \mid ?\mathscr{F}$$

where everything is as in Section 2.1. The modalities are dual to each other:

$$(!A)^\perp = ?A^\perp \qquad (?A)^\perp = !A^\perp$$

The inference rules in the sequent calculus are the same as in (1) plus the ones for the new modalities:

$$?\mathsf{w}\,\frac{\vdash \Gamma}{\vdash ?A, \Gamma} \qquad ?\mathsf{c}\,\frac{\vdash ?A, ?A, \Gamma}{\vdash ?A, \Gamma} \qquad ?\mathsf{d}\,\frac{\vdash A, \Gamma}{\vdash ?A, \Gamma} \qquad !\,\frac{\vdash A, ?B_1, \ldots, ?B_n}{\vdash !A, ?B_1, \ldots, ?B_n} \tag{61}$$

[16] It first appears in Danos' thesis [Dan90], but he insists that it is joint work with Regnier.

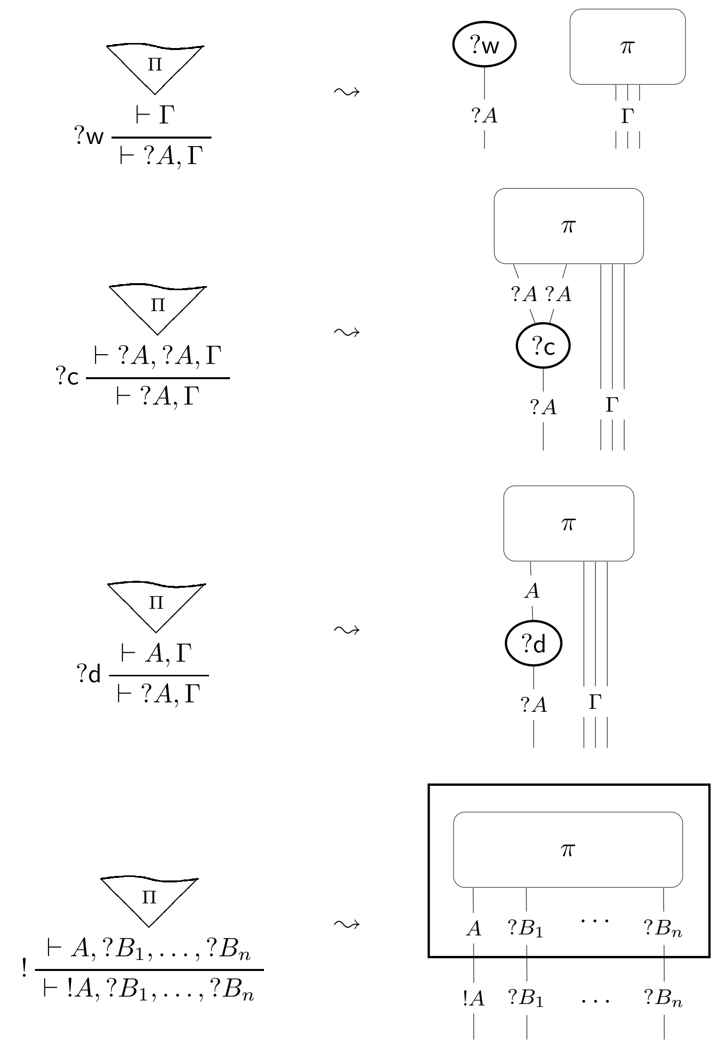


Figure 10: From sequent calculus to proof nets: exponentials

where in the !-rule $n \geq 0$. In Figure 10 we show how these rules are translated into proof nets according to Ideology 2.2.1. This is exactly the way how Girard introduced them in [Gir87]. Note that for dealing with the !-modality, we do not exactly follow Ideology 2.2.1. Instead, the concept of *box* around a proof net is introduced. Such a box always has a main door, the formula $!A$, and auxiliary doors, which are all occupied by ?-formulas.

The correctness of such a proof net with boxes is defined as in Section 2.5 (?c behaves as $⅋$) with the difference, that each box has to be treated separately. However, the correctness criteria do only work if no instance of ?w is present.[17]

Cut elimination is not as simple as for MLL^-. There are now not only more reduction rules, we also have the problem that showing termination is no longer trivial because the size of the net can increase during the reduction process. This is due to the presence of contraction. In the sequent calculus, the problematic case

[17] There is a general problem with multiplicative proof nets when some form of weakening is around. We come back to this in Sections 3.5 and 5.1.

appears when

$$\mathsf{cut}\,\dfrac{\mathsf{?c}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_1}{\vdash \Gamma, ?A, ?A}}{\vdash \Gamma, ?A} \qquad \mathsf{!}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_2}{\vdash A, ?B_1, \ldots, ?B_n}}{\vdash !A, ?B_1, \ldots, ?B_n}}{\vdash \Gamma, ?B_1, \ldots, ?B_n}$$

is reduced to

$$\mathsf{?c}\,\dfrac{\mathsf{?c}\,\dfrac{\mathsf{cut}\,\dfrac{\mathsf{cut}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_1}{\vdash \Gamma, ?A, ?A} \qquad \mathsf{!}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_2}{\vdash A, ?B_1, \ldots, ?B_n}}{\vdash !A, ?B_1, \ldots, ?B_n}}{\vdash \Gamma, ?A, ?B_1, \ldots, ?B_n} \qquad \mathsf{!}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_2}{\vdash A, ?B_1, \ldots, ?B_n}}{\vdash !A, ?B_1, \ldots, ?B_n}}{\vdash \Gamma, ?B_1, \ldots, ?B_n, ?B_1, \ldots, ?B_n}}{\vdots}}{\vdash \Gamma, ?B_1, \ldots, ?B_n}$$

where the proof Π_2 has been duplicated. In terms of proof nets, this means that the whole box with π_2 inside is duplicated.

3.1.1 Exercise Visualize this reduction in terms of proof nets by using the translation of Figure 10.

In spite of this nasty behavior, we have the following theorem, which we are not going to prove her. The interested reader is referred to [Gir87, Dan90, Joi92].

3.1.2 Theorem *Cut elimination for* MELL^- *proof nets is terminating, confluent, and preserves correctness.*

So far, there are no proof nets for MELL^- that follow Ideology 2.3.1. One of the reasons might be that the sequent calculus rules in (61) do not allow to properly trace the modalities in the derivation. However, in the calculus of structures, this becomes very natural (see [Str03a, Str03b, GS02]).

3.1.3 Open Research Problem Find for MELL^- a notion of proof nets without boxes, that is based on Ideology 2.3.1.

3.2 Multiplicative additive linear logic (without units)

Let us now turn to unit-free multiplicative additive linear logic (MALL^-). The formulas of MALL^- are generated by the syntax

$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{A}^\perp \mid \mathscr{F} \parr \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F} \mid \mathscr{F} \oplus \mathscr{F} \mid \mathscr{F} \mathbin{\&} \mathscr{F}$$

where again everything is as in Section 2.1. The two new connectives are dual to each other:

$$(A \mathbin{\&} B)^\perp = B^\perp \oplus A^\perp \qquad\qquad (A \oplus B)^\perp = B^\perp \mathbin{\&} A^\perp$$

The inference rules in the sequent calculus are the same as in (1) plus the ones for the new connectives:

$$\mathbin{\&}\,\dfrac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A \mathbin{\&} B, \Gamma} \qquad \oplus_1\,\dfrac{\vdash A, \Gamma}{\vdash A \oplus B, \Gamma} \qquad \oplus_2\,\dfrac{\vdash B, \Gamma}{\vdash A \oplus B, \Gamma} \tag{62}$$

In his first proposal for proof nets for MALL^-, Girard [Gir87] used boxes (as for the exponentials). The problem with this approach is that it distinguishes between the two proofs

$$\mathbin{\&}\,\dfrac{\mathbin{\&}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_1}{\vdash A, C, \Gamma} \quad \genfrac{}{}{0pt}{}{\Pi_2}{\vdash A, D, \Gamma}}{\vdash A, C \mathbin{\&} D, \Gamma} \qquad \mathbin{\&}\,\dfrac{\genfrac{}{}{0pt}{}{\Pi_3}{\vdash B, C, \Gamma} \quad \genfrac{}{}{0pt}{}{\Pi_4}{\vdash B, D, \Gamma}}{\vdash B, C \mathbin{\&} D, \Gamma}}{\vdash A \mathbin{\&} B, C \mathbin{\&} D, \Gamma} \tag{63}$$

and

$$\&\,\dfrac{\&\,\dfrac{\begin{matrix}\Pi_1\\ \vdash A, C, \Gamma\end{matrix} \quad \begin{matrix}\Pi_3\\ \vdash B, C, \Gamma\end{matrix}}{\vdash A \mathbin{\&} B, C, \Gamma} \qquad \&\,\dfrac{\begin{matrix}\Pi_2\\ \vdash A, D, \Gamma\end{matrix} \quad \begin{matrix}\Pi_4\\ \vdash B, D, \Gamma\end{matrix}}{\vdash A \mathbin{\&} B, D, \Gamma}}{\vdash A \mathbin{\&} B, C \mathbin{\&} D, \Gamma} \tag{64}$$

In his second proposal [Gir96], Girard proposed the notion of *monomial proof nets* (still following Ideology 2.2.1), where he introduces the notion of slice and attaches a monomial (boolean) weight to the identity links.

Here we will sketch a very recent proposal by Hughes and van Glabbeek [HvG03], which follows Ideology 2.3.1. Remember that in Section 2.3, we observed that the additional graph structure that captures the essence of the proof for MLL^- consist of a linking, which is just a pairing of dual atom occurrences.

The discovery of Hughes and van Glabbeek was that for MALL^-, this additional graph structure is a *set of linkings*, where a linking is again, simply a pairing of dual atom occurrences. But this time, a single linking need not to be exhaustive (i.e., it does not necessarily pair up everyone).

We can extract the proof net from the sequent calculus proof in the same way as in Section 2.3. But when we encounter a $\&$-rule, we have to separate the two linkings of the two branches, and we have to keep track of which pairing belongs to which linking. Figure 11 shows an example (taken from [HvG03]) where we do this by choosing different line style/colors.

In [HvG03], Hughes and van Glabbeek provide a correctness criterion and a cut elimination that preserves the correctness. Part of the correctness criterion is that each linking itself has to be multiplicatively correct (in the sense of Section 2.5). Unfortunately, we do here not have time and space to go into the details. However, from the viewpoint of the identity of proofs, there is an important observation to make about these proof nets: The following two sequent proofs are mapped to the same proof net for all possible Π_1, Π_2, and Π_3:

$$\&\,\dfrac{\otimes\,\dfrac{\begin{matrix}\Pi_1\\ \vdash \Gamma, A, C\end{matrix} \quad \begin{matrix}\Pi_3\\ \vdash D, \Delta\end{matrix}}{\vdash \Gamma, A, C \otimes D, \Delta} \qquad \otimes\,\dfrac{\begin{matrix}\Pi_2\\ \vdash \Gamma, B, C\end{matrix} \quad \begin{matrix}\Pi_3\\ \vdash D, \Delta\end{matrix}}{\vdash \Gamma, B, C \otimes D, \Delta}}{\vdash \Gamma, A \mathbin{\&} B, C \otimes D, \Delta} \tag{65}$$

$$\otimes\,\dfrac{\&\,\dfrac{\begin{matrix}\Pi_1\\ \vdash \Gamma, A, C\end{matrix} \quad \begin{matrix}\Pi_2\\ \vdash \Gamma, B, C\end{matrix}}{\vdash \Gamma, A \mathbin{\&} B, C} \qquad \begin{matrix}\Pi_3\\ \vdash D, \Delta\end{matrix}}{\vdash \Gamma, A \mathbin{\&} B, C \otimes D, \Delta} \tag{66}$$

In other words, the proof nets identify the two sequent proofs (65) and (66). Is this wanted from the proof theoretical perspective? At the current state of the art there is no definite answer.

An important point against the identification is that in (65) the subproof Π_3 appears twice, while in (66) it appears only once, which means that with the identification of (65) and (66) it becomes difficult to speak about the size of proofs.

An important point in favor of the identification comes from algebra, where adding the connectives $\&$ and $\oplus$ means adding cartesian products and coproducts to the axioms of *-autonomous categories. If we want that $\&$ behaves as cartesian product in our category of proofs, we have to identify (65) and (66).

Maybe in the end both worlds have their right to exist. In any case, the observation above leads to the following:

3.2.1 Open Research Problem Find for MALL^- a notion of proof nets that does not identify (65) and (66). A good starting point could be to consider flow-graphs in the calculus of structures for MALL (see [Str02, Str03a]). Of course, the problem is to find the right correctness criterion and the right notion of cut elimination.

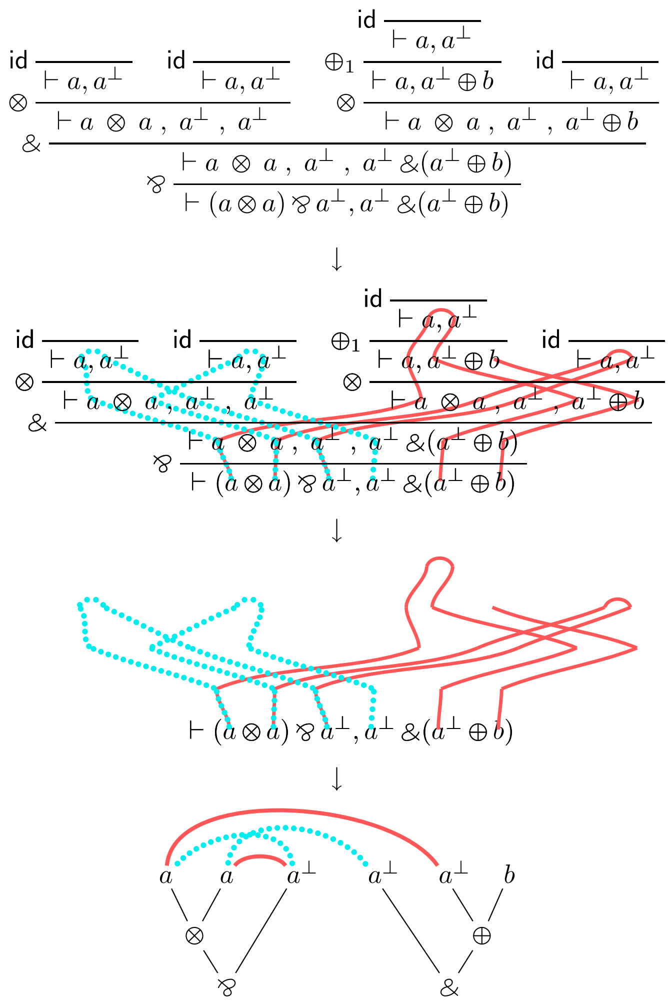


Figure 11: From sequent calculus to MALL proof nets via coherence graphs

3.3 Intuitionistic multiplicative linear logic (without unit)

Formulas of intuitionistic multiplicative linear logic[18] without unit (IMLL^-) are generated by the syntax:

$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{F} \multimap \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F} \tag{67}$$

[18]This is the logic that Kelly and MacLane studied in their seminal paper [KM71]. Of course, they did not use that name. But they did have the correctness criterion of proof nets, called coherence graphs. They only had the acyclicity condition and not the connectedness condition because the unit was present. They proved their results via cut elimination (they used that terminology). They even envisaged the notion of polarity.

Note that there is no $\parr$ and no $\cdot^\perp$. The sequent calculus for IMLL^- is usually given in two-sided form:

$$
\begin{array}{cc}
\mathsf{id}\,\dfrac{}{A \vdash A} & \mathsf{cut}\,\dfrac{\Gamma \vdash A \quad A, \Gamma' \vdash C}{\Gamma, \Gamma' \vdash C} \\[2ex]
\multicolumn{2}{c}{\mathsf{exchL}\,\dfrac{\Gamma, B, A, \Gamma' \vdash C}{\Gamma, A, B, \Gamma' \vdash C}} \\[2ex]
\otimes\mathsf{L}\,\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} & \otimes\mathsf{R}\,\dfrac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\[2ex]
\multimap\mathsf{L}\,\dfrac{\Gamma \vdash A \quad \Gamma', B \vdash C}{\Gamma, \Gamma', A \multimap B \vdash C} & \multimap\mathsf{R}\,\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}
\end{array}
\tag{68}
$$

Note that the right-hand side of each sequent contains at most one formula. Apart from that, the rules are exactly the same as in (27). We can therefore simply take our notion of two-sided proof nets from Section 2.6. The correctness criterion remains unchanged. This can be used to prove that IMLL^- is a conservative extension of MLL^-, i.e., an IMLL^- formula is provable in IMLL^- if and only if it is provable in MLL^-.[19]

We can use polarities to provide a one-sided version of proof nets for IMLL^-. For this we define the set of negative and positive formulas as follows:

$$
\begin{array}{rl}
\mathscr{F}^\bullet ::= \mathscr{A}^\perp & | \; \mathscr{F}^\bullet \parr \mathscr{F}^\bullet \; | \; \mathscr{F}^\bullet \otimes \mathscr{F}^\circ \; | \; \mathscr{F}^\circ \otimes \mathscr{F}^\bullet \\
\mathscr{F}^\circ ::= \mathscr{A} & | \; \mathscr{F}^\circ \otimes \mathscr{F}^\circ \; | \; \mathscr{F}^\circ \parr \mathscr{F}^\bullet \; | \; \mathscr{F}^\bullet \parr \mathscr{F}^\circ
\end{array}
\tag{69}
$$

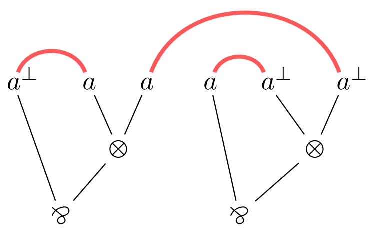
We can now define a proof net for IMLL^- to be a proof net for MLL^- in which at most one of the conclusions is in $\mathscr{F}^\circ$ and all other conclusions are in $\mathscr{F}^\bullet$. See [LR96] or [Lam01] for further details.

3.4 Cyclic linear logic (without units)

In Section 2.1 we defined sequents to be lists of formulas, but we also had the exchange rule which made the order of the formulas irrelevant for provability, and indeed, the two connectives $\parr$ and $\otimes$ were commutative. In this section we discuss what happens if we drop the exchange rule. As expected, the two connectives $\parr$ and $\otimes$ will lose their property of being commutative, and we enter the realm of non-commutative logics. But not everything works as one might expect...

Recall that in Observation 2.3.2 we noted that in the proof nets we always get a crossing of edges when we apply the exchange rule. Let us hence change the correctness criterion. We keep everything as in Section 2.5, but we add as condition that there must be no crossing of edges in the proof net (in graph theoretical terminology: the graph has to be *planar*). Hence, our running examples from Sections 2.2–2.4 are not correct in the sense of this new criterion. And indeed, we used the exchange rule in all sequent proofs.

Luckily, there is another proof net for our favorite sequent in which there are no crossings:



(70)

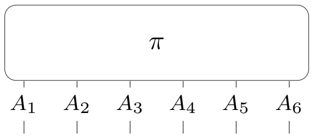
It clearly obeys the correctness criterion.

3.4.1 Exercise Give a sequent calculus proof in the system in (1) that translates into the net (70). Then try to find such a proof that does not use the exchange rule.

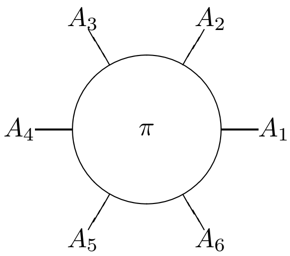
Here we are in for a surprise. It is very easy to find a sequent calculus proof for (70), but it is impossible to find one that does not use the exch-rule. What went wrong?

[19] This is not true for full intuitionistic linear logic with respect to linear logic. Also classical logic is certainly not a conservative extension of intuitionistic logic, since there are formulas, like Peirce's law $((A \to B) \to A) \to A)$ that are provable in classical logic, but not in intuitionistic logic.

There are two explanations. The first is based on the following observation. If we have a planar proof net, i.e., a list of formula trees together with a perfect matching of their atom occurrences such that there is no crossing among the links that connect the atoms, then we can from that data not tell *a priori* which formula is the first in the list. We have to think of the formulas to be grouped in a *cycle* around the graph with the axiom links. We now have to think of a proof net not as something like



but as something like

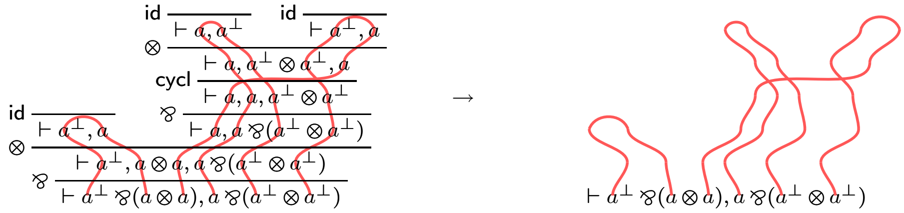


To go along with this behavior, we cannot just drop the exchange rule, but have replace it by the *cyclic exchange rule*:

$$\mathsf{cycl}\,\frac{\vdash \Gamma, A}{\vdash A, \Gamma}$$

where Γ is an arbitrary list of formulas. This is also the reason why the name *cyclic linear logic* has been chosen for this logic. It has been investigated in detail by Yetter [Yet90].

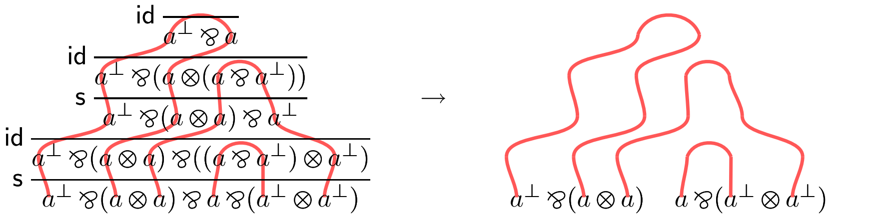
We can now give a sequent proof that translates into (70):



But there is still this disturbing fact that there are crossings in the flow-graph but no crossings in the net.

This leads us to our second explanation: The sequent calculus is simply too rigid to fully capture the phenomenon. Recall that in Observation 2.4.4 we noted that in the calculus of structures we can provide a derivation such that crossings appear in the flow-graph only if they also appear in the net. This means we should now be able to come up with a derivation that has no crossings in the flow-graph, and hence does not need the exchange rule nor a cyclic replacement. Of course, we need to adapt the system a little bit: we need another version of $\mathsf{i}{\downarrow}$ and $\mathsf{i}{\uparrow}$, namely the one in (13) and we need two versions of switch. Note that since we do not have $\sigma{\downarrow}$ nor $\sigma{\uparrow}$, we cannot generate the different four versions of switch as in (12). The new system is shown in Figure 12 (both versions of switch are self-dual).

Here is now our favorite example without crossings:



That deep inference is powerful enough to get rid of the cyclic exchange rule has first been observed by Di Gianantonio [DG04]. But he did not use proof nets to show this fact.

$$\mathsf{i}{\downarrow}\,\frac{}{A^\perp \parr A} \qquad \mathsf{i}{\uparrow}\,\frac{A \otimes A^\perp}{}$$

$$\mathsf{i}{\downarrow}\,\frac{S\{B\}}{S\{(A^\perp \parr A) \otimes B\}} \qquad \mathsf{i}{\uparrow}\,\frac{S\{(A \otimes A^\perp) \parr B\}}{S\{B\}}$$

$$\mathsf{i}{\downarrow}\,\frac{S\{B\}}{S\{B \otimes (A^\perp \parr A)\}} \qquad \mathsf{i}{\uparrow}\,\frac{S\{B \parr (A \otimes A^\perp)\}}{S\{B\}}$$

$$\alpha{\downarrow}\,\frac{S\{A \parr (B \parr C)\}}{S\{(A \parr B) \parr C\}} \qquad \alpha{\uparrow}\,\frac{S\{A \otimes (B \otimes C)\}}{S\{(A \otimes B) \otimes C\}}$$

$$\mathsf{s}\,\frac{S\{A \otimes (B \parr C)\}}{S\{(A \otimes B) \parr C\}}$$

$$\mathsf{s}\,\frac{S\{(A \parr B) \otimes C\}}{S\{A \parr (B \otimes C)\}}$$

Figure 12: A system for cyclic MLL^- in the calculus of structures

By combining the last two sections, i.e., proof nets for cyclic linear logic and intuitionistic linear logic, we can obtain proof nets for the Lambek calculus [Lam58]. We leave the details as an exercise. The interested reader is referred to [LR96] or [Lam01], which contain a systematic treatment.

It is also possible to combine usual commutative MLL^- and cyclic MLL^- together in a single system. Then there is a par/tensor pair which is commutative and one which is non-commutative. This is done in [AR00].

3.5 Multiplicative linear logic with units

Let us now finally discuss the problem of the units. We stick to the multiplicative fragment of linear logic. The formulas of MLL are generated by

$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{A}^\perp \mid \mathscr{F} \parr \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F} \mid \perp \mid \mathbf{1}$$

where again everything is as in Section 2.1. The two units are dual to each other:

$$\perp^\perp = \mathbf{1} \qquad\qquad \mathbf{1}^\perp = \perp$$

The inference rules in the sequent calculus are the same as in (1) plus the ones for the units:

$$\perp\,\frac{\vdash \Gamma, \Delta}{\vdash \Gamma, \perp, \Delta} \qquad\qquad \mathbf{1}\,\frac{}{\vdash \mathbf{1}} \tag{71}$$

If we now follow the argumentation of avoiding trivial rule permutations as in (10), we have to identify the following three derivations in the sequent calculus:

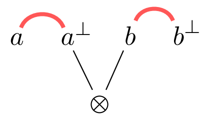
$$\otimes\,\frac{\perp\,\dfrac{\Gamma, A}{\perp, \Gamma, A} \quad B, \Delta}{\perp, \Gamma, A \otimes B, \Delta} \quad\longleftrightarrow\quad \perp\,\frac{\otimes\,\dfrac{\Gamma, A \quad B, \Delta}{\Gamma, A \otimes B, \Delta}}{\perp, \Gamma, A \otimes B, \Delta} \quad\longleftrightarrow\quad \otimes\,\frac{\Gamma, A \quad \perp\,\dfrac{B, \Delta}{\perp, B, \Delta}}{\perp, \Gamma, A \otimes B, \Delta}$$

because they are obtained from each other by permuting the $\perp$-rule under and over the $\otimes$-rule. Furthermore, following the equations that are imposed on proofs by the axioms of *-autonomous categories (with units) these three proofs have to be identified.

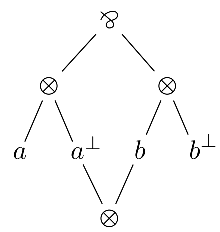
However, if we follow the idea of having a graph as proof net, we have to attach the $\perp$ somewhere. But there is no canonical place to do so. This is explained in further detail in [LS06]. That there is this problem with the

$\perp$-rule, and in fact with every kind of weakening rule has already been observed in [Gir87, Laf95, Gir96] and others.[20]

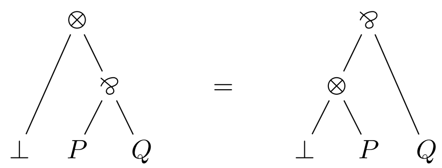
The solution to the problem is instead of considering graphs as proof nets, to consider equivalence classes of such graphs. There are now three different proposals in the literature to do so. In [BCST96] the authors use a two-sided version of proof nets and attach the $\perp$ to the edges in the graph (see also Section 5.1). We will show here the approach taken by [SL04, LS06] which clearly follows Ideology 2.3.1. The additional graph structure attached to the sequent forest does no longer consist of only the identity links, but of another formula tree. Then the resulting graph behaves as ordinary unit-free proof net and obeys the usual correctness criteria. The basic idea is the following. The ordinary proof net



is now written as



i.e., the identity links are replaced by $\otimes$-nodes which are connected by a $⅋$.[21] If we now have to attach a $\perp$, we attach it via a $\otimes$ to the root of linking tree of the proof to which we apply the $\perp$-rule. Figure 13 shows three examples. Since all three of them are the same proof according to what has been said above, we have to put them in the same equivalence class. To achieve this, we consider linking trees equivalent modulo the following equation:

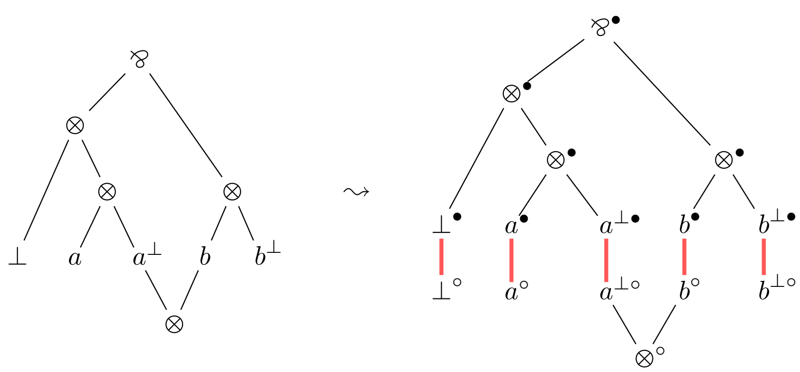


(72)

provided going from right to left in (72) does not destroy the correctness.[22]

The details of this kind of proof nets are carried out in [SL04, LS06], where it is also shown that they form the free *-autonomous category (in the usual sense, with units).

Let us note here only that the linking tree of an MLL proof net can be seen as the up-side in a two-sided unit-free proof net, as we discussed it in Section 2.6, provided we read the units $\perp$ and $\mathbf{1}$ as ordinary atoms. Here is an example:



[20] To be precise, one should note that the problem has already been noted in [KM71].

[21] If there are more than two axiom links we either use n-ary $⅋$ or take the equivalence class modulo associativity and commutativity of $⅋$ in the linking tree. This is not problematic since we have to take equivalence classes anyway.

[22] Observe that going to left to right cannot destroy correctness because this corresponds to an application of the switch rule in the calculus of structures, which always preserves correctness (see second proof of Theorem 2.5.5).

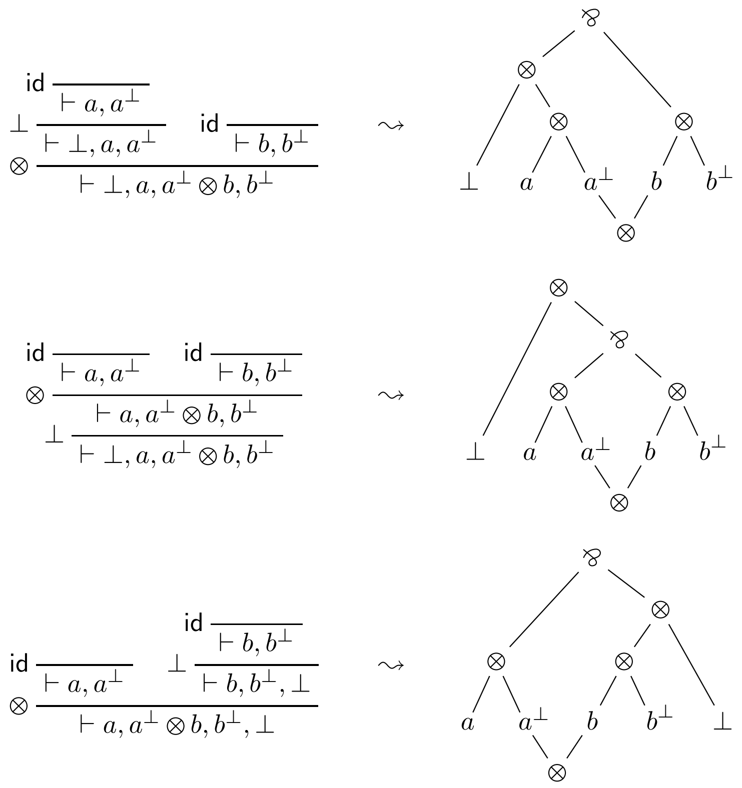


Figure 13: From sequent calculus to MLL proof nets with units

An alternative approach to the problem of the units has recently proposed by Hughes. In [Hug05b] he defines a notion for proof nets for MLL (also following Ideology 2.3.1) where he attaches the $\bot$ to some atom in the proof. This has the advantage that there is no need to consider equivalence classes of proof nets for defining cut elimination. However, for constructing the free *-autonomous category he also needs equivalence classes [Hug05a].

3.5.1 Open Research Problem Find for a notion of proof nets that is also able to accommodate the additive units $\top$ and 0 of linear logic.

3.6 Mix

In order to avoid the problems with the units, one can simply ignore them, as we did in the whole first part of these notes. An alternative is to add the rules

$$\mathsf{mix}\,\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta} \qquad \text{and} \qquad \mathsf{mix}_0\,\frac{}{\vdash} \tag{73}$$

called *mix* and *nullary mix*. Adding these two rule to (1) or (1)+(71) is equivalent to saying that $\bot = \mathbf{1}$. By doing this, the correctness criteria shown in Section 2.5 change only slightly: The acyclicity condition remains, but the connectedness condition has to be dropped. A detailed analysis for this can be found in [FR94].

3.7 Pomset logic and BV

Before we leave the realm of linear logic, let us show another variation, which has a particularly nice theory of proof nets. It has been introduced by Retoré [Ret97] under the name *pomset logic*. The set of formulas is generated via

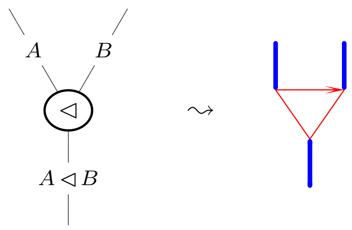
$$\mathscr{F} ::= \mathscr{A} \mid \mathscr{A}^{\bot} \mid \mathscr{F} \parr \mathscr{F} \mid \mathscr{F} \otimes \mathscr{F} \mid \mathscr{F} \lhd \mathscr{F}$$

Negation is defined as in Section 2.1. The new connective $\lhd$ is non-commutative and self-dual, i.e.,

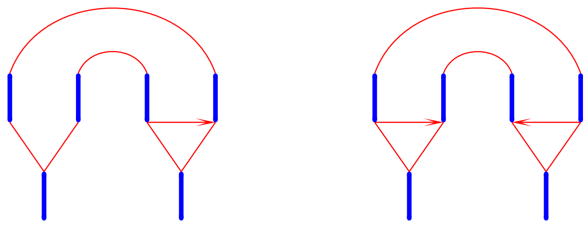
$$(A \lhd B)^{\perp} = A^{\perp} \lhd B^{\perp}$$

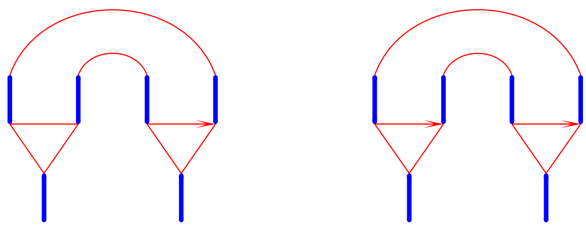
We will also allow to write $A \rhd B$ for $B \lhd A$.

For defining proof nets we use Retoré's RB-graph presentation. The $\parr$, the $\otimes$, the cut and the identity are translated into RB-graphs as in Section 2.5. The new connective is translated as follows:



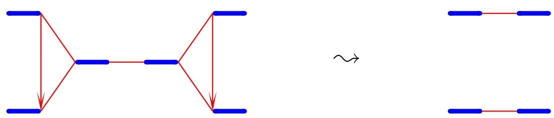
The correctness criterion hast to be changed such that in an Æ-cycle we can walk through a red edge with an arrow only in the direction of the arrow. Through undirected edges we can still walk either way. Furthermore, the connectedness condition has to be dropped.[23] Here are 4 examples:


(74)


(75)

The two examples in (74) fulfill the correctness criterion, while the two examples in (75) do not.

Note that we can define cut elimination as before, but we have to make sure that the arrows go in the right direction. Here is the new reduction rule that has to be added to the ones in (47) and (48):



(76)

With the same methods as in Section 2.7, we can show the following theorem:

3.7.1 Theorem *Cut elimination for pomset logic proof nets is terminating, confluent, and preserves correctness.*

You might already have wondered why for pomset logic we did not introduce the sequent calculus system before introducing the proof nets, as we did it with all other logics so far. The reason is simple. There is no sequent calculus system for pomset logic.

This of course questions the label "logic" for the object we have here, if there is not even an ordinary deductive system for it. And what is the meaning of cut elimination in that respect?

Fortunately, there is deep inference and the calculus of structures. The system shown in Figure 14 has been introduced by Guglielmi [Gug02] who called it SBV. For that system, we extend our language for formulas with another generator, the unit $\circ$. Furthermore, we consider formulas to be equivalent modulo the smallest congruence relation generated by the equations

$$A \parr \circ = A = \circ \parr A \qquad A \otimes \circ = A = \circ \otimes A \qquad A \lhd \circ = A = \circ \lhd A \tag{77}$$

[23] The reason for this is that because of the self-duality of the new connective $\lhd$ the two units $\perp$ and $\mathbf{1}$ are identified.

$$
\mathsf{i}{\downarrow}\,\frac{S\{\circ\}}{S\{A^\perp \parr A\}} \qquad \mathsf{i}{\uparrow}\,\frac{S\{A\otimes A^\perp\}}{S\{\circ\}}
$$

$$
\sigma{\downarrow}\,\frac{S\{A\parr B\}}{S\{B\parr A\}} \qquad \sigma{\uparrow}\,\frac{S\{A\otimes B\}}{S\{B\otimes A\}}
$$

$$
\alpha{\downarrow}\,\frac{S\{A\parr(B\parr C)\}}{S\{(A\parr B)\parr C\}} \qquad \alpha{\uparrow}\,\frac{S\{A\otimes(B\otimes C)\}}{S\{(A\otimes B)\otimes C\}}
$$

$$
\alpha^{\lhd}{\downarrow}\,\frac{S\{A\lhd(B\lhd C)\}}{S\{(A\lhd B)\lhd C\}} \qquad \alpha^{\lhd}{\uparrow}\,\frac{S\{(A\lhd B)\lhd C\}}{S\{A\lhd(B\lhd C)\}}
$$

$$
\mathsf{s}\,\frac{S\{A\otimes(B\parr C)\}}{S\{(A\otimes B)\parr C\}}
$$

$$
\mathsf{q}{\downarrow}\,\frac{S\{(A\parr C)\lhd(B\parr D)\}}{S\{(A\lhd B)\parr(C\lhd D)\}} \qquad \mathsf{q}{\uparrow}\,\frac{S\{(A\otimes C)\lhd(B\otimes D)\}}{S\{(A\lhd B)\otimes(C\lhd D)\}}
$$

Figure 14: System SBV in the calculus of structures

The reason for this is to avoid to have five different versions of $\mathsf{i}{\downarrow}$ and $\mathsf{i}{\uparrow}$.[24] Furthermore, we want the following derivation to be valid:

$$
\begin{array}{l}
= \dfrac{A\otimes B}{(A\lhd\circ)\otimes(\circ\lhd B)} \\
\mathsf{q}{\uparrow}\,\dfrac{}{(A\otimes\circ)\lhd(\circ\otimes B)} \\
= \dfrac{}{A\lhd B} \\
= \dfrac{}{(A\parr\circ)\lhd(\circ\parr B)} \\
\mathsf{q}{\downarrow}\,\dfrac{}{(A\lhd\circ)\parr(\circ\lhd B)} \\
= \dfrac{}{A\parr B}
\end{array}
\quad\rightsquigarrow\quad
\begin{array}{l}
\mathsf{q}{\uparrow}\,\dfrac{A\otimes B}{A\lhd B} \\
\mathsf{q}{\downarrow}\,\dfrac{}{A\parr B}
\end{array}
\tag{78}
$$

Here we implicitly use the "fake inference rule"

$$
=\frac{S\{B\}}{S\{A\}}
$$

where $A = B$ according to the equations in (77). Without it, we would need several more variants of $\mathsf{q}{\downarrow}$ and $\mathsf{q}{\uparrow}$, and the system would become rather big. The fragment of SBV without the up-rules (i.e., without the rules with the $\uparrow$ in the name) is called BV. As usual in the calculus of structures, the up-fragment corresponds to the cut, and we have for BV the cut elimination theorem (proved by Guglielmi [Gug02]):

3.7.2 Theorem *Let A be any pomset formula. Then,*

$$
\textit{for every derivation}\quad \begin{array}{c}\circ\\ \mathsf{SBV}\,\Big\Vert\\ A\end{array} \quad \textit{there is a derivation}\quad \begin{array}{c}\circ\\ \mathsf{BV}\,\Big\Vert\\ A\end{array}\ .
$$

The proof uses the technique of *splitting* and works essentially the same way as we have sketched it for Theorem 2.7.3. The logic that is defined by BV is the first logic that definitely needs deep inference. Alwen Tiu [Tiu01, Tiu06] has shown that there cannot be a shallow inference system (in particular, no sequent calculus system) defining the logic of BV.

We can apply the flow-graph method (see Section 2.4 to obtain proof nets from proofs in BV. Or, if you prefer, you can obtain two-sided proof nets from derivations in SBV. Everything that has been said in Section 2.6

[24] Remember that for cyclic linear logic we had already three of them.

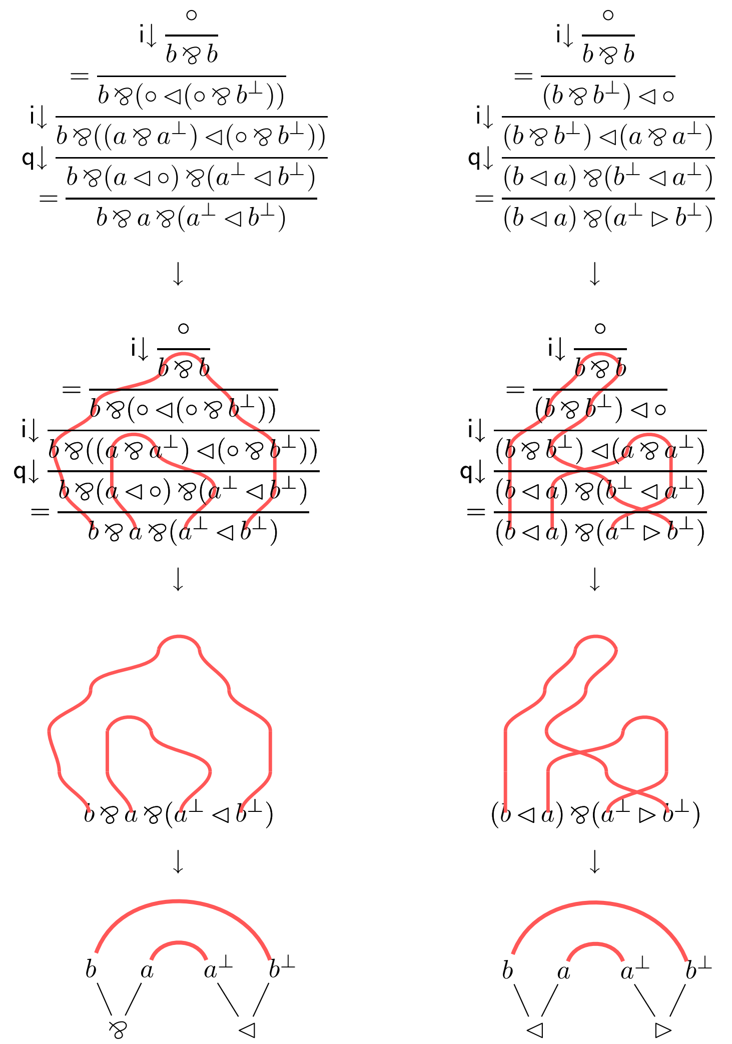


Figure 15: From BV to proof nets

does also apply here. Figure 15 shows two small examples of proof nets obtained from BV derivations. They, in fact, correspond to the two RB-graphs in (74).

The reason for mentioning all this is the following theorem:

3.7.3 Theorem *Let A and B be any pomset logic formulas. Then*

$$\mathsf{SBV}\,\begin{array}{c}A\\ \Big\Vert\\ B\end{array} \qquad \textit{if and only if} \qquad \mathsf{BV}\,\begin{array}{c}\circ\\ \Big\Vert\\ A^{\perp} \mathbin{⅋} B\end{array} \tag{79}$$

and if these derivations exist, then they have the same proof net, and this proof net obeys the pomset logic correctness criterion, when written as RB-graph.

Proof (Sketch): Let us start with the first part of the theorem which is easy, provided we have cut elimination: From

$$\mathsf{SBV}\,\begin{array}{c}A\\ \Big\Vert\\ B\end{array}$$

we can build

$$\mathsf{i}{\downarrow}\,\frac{\circ}{\begin{array}{c} A^{\perp} \bindnasrepma A \\ \mathsf{SBV}\,\Big\| \\ A^{\perp} \bindnasrepma B \end{array}}$$

Now we apply Theorem 3.7.2 to get

$$\begin{array}{c} \circ \\ \mathsf{BV}\,\Big\| \\ A^{\perp} \bindnasrepma B \end{array} \tag{80}$$

The other direction is even easier. From (80), we can directly build

$$\begin{array}{c} = \dfrac{A}{A \otimes \circ} \\ \mathsf{BV}\,\Big\| \\ \mathsf{s}\,\dfrac{A \otimes (A^{\perp} \bindnasrepma B)}{(A \otimes A^{\perp}) \bindnasrepma B} \\ \mathsf{i}{\uparrow}\,\dfrac{}{\circ \bindnasrepma B} \\ = \dfrac{}{B} \end{array}$$

Note that this part of the proof did not use the fact that we are speaking about BV. It in fact holds for any logical system in the calculus of structures, provided we have the switch rule and a cut elimination that preserves the flow-graph.[25]

Let us now come to the second more interesting and more difficult part or the theorem. There are two ways to show that every BV proof net obeys the pomset logic criterion. The first is to show that every rule in BV preserves the criterion (and that the unique net with $\circ$ as conclusion is correct). This has been done already by Retoré [Ret99b]. The second way, which has been used in [Str03a, GS02] is based on the following claims:

Claim 1: Let $n \geq 1$ and atoms $a_1, \ldots, a_n$ and formulas $A_1, \ldots, A_n$ be given, such that for all $i \in \{1, \ldots, n\}$, we have $A_i = a_i^{\perp} \otimes a_{i+1}$ or $A_i = a_i^{\perp} \lhd a_{i+1}$ (where we count indices modulo n). Then there is no derivation

$$\begin{array}{c} \circ \\ \mathsf{BV}\,\Big\| \\ A_1 \bindnasrepma \cdots \bindnasrepma A_n \end{array} \tag{81}$$

For proving the claim we proceed by way of contradiction and by induction on n. We have to perform a case analysis how the rules of BV could be applied inside $A_1 \bindnasrepma \cdots \bindnasrepma A_n$. We also need the concept of a quasi-subformula. We say that A is a *quasi-subformula* of B, if A can be obtained from B by replacing some atom occurrences by $\circ$.[26]

Claim 2: Let π be a pomset logic pre-proof net with conclusion B, and let A be a quasi-subformula of B such that whenever an atom occurrence is replaced by $\circ$, then also its mate (according to the linking in π) is replaced by $\circ$. Then we have the following:

$$\text{If} \quad \begin{array}{c} \circ \\ \mathsf{BV}\,\Big\| \\ B \end{array} \quad \text{then} \quad \begin{array}{c} \circ \\ \mathsf{BV}\,\Big\| \\ A \end{array} \quad .$$

This is easy to see because the proof of B remains valid if we replace pairs of atoms by $\circ$.

Claim 3: Let π be a pomset logic pre-proof net with conclusion B. Then π does not fulfill the pomset logic correctness criterion if and only if B has a quasi-subformula $A_1 \bindnasrepma \cdots \bindnasrepma A_n$, where all A_i are as in Claim 1, and all pairs a_i, $a_i^{\perp}$ are paired up in the linking of π.

Claims 1–3 together imply the result. □

[25]Roughly speaking, this is the same as saying that the proofs in the logic form some sort of *-autonomous category.
[26]When A is a quasi-subformula of B, then the relation web (see Definition 2.5.18) of A is a subweb of the relation web of B.

3.7.4 Remark The Claim 3 in the proof above states a second correctness criterion for BV/pomset logic, which does not need RB-graphs, and which first appears in [Str03a]. Claims 1 and 2 in the proof above show that BV derivations obey this criterion. Proving Claim 3 then means showing that the two criteria are indeed equivalent.

3.7.5 Exercise Complete the proof of the second part of Theorem (3.7.3), i.e., show that every proof net coming from a BV derivation obeys the pomset logic correctness criterion.

3.7.6 Open Research Problem Prove the converse. I.e., show that for every correct pomset logic proof net there is a BV derivation having it as flow graph.

Solving this problem would finally (because of Tiu's result [Tiu06]) show that it is indeed impossible to give a sequent calculus system for pomset logic.

4 Intuitionistic logic

We will not speak about intuitionistic logic in this course. But I will mention here some important facts related to the title of the lecture notes, because these facts have had a strong impact on the development of proof nets for classical logic that we will discuss in the next section.

The inference rules in natural deduction for implication in intuitionistic logic

$$\to\mathsf{I}\,\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \text{and} \qquad \to\mathsf{E}\,\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

are the same as the typing rules

$$\mathsf{abs}\,\frac{\Gamma, x\colon A \vdash u\colon B}{\Gamma \vdash \lambda x.u\colon A \to B} \qquad \text{and} \qquad \mathsf{app}\,\frac{\Gamma \vdash u\colon A \to B \quad \Gamma \vdash v\colon A}{\Gamma \vdash uv\colon B}$$

for the simply typed λ-calculus. This is the basis for the so-called *Curry-Howard-correspondence* (also known as *formulas-as-types-correspondence* and *proofs-as-programs-correspondence*). It is also called "isomorphism" because the normalization in natural deduction [Pra65] does the same as β-reduction in the λ-calculus.[27] If we add conjunction to the logic (or equivalently product types to the λ-calculus) we can use the proofs in natural deduction (or equivalently λ-terms) for specifying morphism in *cartesian closed categories* (short: *CCCs*). What makes things interesting is the fact that the identity forced on proofs by the notion of normalization in natural deduction (or equivalently the identity forced on λ-terms by normalization[28]) is exactly the same as the identity of morphism that is determined by the axioms of CCCs. For further details on this see [LS86]. Of course, this simple observation has been extended to more expressive logics and larger type systems (e.g., System F [Gir72], calculus of constructions [CH88], ...).

Since there is already a well-understood canonical and bureaucracy-free presentation of proofs in intuitionistic logic, namely terms in the (simply) typed λ-calculus, the concept of "proof nets for intuitionistic logic" is not well investigated. To my knowledge there are only two proposals in that direction. The first, by Lamarche [Lam94] uses the encoding of intuitionistic logic into (intuitionistic) linear logic (the multiplicative exponential fragment). The second, by Horbach [Hor06], restricts a class of proof nets for classical logic (that we will discuss in the next section) to intuitionistic logic. Of course, this cannot be as simple as for IMLL^- (see Section 3.3) because classical logic is *not* a conservative extension of intuitionistic logic.

5 Classical Logic

It is surely very tempting to extend the beautiful connection between deductive system, λ-calculus, and category theory from intuitionistic logic to classical logic.

We get from intuitionistic logic to classical logic by adding the law of excluded middle (i.e., $A \vee \bar{A}$), or equivalently, an involutive negation (i.e., $\bar{\bar{A}} = A$). Adding this to a Cartesian closed category $\mathscr{C}$, means adding a contravariant functor $\overline{(-)}\colon \mathscr{C} \to \mathscr{C}$ such that $\bar{\bar{A}} \cong A$ and $\overline{(A \wedge B)} \cong \bar{A} \vee \bar{B}$ where $A \vee B = \bar{A} \Rightarrow B$. However, if

[27] One could also argue that we have here just two different syntactic presentations of the same mathematical objects.
[28] β-reduction, η-expansion, and α-conversion

we do this we get a collapse: all proofs of the same formula are identified, which leads to a rather boring proof theory. This observation is due to André Joyal, and a detailed proof and discussion can be found in [LS86] and in the appendix of [Gir91].

We will not show the category theoretic proof of the collapse. But we will explain the phenomenon in terms of the sequent calculus (the argumentation is due to Yves Lafont [GLT89, Appendix B]). Suppose we have two proofs of the formula B in some sequent calculus system:

$$\begin{array}{c}\Pi_1\\ \vdash B\end{array} \qquad \text{and} \qquad \begin{array}{c}\Pi_2\\ \vdash B\end{array}$$

Then we can with the help of the rules weakening, contraction, and cut

$$\mathsf{weak}\,\frac{\vdash \Gamma}{\vdash \Gamma, A} \qquad \mathsf{cont}\,\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} \qquad \mathsf{cut}\,\frac{\vdash \Gamma, A \quad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta}$$

form the following proof of B

$$\mathsf{cont}\,\dfrac{\mathsf{cut}\,\dfrac{\mathsf{weak}\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash B\end{array}}{\vdash B, A} \qquad \mathsf{weak}\,\dfrac{\begin{array}{c}\Pi_2\\ \vdash B\end{array}}{\vdash A^{\perp}, B}}{\vdash B, B}}{\vdash B} \tag{82}$$

If we eliminate the cut from this proof, we get either

$$\mathsf{cont}\,\dfrac{\mathsf{weak}\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash B\end{array}}{\vdash B, B}}{\vdash B} \qquad \text{or} \qquad \mathsf{cont}\,\dfrac{\mathsf{weak}\,\dfrac{\begin{array}{c}\Pi_2\\ \vdash B\end{array}}{\vdash B, B}}{\vdash B} \tag{83}$$

depending on a nondeterministic choice. Now note that one can hardly find a reason why for any proof Π, the two proofs

$$\mathsf{cont}\,\dfrac{\mathsf{weak}\,\dfrac{\begin{array}{c}\Pi\\ \vdash B\end{array}}{\vdash B, B}}{\vdash B} \qquad \text{and} \qquad \begin{array}{c}\Pi\\ \vdash B\end{array} \tag{84}$$

should be distinguished. After all, duplicating a formula and immediately afterwards deleting one copy is not doing much. Also the laws of category theory tell us to identify the two.

On the other hand, if we want the nice relationship between deductive system and category theory, we need a confluent cut elimination, which means we have to equate the two proofs in (83). Consequently, by (84), we have to equate Π_1 and Π_2. Since there was no initial condition on Π_1 and Π_2, we conclude that any two proofs of B must be equal.

But the problem with weakening, which could in fact be solved by using mix (see Sections 3.6 and 5.2), is not the only one. We run into similar problems with the contraction rule. If we try to eliminate the cut from

$$\mathsf{cut}\,\dfrac{\mathsf{cont}\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A, A\end{array}}{\vdash \Gamma, A} \qquad \mathsf{cont}\,\dfrac{\begin{array}{c}\Pi_2\\ \vdash \bar{A}, \bar{A}, \Delta\end{array}}{\vdash \bar{A}, \Delta}}{\vdash \Gamma, \Delta} \quad . \tag{85}$$

we again have to make a nondeterministic choice. In the sections below, we will see a concrete example for this.

There are several possibilities to cope with these problems. The easiest is to say that there cannot be any good proof theory for classical logic, and stop thinking about the problem[29]. A more serious and more difficult

[29] and making sure that also other people do not think about the problem...

approach is of course to say that we have to drop some of the axioms, i.e., some of the equations that we would like to hold between proofs in classical logic. But which ones should go?

There are now essentially two different approaches, and both have their advantages and disadvantages.

1. The first says that the axioms of cartesian closed categories are essential and cannot be dispensed with. Instead, one sacrifices the duality between $\wedge$ and $\vee$. The motivation for this approach is that a proof system for classical logic can now be seen as an extension of the λ-calculus and the notion of normalization does not change. With this approach one has a term calculus for proofs, namely Parigot's $\lambda\mu$-calculus [Par92] and a denotational semantics [Gir91]. An important aspect is the computational meaning in terms of continuations [Thi97, SR98]. There is a well explored category theoretical axiomatization [Sel01], and, of course, a theory of proof nets [Lau99, Lau03], which is based on the proof nets for MELL (see Section 3.1). Although these proof nets are very important from the "normalization-as-computation" point of view, we will not discuss them here because at the current state of the art it is not clear how they can be used to identify proofs in various deductive systems for classical logic (sequent calculus, resolution, tableaux, ...).
2. The second approach considers the perfect symmetry between $\wedge$ and $\vee$ to be an essential facet of Boolean logic, that cannot be dispensed with. Consequently, the axioms of cartesian closed categories and the close relation to the λ-calculus have to be sacrificed. It is much less clear than in the first approach, what the category theoretical axiomatization [DP04, FP04, LS05a, McK05, Str05b, Lam06] should be, and how a theory of proof nets should look like. There are now two different versions in the literature, one following Ideology 2.2.1 (Section 5.1), and the other following Ideology 2.3.1 (Sections 5.2 and 5.3).

5.1 Sequent calculus rule based proof nets

Classical logic can be obtained from multiplicative linear logic by adding the rules of contraction and weakening. But for obvious reasons we will here change the notation and use the symbols $\wedge$ and $\vee$ instead of $\otimes$ and $⅋$. Negation will be denoted by $\bar{\cdot}$, instead of $\cdot^{\perp}$. In other words, our formulas are generated by the syntax

$$\mathscr{F} ::= \mathscr{A} \mid \bar{\mathscr{A}} \mid \mathscr{F} \vee \mathscr{F} \mid \mathscr{F} \wedge \mathscr{F} \tag{86}$$

Note that as before, we ignore the units[30] and push negation to the atoms. Here are the inference rules in the sequent calculus:

$$\begin{gathered}
\mathsf{id}\,\frac{}{\vdash \bar{A}, A} \qquad \mathsf{cut}\,\frac{\vdash \Gamma, A \quad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta} \\[1ex]
\mathsf{exch}\,\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta} \qquad \mathsf{weak}\,\frac{\vdash \Gamma}{\vdash A, \Gamma} \qquad \mathsf{cont}\,\frac{\vdash A, A, \Gamma}{\vdash A, \Gamma} \\[1ex]
\vee\,\frac{\vdash A, B, \Gamma}{\vdash A \vee B, \Gamma} \qquad \wedge\,\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \wedge B, \Delta}
\end{gathered} \tag{87}$$

Note that there are various different sequent calculus systems for classical propositional logic, starting with the one by Gentzen [Gen34]. For a systematic treatment, the reader is referred to [TS00]. The motivation for choosing the one in (87) is that it allows for the simplest notion for proof nets according to Ideology 2.2.1. When we look at Ideology 2.3.1 and the notion of flow-graphs in the next sections, it does not matter which sequent system is taken as starting point.

The rules in (87) are essentially the same as the ones in (1) plus the ones for contraction and weakening. In order to obtain proof nets according to Ideology 2.2.1, we can therefore proceed as shown in Figures 1 and 10.

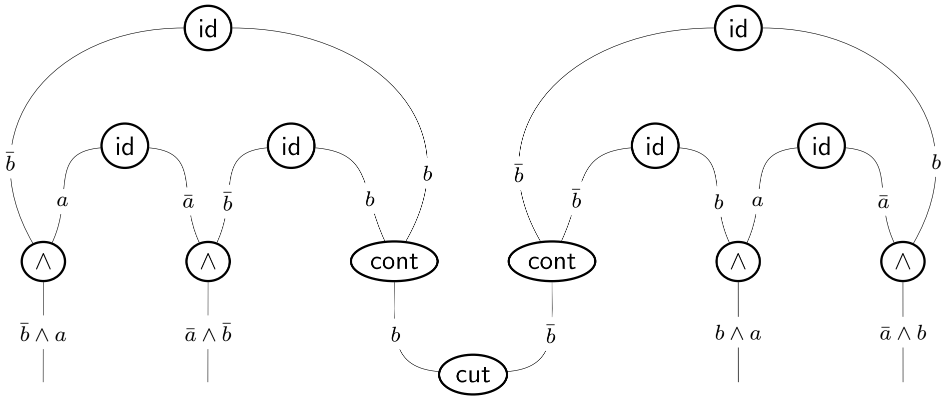
5.1.1 Exercise Give the translation of the rules in (87) into proof nets as it is done in Figures 1 and 10.

Here is an example of a sequent calculus proof

$$\mathsf{cut}\,\dfrac{\mathsf{cont}\,\dfrac{\wedge\,\dfrac{\wedge\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash \bar{b}, b} \quad \mathsf{id}\,\dfrac{}{\vdash a, \bar{a}}}{\vdash \bar{b} \wedge a, \bar{a}, b} \quad \mathsf{id}\,\dfrac{}{\vdash \bar{b}, b}}{\vdash \bar{b} \wedge a, \bar{a} \wedge \bar{b}, b, b}}{\vdash \bar{b} \wedge a, \bar{a} \wedge \bar{b}, b} \quad \mathsf{cont}\,\dfrac{\wedge\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash \bar{b}, b} \quad \wedge\,\dfrac{\mathsf{id}\,\dfrac{}{\vdash a, \bar{a}} \quad \mathsf{id}\,\dfrac{}{\vdash \bar{b}, b}}{\vdash \bar{b}, a, \bar{a} \wedge b}}{\vdash \bar{b}, \bar{b}, \bar{b} \wedge a, \bar{a} \wedge \bar{b}}}{\vdash \bar{b}, b \wedge a, \bar{a} \wedge b}}{\vdash \bar{b} \wedge a, \bar{a} \wedge \bar{b}, b \wedge a, \bar{a} \wedge b} \tag{88}$$

[30]Note, that in classical logic, this does not change the logic because we can pick a distinguished atom, say p_0, which may not appear in the formulas, and define $\mathbf{t} = p_0 \vee \bar{p}_0$ and $\mathbf{f} = p_0 \wedge \bar{p}_0$.

and its translation into a proof net:



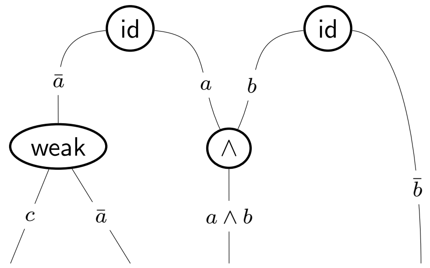
(89)

That this can easily be done has already been noted by Girard in the appendix of [Gir91], where he also explained the problems with this approach that we will discuss below. In [Rob03], Robinson carries out the details. He uses a two-sided version of the set of rules in (87), but with what we learned in Section 2.6, it is an easy exercise to translate back and forth between the one- and two-sided version.

Robinson also proves the soundness and completeness of the switching correctness criterion (see Definition 2.5.4 and Theorem 2.5.5). For the acyclicity condition this is not surprising since the rules in (87) are the same as in (1) (recall that contraction behaves as $\bindnasrepma$). The connectedness condition is obtained simply by attaching a weakening somewhere, similarly as we did it with $\bot$ in Section 3.5. The price to pay is that now certain proofs are distinguished that should be identified according the rule-permutability-argument. To see a very simple example, consider the following three sequent calculus proofs (we systematically omit the exchange rule).

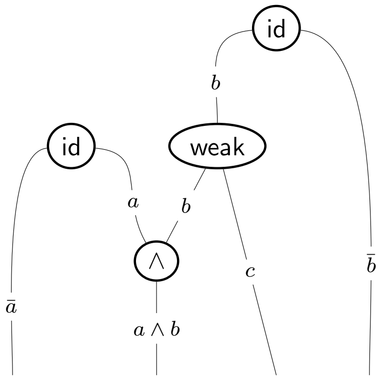
$$
\wedge\frac{\mathsf{weak}\dfrac{\mathsf{id}\dfrac{}{\vdash \bar{a}, a}}{\vdash c, \bar{a}, a} \qquad \mathsf{id}\dfrac{}{\vdash b, \bar{b}}}{\vdash c, \bar{a}, a \wedge b, \bar{b}}
\qquad
\mathsf{weak}\frac{\wedge\dfrac{\mathsf{id}\dfrac{}{\vdash \bar{a}, a} \qquad \mathsf{id}\dfrac{}{\vdash b, \bar{b}}}{\vdash \bar{a}, a \wedge b, \bar{b}}}{\vdash c, \bar{a}, a \wedge b, \bar{b}}
\qquad
\wedge\frac{\mathsf{id}\dfrac{}{\vdash \bar{a}, a} \qquad \mathsf{weak}\dfrac{\mathsf{id}\dfrac{}{\vdash b, \bar{b}}}{\vdash c, b, \bar{b}}}{\vdash c, \bar{a}, a \wedge b, \bar{b}}
\tag{90}
$$

They differ from each other only via some trivial rule permutation, and should therefore be identified. But according to [Rob03] they can be translated into five different proof nets. Two of them are shown below:



(91)

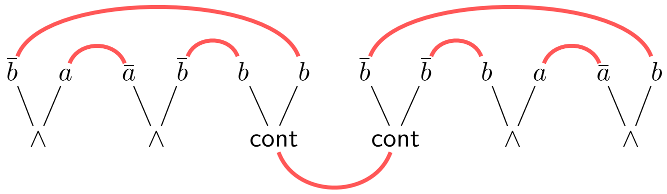
and



(92)

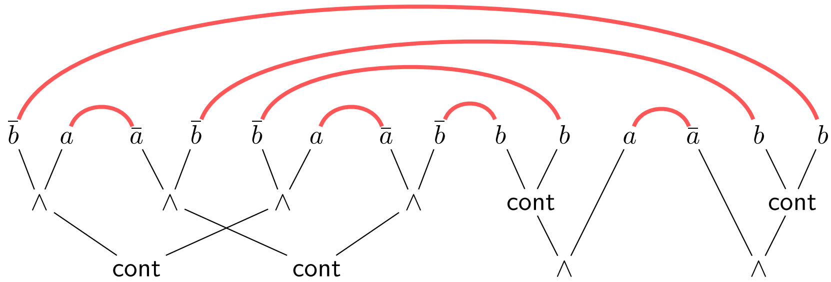
5.1.2 Exercise What are the other three proof nets corresponding to (90)? Compare the nets with the ones in Figure 13.

Let us now come to cut elimination. We have already seen in the introductory part of Section 5, that there is a problem with weakening. The short discussion in Section 3.1 suggests that there might also be a problem with contraction. Observe that in Section 3.1, the contraction rule could appear only on one side of the cut, never at both sides at the same time (because only ?-formulas could be contracted, never !-formulas). This is the reason why in the end we get confluence of cut elimination for MELL^- proof nets. However, for classical logic the situation is different. Contraction can appear on both sides of the cut, as it is shown in the example in (88) and (89). For typesetting reasons, let us use the more compact notation (as we also did in Section 2.5):



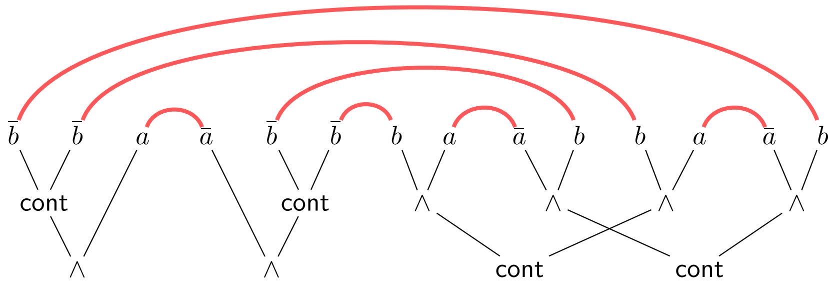
(93)

Let us stress that (89) and (93) are only different notations for the *same* proof net. Note that we have here an example for the general case in (85). If we want to eliminate the cut from (93), we have to make a nondeterministic choice, which subproof we duplicate. As outcome we get either



(94)

or



(95)

In the appendix of [Gir91], Girard argues that for this reason it is impossible to have a confluent notion of cut elimination for proof nets for classical logic. Of course, his argumentation is valid only for proof nets following Ideology 2.2.1.

5.1.3 Exercise Show that (94) and (95) obey the switching criterion, and give the sequent proofs corresponding to (94) and (95).

Although there is no confluent cut elimination, Führmann and Pym [FP04] managed to give a category theoretical axiomatization for the proof identifications made by these proof nets. This could be done because the authors dropped the "equation"

$$\text{cut elimination} \quad = \quad \text{arrow composition in the category of proofs}$$

and added an order enrichment instead. They defined for two proofs $f, g \colon A \to B$ that $f \preccurlyeq g$ iff f reduces to g via cut elimination. See [McK05] for relating this to the calculus of structures.

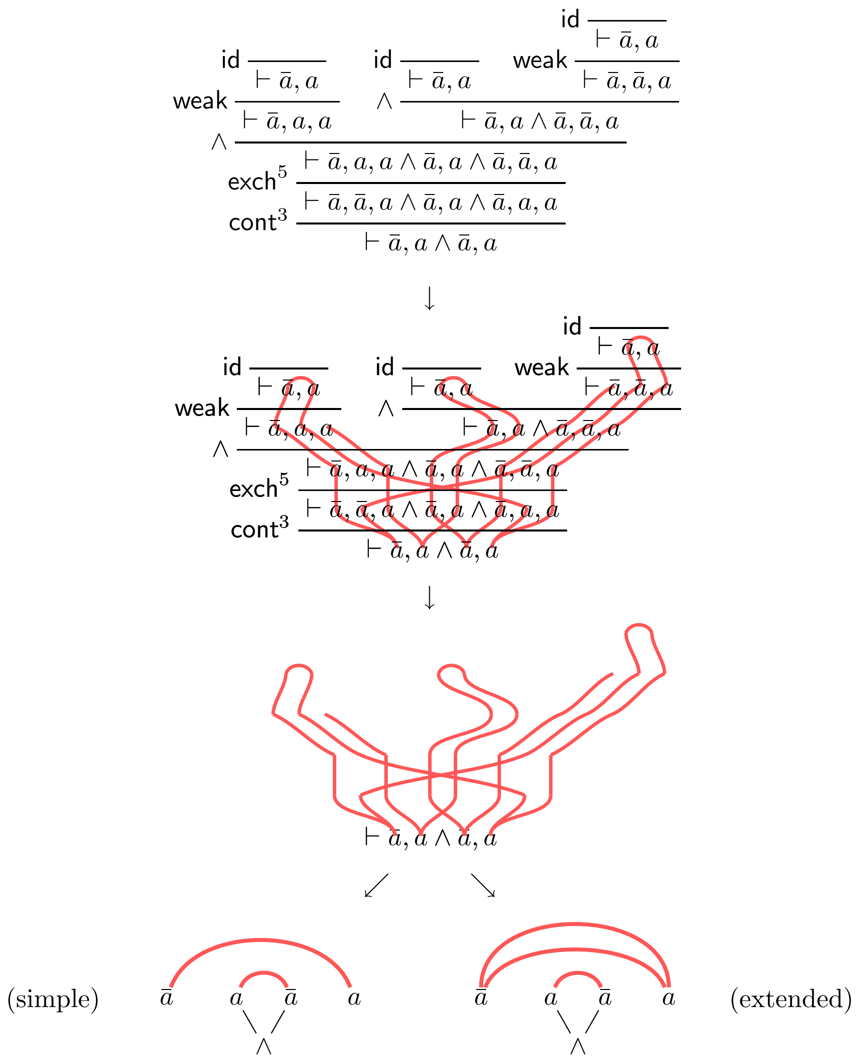


Figure 16: From sequent calculus to classical logic proof nets via flow graphs

5.2 Flow graph based proof nets (simple version)

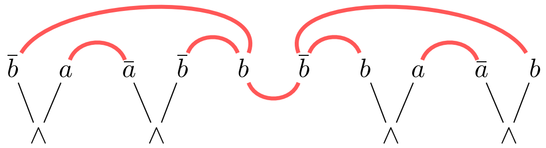
In the previous section we have seen what happens if we naively carry the approach of Section 2.2 to classical logic. Now we are going to see what happens if we naively carry the approach of Section 2.3 to classical logic. We can define a pre-proof net as in Definition 2.5.1, but now the identity links do not provide a perfect matching for the set of leaves of the sequent forest. It can happen that some atoms have no mate, i.e., live celibate, and that some atoms have more than one mate, i.e., live polygamous. It could even happen that there are two or more identity links between a pair of atoms, as the example (on the right) in Figure 16 shows. This example also shows that we can now completely forget about the correctness criteria that we have seen in Section 2.5. But most importantly, this example shows that for classical logic it makes a huge difference whether we follow Ideology 2.2.1 or Ideology 2.3.1 to obtain proof nets. This also means, that the term "proof net" should be used with care, because it is not necessarily clear what a proof net for classical logic actually is.

Let us, for the time being, restrict ourselves to those nets that have *at most one* identity link between any pair of atoms, and call them *simple prenets*:

5.2.1 Definition A *simple prenet* is a sequent forest Γ, possibly with cuts, together with a symmetric, irreflexive binary relation P on its set of leaves, such that whenever two atom occurrences are related by P then they must be dual to each other. We will use the notation $P \triangleright \Gamma$.

As in Section 2.5, we have to treat cuts as special formulas (see page 19).

In the left of Figure 16 we have shown an example how to translate a sequent calculus proof into a simple prenet. We draw the flow-graph as we did in Section 2.3 and forget how often a pair of atoms is connected to each other. Here is an example of a simple prenet with cut. It is the one obtained from (88):



(96)

We will call a simple prenet *sequentializable* if it can be obtained via the flow-graph method from the sequent calculus system in (87) extended by the mix-rule

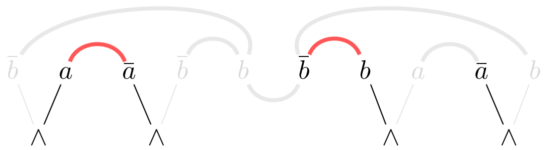
$$\mathsf{mix}\,\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}$$

5.2.2 Exercise Draw the simple prenets that are obtained from the sequent calculus proofs corresponding to the proof nets in (94) and (95). Hint: You can do this exercise without doing Exercise 5.1.3.

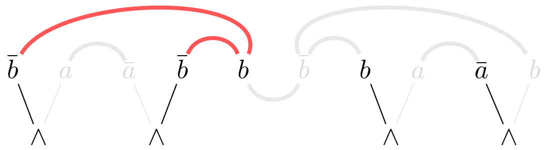
We are now going to define a correctness criterion for simple prenets.

5.2.3 Definition Let Γ be a sequent. A *conjunctive pruning* of Γ is the subforest obtained Γ by deleting one of the two immediate subformulas for every $\wedge$ and for every cut. Let $\pi = P \rhd \Gamma$ be a simple prenet. we call the prenet $\pi' = P' \rhd \Gamma'$ a *conjunctive pruning* of π if Γ' is a conjunctive pruning of Γ and P' is the restriction of P to (the set of leaves of) Γ'.[31]

Here are two examples for conjunctive prunings of (96):



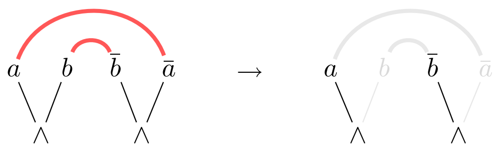
and



5.2.4 Exercise How many others are there?

5.2.5 Definition A simple prenet $P \rhd \Gamma$ is called *correct* (or, *obeys the pruning condition*, if every conjunctive pruning $P' \rhd \Gamma'$ of π contains at least one identity link, i.e., P' is not empty.

Clearly, the example in (96) is correct. Here is an example, which is not correct, because there is a pruning, in which all links disappear:



Interestingly, also in the correctness criterion given by Hughes and van Glabbeek in [HvG03] for MALL^- (see Section 3.2) the pruning condition plays a role.

[31]Strictly speaking, π' is not a simple prenet because Γ' is not a sequent because every $\wedge$ in Γ' is unary. For this reason π' is not a subprenet in the sense of Definition 2.5.7.

5.2.6 Theorem *A simple prenet π is correct, if and only if it is sequentializable.*

Proof: First, observe that the id-rule produces correct prenets and that all rules in (87), including mix, preserve correctness. For the rules $\vee$, exch, weak, cont, and mix, this is obvious. Let us show it here for the $\wedge$-rule. Let π_1 and π_2 be the simple prenets obtained from

$$\begin{array}{c}\Pi_1\\ \vdash \Gamma, A\end{array} \qquad \text{and} \qquad \begin{array}{c}\Pi_2\\ \vdash B, \Delta\end{array}$$

respectively. Let us assume by induction hypothesis that they are correct, and let π be the simple prenet obtained from

$$\wedge\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A\end{array} \quad \begin{array}{c}\Pi_2\\ \vdash B, \Delta\end{array}}{\vdash \Gamma, A \wedge B, \Delta}$$

Let π' be a conjunctive pruning of π. If it removes B from $A \wedge B$, then π' contains a link because π_1 is correct, and if π' removes A, then it must contain a link because π_2 is correct. For the cut-rule it is similar.

Conversely, let $\pi = P \rhd \Delta$ be a correct prenet. we will proceed by induction on the size of Δ (that is, the number of $\wedge$, $\vee$, atoms, and cuts appearing in it) to construct a a corresponding sequent proof using the rules in (87) plus mix.

- If Δ contains a formula $A \vee B$, then we can apply the $\vee$-rule and proceed by induction hypothesis.

- If Δ contains a formula $A \wedge B$, i.e., $\Delta = \Gamma, A \wedge B$, then we can form three correct simple prenets $\pi_1 = P' \rhd \Gamma, A$, and $\pi_2 = P'' \rhd \Gamma, B$, and $\pi_3 = P \rhd \Gamma, A, B$, where P' and P'' are the restrictions of P to Γ, A and Γ, B, respectively. Since π_1, and π_2, and π_3 are all correct, we get three sequent calculus proofs

$$\begin{array}{c}\Pi_1\\ \vdash \Gamma, A\end{array} \qquad \begin{array}{c}\Pi_2\\ \vdash \Gamma, B\end{array} \qquad \begin{array}{c}\Pi_3\\ \vdash \Gamma, A, B\end{array}$$

  from which we can form the following proof (we omit the instances of exch):

$$\mathsf{cont}\,\dfrac{\vdots}{\vdash \Gamma, A \wedge B} \quad \text{where the top is} \quad \mathsf{cont}\,\dfrac{\wedge\,\dfrac{\wedge\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A\end{array} \quad \begin{array}{c}\Pi_3\\ \vdash \Gamma, A, B\end{array}}{\vdash \Gamma, \Gamma, A \wedge B, A} \quad \begin{array}{c}\Pi_2\\ \vdash \Gamma, B\end{array}}{\vdash \Gamma, \Gamma, \Gamma, A \wedge B, A \wedge B}}{\vdots} \tag{97}$$

  which translates into π. Let us make three remarks about that case:

  - Note that we made crucial use of the fact that we forget how often an identity link is used inside the proof.
  - The proof Π_3 is needed for keeping the links that cross $A \wedge B$. If there is in π no link between an atom in A and an atom in B, then we do not need Π_3 and could replace (97) by

$$\mathsf{cont}\,\dfrac{\mathsf{cont}\,\dfrac{\wedge\,\dfrac{\begin{array}{c}\Pi_1\\ \vdash \Gamma, A\end{array} \quad \begin{array}{c}\Pi_2\\ \vdash \Gamma, B\end{array}}{\vdash \Gamma, \Gamma, A \wedge B}}{\vdots}}{\vdash \Gamma, A \wedge B}$$

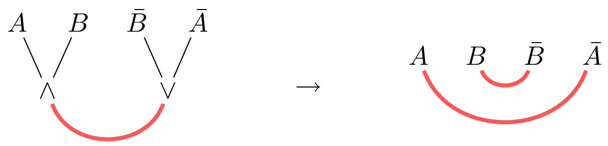
– For dealing with cuts we proceed similarly. But to be formally precise, we need to allow to apply the contraction rule not only to formulas in the sequent, but also to cuts.

- If $\Delta = \Gamma, A$ such that no link is coming out of A, then we apply the weakening rule and proceed by induction hypothesis.
- The only remaining case is where all formulas in Δ are atoms. Then our sequent calculus proof is obtained by a sufficient number of instances of id, cont, exch, and mix. □

We will use the term *simple proof net* for the correct (i.e., sequentializable) simple prenets. They have been introduced in [LS05b]. However, the idea of looking at paired atom occurrences in classical logic is much older. In [And76] Andrews applied the "flow-graph" method to resolution proofs and called the result *matings*. He had, in fact, exactly the same correctness criterion as we have seen above. Independently, Bibel [Bib81] investigated the same objects and called them *connection proofs*. Also the term matrix proofs is used because formulas have been written in form of matrices.

However, both authors, Andrews and Bibel, did not allow the linking of two atoms that are connected in the tree by a conjunction[32]. From the viewpoint of provability/correctness, this perfectly makes sense. Note that such links (we have an example in Figure 16) disappear in *every* conjunctive pruning. So, why having them in the first place?

The reason is cut elimination, i.e., the composition of proofs. We define it in the same way as for MLL^- in Section 2.7. In the case of a cut on a compound formula, we do the obvious thing:

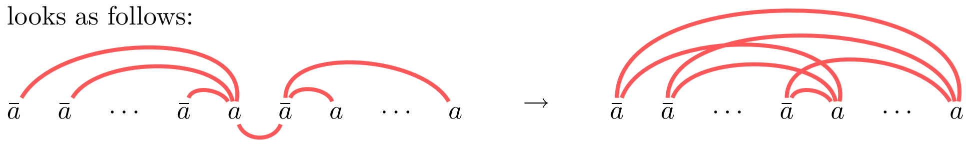


(98)

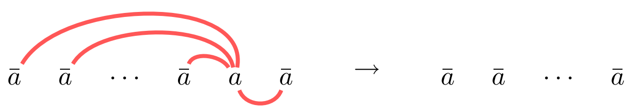
To see that this preserves correctness, note that a conjunctive pruning of the cut on the left yields either A or B or $\bar{B} \vee \bar{A}$, and a conjunctive pruning of the cut on the right yields either A, B or $A, \bar{B}$ or $\bar{A}, B$ or $\bar{A}, \bar{B}$. Hence, every conjunctive pruning of the reduced simple prenet contains a conjunctive pruning of the non-reduced simple prenet. Therefore it most contain at least one link.

For the cut reduction on atomic cuts, we have to be careful, since the atoms can be connected to many other atoms (or no other atoms). Instead of simply having:

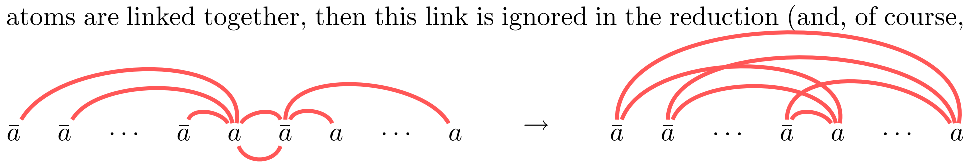
the reduction looks as follows:



If one of the two cut atoms is celibate, no link remains:



If the two cut atoms are linked together, then this link is ignored in the reduction (and, of course, removed with the cut):



The atomic cut reduction also preserves correctness. If we could in the reduced simple prenet construct a conjunctive pruning that does not contain an identity link, then the same pruning would not contain a link in the non-reduced net, where one cut atom is always removed.

[32]That means they are connected by a red edge in the relation web, see Definition 2.5.18.

5.2.7 Theorem *Cut elimination for simple proof nets preserves correctness, is confluent, and terminating.*

Proof: Preservation of correctness has already been shown above. Termination is obvious since each step reduces the size of the simple proof net. For confluence we only need to consider atomic cuts since the reduction of compound cuts does not create critical pairs. Let π be a simple proof net with atomic cuts and let π' be the result of reducing these atomic cuts. Then there is a link between two dual atom occurrences a and $\bar{a}$ in π', if either that link is already present in π, or there is an alternating link-cut-link-cut-...-link path in π that connects a and $\bar{a}$. This is independent from the order in which the atomic cuts are reduces. $\square$

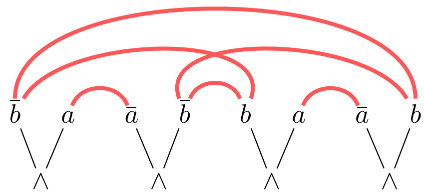
The natural question that arises now is: How does this confluent cut elimination relate to the non-confluent cut elimination in the sequent calculus?

Let us look again at the two problematic cases (82) and (85). The problem with weakening (82) can easily be solved by using the mix-rule:

$$\mathsf{cut}\,\dfrac{\mathsf{weak}\,\dfrac{\overset{\Pi_1}{\vdash \Gamma}}{\vdash \Gamma, A} \qquad \mathsf{weak}\,\dfrac{\overset{\Pi_2}{\vdash \Delta}}{\vdash \bar{A}, \Delta}}{\vdash \Gamma, \Delta} \qquad \to \qquad \mathsf{mix}\,\dfrac{\overset{\Pi_1}{\vdash \Gamma} \qquad \overset{\Pi_2}{\vdash \Delta}}{\vdash \Gamma, \Delta}$$

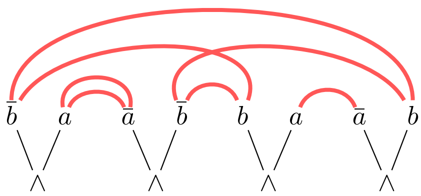
Both subproofs Π_1 and Π_2 are kept in the reduced net. With the help of mix, this can also be done in the sequent calculus, and in simple proof nets it is done in the same way.

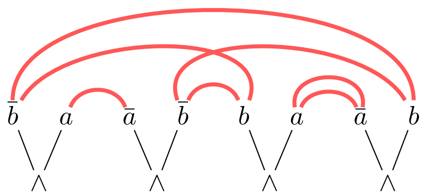
For the contraction case (85) the situation is less obvious. Consider again the simple proof net in (96), which corresponds to the sequent calculus proof in (88). If we apply the cut elimination for simple proof nets, we obtain the following result:



(99)

If you did Exercise 5.2.2, you will notice that this is exactly the simple proof net obtained from the sequent proofs corresponding to (94) and (95). However, let us emphasize that this correspondence also makes crucial use of the fact that we deliberately forget how often an identity link is used in the proof. If we kept this information, the proofs in (94) and (95) would be represented by



and


respectively. See [LS05b] for further details.

It should be clear that simple proof nets are not particularly connected to the sequent calculus. We can obtain them in the same way from proofs presented in the calculus of structures. Figure 17 shows a deductive system for classical logic in the calculus of structures. It is a (unit-free) variation of system SKS, presented in [BT01]. The rules are the same as the ones for MLL^- in Figure 5. As in the sequent calculus, we add the rules for contraction, weakening, and mix. We also add the m-rule, called *medial.* As it is the case with mix, it is is not necessary from the viewpoint of provability, but its presence gives the system a much nicer proof-theoretic behavior. See [BT01, Brü03, Str05b, Lam06] for further details.

Most of the theory on the relation between proof nets, sequent calculus and calculus of structures, that has been developed in Section 2, can be ported easily to classical logic. In particular, we can obtain Theorem 2.6.2 and the commutativity of (49), provided we restrict ourselves to simple proof nets.

5.2.8 Open Research Problem Find a category theoretical axiomatization that generates the same identification for proofs as it is done by simple proof nets for classical logic. (This is meant in the same sense as

$$
\mathsf{i}\!\downarrow \frac{}{\bar{A} \vee A} \qquad \mathsf{i}\!\uparrow \frac{A \wedge \bar{A}}{}
$$

$$
\mathsf{i}\!\downarrow \frac{S\{B\}}{S\{(\bar{A} \vee A) \wedge B\}} \qquad \mathsf{i}\!\uparrow \frac{S\{B \vee (A \wedge \bar{A})\}}{S\{B\}}
$$

$$
\sigma\!\downarrow \frac{S\{A \vee B\}}{S\{B \vee A\}} \qquad \sigma\!\uparrow \frac{S\{A \wedge B\}}{S\{B \wedge A\}}
$$

$$
\alpha\!\downarrow \frac{S\{A \vee (B \vee C)\}}{S\{(A \vee B) \vee C\}} \qquad \alpha\!\uparrow \frac{S\{A \wedge (B \wedge C)\}}{S\{(A \wedge B) \wedge C\}}
$$

$$
\mathsf{s} \frac{S\{A \wedge (B \vee C)\}}{S\{(A \wedge B) \vee C\}}
$$

$$
\mathsf{mix} \frac{S\{A \wedge B\}}{S\{A \vee B\}}
$$

$$
\mathsf{m} \frac{S\{(A \wedge B) \vee (C \wedge D)\}}{S\{(A \vee C) \wedge (B \vee D)\}}
$$

$$
\mathsf{c}\!\downarrow \frac{S\{A \vee A\}}{S\{A\}} \qquad \mathsf{c}\!\uparrow \frac{S\{A\}}{S\{A \wedge A\}}
$$

$$
\mathsf{w}\!\downarrow \frac{S\{B\}}{S\{B \vee A\}} \qquad \mathsf{w}\!\uparrow \frac{S\{A \wedge B\}}{S\{B\}}
$$

Figure 17: A system for classical logic in the calculus of structures

*-autonomous categories provide the axiomatization for proof nets for MLL, and cartesian closed categories for typed λ-terms.)

There is already preliminary research in this direction in [LS05b, LS05a, Str05a, Lam06, Str05b], but there is still no satisfactory solution.

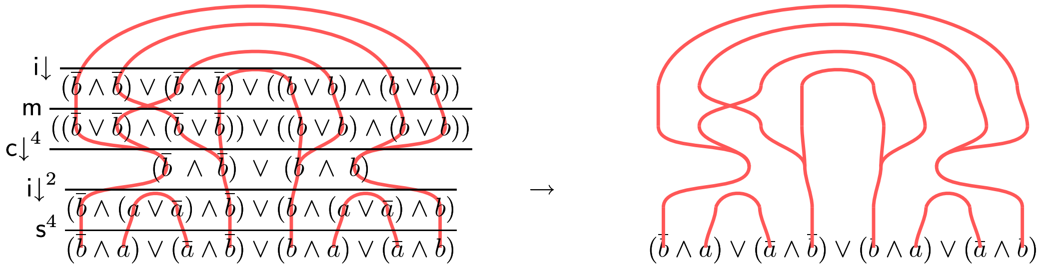
5.3 Flow graph based proof nets (extended version)

The first problem with simple proof nets is that they are too simple. When speaking about the identity of proofs one should also take into account the size of proofs. But simple proof nets have a size at most quadratic in the size of the conclusion sequent. This means they are not able to observe any kind of complexity, not even the exponential blow-up related to cut elimination.

The second problem with simple proof nets is that they are too simple. They omit too much important information about the proof. Checking their correctness takes exponential time, which is not faster than trying to prove the conclusion from scratch. But checking a proof should be a feasible task, taking only linear time in the size of the proof.

A naive solution could be to simply keep track of how often an identity link is used in the proof, i.e., allow more than one link between a pair of dual atoms, as shown on the right in Figure 16. Let us call these new objects *extended prenets*.

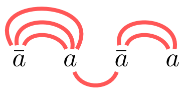
The first problem we encounter now is not only to define what *correctness* means, but also, what *sequentializable* means. To see the difficulty, consider again the simple proof net in (99). Now, all of the sudden, there is no sequent calculus proof that has it a flow-graph. However, we can find a proof in the calculus of structures, in the system shown in Figure 17:



This means that for extended prenets, it depends on the chosen deductive system whether a given prenet is sequentializable or not.

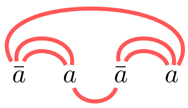
5.3.1 Open Research Problem Find a good notion of "sequentializability" and a corresponding correctness criterion for extended prenets.

The second, more serious problem comes with cut elimination. In [LS05b] it is explained, how cut elimination for extended prenets has to look like. For a cut on compound formulas it is the same as for simple proof nets. But for an atomic cut, we now have to multiply the number of edges. For example



reduces to

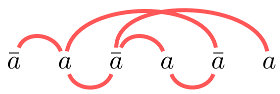
If there are already some links between the remaining pair of atoms, then these links have to be added. For example



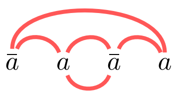
reduces to

Of course, we cannot say whether this preserves correctness, since we do not know what correctness means. But the good news is that cut reduction is still terminating, and that we can get an exponential blow-up in the size of the proof when doing cut elimination.

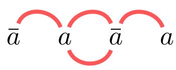
The bad news is that cut elimination is no longer confluent. This has already been observed in [LS05b]. The following example is taken from [Hor06]. Depending on which cut in



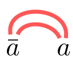
we reduce first, we get either



or



If we reduce the remaining cut, we get



or

respectively. The basic reason of this non-confluence is that we do not have the acyclicity condition anymore, and it depends on the order of the cut reductions how often "the flow-graph runs through a cycle". In [Hor06], Horbach proposes a way of keeping track of the cycles in order to get confluence for cut elimination on extended prenets representing intuitionistic proofs.

An alternative possible solution could be to redefine cut elimination such that no longer all atomic cuts are reduced. In [Str05a, Str05b] it is shown how this can be done such that we get a confluent cut elimination which corresponds to composition of derivations in the calculus of structures:

$$\begin{array}{c} A \\ \Vert \\ B \\ \Vert \\ C \end{array} \quad \to \quad \begin{array}{c} A \\ \Vert \\ C \end{array}$$

Which still allows to keep the equation

$$\text{cut elimination} \quad = \quad \text{arrow composition in the category of proofs}$$

But then cut elimination for extended prenets does no longer correspond to cut elimination in the sequent calculus. That it is indeed impossible to make the diagram (49) commute for the sequent calculus and extended prenets has already been shown with the example in (99) in the previous section. But it is not known whether we can make it commute for the calculus of structures.

5.3.2 Open Research Problem Find for the calculus of structures (possibly for the system in Figure 17) a cut elimination procedure, possibly based on splitting (as done in the proof of Theorem 2.7.3) such that it behaves in the same way as the cut elimination for extended prenets, i.e., such that diagram (49) commutes.

In any case, what has been said in this section shows that for classical logic the answer to the Big Question 2.7.9 on page 39 is no longer an obvious yes. In fact, the answer might be **No!**

5.3.3 Open Research Problem Make a nice theory out of this mess.

References

[And76] Peter B. Andrews. Refutations by matings. *IEEE Transactions on Computers*, C-25:801–807, 1976.

[AR00] Vito Michele Abrusci and Paul Ruet. Non-commutative logic I: The multiplicative fragment. *Annals of Pure and Applied Logic*, 101:29–64, 2000.

[Bar79] Michael Barr. **-Autonomous Categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, 1979.

[BCST96] Richard Blute, Robin Cockett, Robert Seely, and Todd Trimble. Natural deduction and coherence for weakly distributive categories. *Journal of Pure and Applied Algebra*, 113:229–296, 1996.

[Bib81] Wolfgang Bibel. On matrices with connections. *Journal of the ACM*, 28:633–645, 1981.

[Blu93] Richard Blute. Linear logic, coherence and dinaturality. *Theoretical Computer Science*, 115:3–41, 1993.

[Boo84] George Boolos. Don't eliminate cut. *Journal of Philosophical Logic*, 13:373–378, 1984.

[Brü03] Kai Brünnler. *Deep Inference and Symmetry for Classical Proofs*. PhD thesis, Technische Universität Dresden, 2003.

[BT01] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 347–361. Springer-Verlag, 2001.

[Bus91] Samuel R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53:72–102, 1991.

[BvdW95] Gianluigi Bellin and Jacques van de Wiele. Subnets of proof-nets in MLL^-. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 249–270. Cambridge University Press, 1995.

[CH88] Thierry Coquand and Gérard P. Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, 1988.

[Dan90] Vincent Danos. *La logique linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul).* Thèse de Doctorat, Université Paris VII, 1990.

[DG04] Pietro Di Gianantonio. Structures for multiplicative cyclic linear logic: Deepness vs cyclicity. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic, CSL 2004*, volume 3210 of *Lecture Notes in Computer Science*, pages 130–144. Springer-Verlag, 2004.

[DHPP99] H. Devarajan, Dominic Hughes, Gordon Plotkin, and Vaughan R. Pratt. Full completeness of the multiplicative linear logic of chu spaces. In *14th IEEE Symposium on Logic in Computer Science (LICS 1999)*, 1999.

[Doš03] Kosta Došen. Identity of proofs based on normalization and generality. *The Bulletin of Symbolic Logic*, 9:477–503, 2003.

[DP04] Kosta Došen and Zoran Petrić. *Proof-Theoretical Coherence.* KCL Publications, London, 2004.

[DP05] Kosta Došen and Zoran Petrić. Proof-net categories. preprint, Mathematical Institute, Belgrade, 2005.

[DR89] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Annals of Mathematical Logic*, 28:181–203, 1989.

[EK66] Samuel Eilenberg and Gregory Maxwell Kelly. A generalization of the functorial calculus. *Journal of Algebra*, 3(3):366–375, 1966.

[FP04] Carsten Führmann and David Pym. Order-enriched categorical models of the classical sequent calculus. To appear in *Journal of Pure and Applied Algebra*, 2004.

[FR94] Arnaud Fleury and Christian Retoré. The mix rule. *Mathematical Structures in Computer Science*, 4(2):273–285, 1994.

[Gen34] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1934.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[Gir72] Jean-Yves Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur.* PhD thesis, Université Paris VII, 1972.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gir91] Jean-Yves Girard. A new constructive logic: Classical logic. *Mathematical Structures in Computer Science*, 1:255–296, 1991.

[Gir96] Jean-Yves Girard. Proof-nets : the parallel syntax for proof-theory. In Aldo Ursini and Paolo Agliano, editors, *Logic and Algebra.* Marcel Dekker, New York, 1996.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[GS01] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag, 2001.

[GS02] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of multiplicative exponential linear logic. Technical Report WV-2002-03, Technische Universität Dresden, 2002. Submitted.

[Gue99] Stefano Guerrini. Correctness of multiplicative proof nets is linear. In *LICS*, pages 454–463, 1999.

[Gug02] Alessio Guglielmi. A system of interaction and structure. To appear in *ACM Transactions on Computational Logic*, 2002.

[Gug04a] Alessio Guglielmi. Formalism A. note, April 2004.

[Gug04b] Alessio Guglielmi. Formalism B. note, December 2004.

[HHS05] Robin Houston, Dominic Hughes, and Andrea Schalk. Modelling linear logic without units (preliminary results). Preprint, available at `http://arxiv.org/abs/math/0504037`, 2005.

[Hor06] Matthias Horbach. Proof nets for intuitionistic logic. Master's thesis, Universität des Saarlandes, 2006.

[Hug04] Dominic Hughes. Deep inderence proof theory equals categorical proof theory minus coherence. preprint, 2004.

[Hug05a] Dominic Hughes. Simple free star-autonomous categories and full coherence. Preprint, available at `http://arxiv.org/abs/math.CT/0506521`, 2005.

[Hug05b] Dominic Hughes. Simple multiplicative proof nets with units. Preprint, available at `http://arxiv.org/abs/math.CT/0507003`, 2005.

[HvG03] Dominic Hughes and Rob van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 1–10, 2003.

[Joi92] Jean-Baptiste Joinet. Une preuve combinatoire de forte normalisation (du fragment multiplicatif et exponentiel) des réseaux de preuves pour la logique linéaire (PN1). Technical Report 35, Équipe de Logique de Paris VII, 1992.

[Joi06] Jean-Baptiste Joinet. Completeness of mll proof nets w.r.t. weak distributivity, 2006. To appear in *Journal of Symbolic Logic*.

[KM71] Gregory Maxwell Kelly and Saunders Mac Lane. Coherence in closed categories. *Journal of Pure and Applied Algebra*, 1:97–140, 1971.

[Laf88] Yves Lafont. The linear abstract machine. *Theoretical Computer Science*, 59(1–2):157–180, 1988.

[Laf95] Yves Lafont. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 225–247. Cambridge University Press, 1995.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–169, 1958.

[Lam68] Joachim Lambek. Deductive systems and categories. I: Syntactic calculus and residuated categories. *Math. Systems Theory*, 2:287–318, 1968.

[Lam69] Joachim Lambek. Deductive systems and categories. II. standard constructions and closed categories. In P. Hilton, editor, *Category Theory, Homology Theory and Applications*, volume 86 of *Lecture Notes in Mathematics*, pages 76–122. Springer, 1969.

[Lam94] François Lamarche. Proof nets for intuitionistic linear logic I: Essential nets. Technical report, Imperial College, London, 1994.

[Lam95] François Lamarche. Games semantics for full propositional linear logic. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS 95)*, pages 464–473, 1995.

[Lam01] François Lamarche. On the algebra of structural contexts. Accepted at *Mathematical Structures in Computer Science*, 2001.

[Lam06] François Lamarche. Exploring the gap between linear and classical logic, 2006. Submitted.

[Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications (TLCA 1999)*, volume 1581 of *Lecture Notes in Computer Science*, pages 213–227. Springer-Verlag, 1999.

[Lau03] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003.

[LR96] François Lamarche and Christian Retoré. Proof nets for the Lambek-calculus — an overview. In V. Michele Abrusci and Claudia Casadio, editors, *Proceedings of the Third Roma Workshop "Proofs and Linguistic Categories"*, pages 241–262. CLUEB, Bologna, 1996.

[LS86] Joachim Lambek and Phil J. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1986.

[LS05a] François Lamarche and Lutz Straßburger. Constructing free Boolean categories. In *Proceedings of the Twentieth Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pages 209–218, 2005.

[LS05b] François Lamarche and Lutz Straßburger. Naming proofs in classical propositional logic. In Paweł Urzyczyn, editor, *Typed Lambda Calculi and Applications, TLCA 2005*, volume 3461 of *Lecture Notes in Computer Science*, pages 246–261. Springer-Verlag, 2005.

[LS06] François Lamarche and Lutz Straßburger. From proof nets to the free *-autonomous category. *Logical Methods in Computer Science*, 2(4:3):1–44, 2006.

[Mac63] Saunders Mac Lane. Natural associativity and commutativity. *Rice University Studies*, 49:28–46, 1963.

[Mac71] Saunders Mac Lane. *Categories for the Working Mathematician.* Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1971.

[McK05] Richard McKinley. Classical categories and deep inference. In *Structures and Deduction 2005 (Satellite Workshop of ICALP'05)*, 2005.

[Moo02] Richard Moot. *Proof Nets for Linguistic Analysis.* PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2002.

[Par92] Michel Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *Logic Programming and Automated Reasoning, LPAR 1992*, volume 624 of *LNAI*, pages 190–201. Springer-Verlag, 1992.

[Pra65] Dag Prawitz. *Natural Deduction, A Proof-Theoretical Study.* Almquist and Wiksell, 1965.

[Pra71] Dag Prawitz. Ideas and results in proof theory. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 235–307. North-Holland Publishing Co., 1971.

[Pui01] Quintijn Puite. *Sequents and Link Graphs.* PhD thesis, University of Utrecht, 2001.

[Ret93] Christian Retoré. *Réseaux et Séquents Ordonnés.* Thèse de Doctorat, spécialité mathématiques, Université Paris VII, February 1993.

[Ret97] Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Ph. de Groote and J. R. Hindley, editors, *Typed Lambda Calculus and Applications, TLCA'97*, volume 1210 of *Lecture Notes in Computer Science*, pages 300–318, 1997.

[Ret99a] Christian Retoré. Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs. Rapport de recherche 3652, INRIA, 1999. Appeared as [Ret03].

[Ret99b] Christian Retoré. Pomset logic as a calculus of directed cographs. In V. M. Abrusci and C. Casadio, editors, *Dynamic Perspectives in Logic and Linguistics*, pages 221–247. Bulzoni, Roma, 1999. Also available as INRIA Rapport de Recherche RR-3714.

[Ret03] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. *Theoretical Computer Science*, 294(3):473–488, 2003.

[Rob03] Edmund P. Robinson. Proof nets for classical logic. *Journal of Logic and Computation*, 13:777–797, 2003.

[See89] R.A.G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. *Contemporary Mathematics*, 92, 1989.

[Sel01] Peter Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11:207–260, 2001.

[SL04] Lutz Straßburger and François Lamarche. On proof nets for multiplicative linear logic with units. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic, CSL 2004*, volume 3210 of *LNCS*, pages 145–159. Springer-Verlag, 2004.

[SR98] Thomas Streicher and Bernhard Reus. Classical logic, continuation semantics and abstract machines. *Journal of Functional Programming*, 8(6):543–572, 1998.

[Str02] Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 388–402. Springer-Verlag, 2002.

[Str03a] Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003.

[Str03b] Lutz Straßburger. MELL in the Calculus of Structures. *Theoretical Computer Science*, 309(1–3):213–285, 2003.

[Str05a] Lutz Straßburger. From deep inference to proof nets. In *Structures and Deduction — The Quest for the Essence of Proofs (Satellite Workshop of ICALP 2005)*, 2005.

[Str05b] Lutz Straßburger. On the axiomatisation of Boolean categories with and without medial, 2005. Preprint, available at `http://arxiv.org/abs/cs.LO/0512086`.

[Thi97] Hayo Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh, 1997.

[Thi03] Rüdiger Thiele. Hilbert's twenty-fourth problem. *American Mathematical Monthly*, 110:1–24, 2003.

[Tiu01] Alwen Fernanto Tiu. Properties of a Logical System in the Calculus of Structures. Master's thesis, Technische Universität Dresden, 2001.

[Tiu06] Alwen Fernanto Tiu. A system of interaction and structure II: The need for deep inference. *Logical Methods in Computer Science*, 2(2):1–24, 2006.

[TS00] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, second edition, 2000.

[Yet90] David N. Yetter. Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic*, 55(1):41–64, 1990.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399