



HAL
open science

From Proof nets to the Free $*$ -Autonomous Categories

François Lamarche, Lutz Strassburger

► **To cite this version:**

François Lamarche, Lutz Strassburger. From Proof nets to the Free $*$ -Autonomous Categories. Logical Methods in Computer Science, 2006, 2 (4:3), 44 p. 10.2168/LMCS-2(4:3)2006 . inria-00099865v2

HAL Id: inria-00099865

<https://inria.hal.science/inria-00099865v2>

Submitted on 16 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FROM PROOF NETS TO THE FREE *-AUTONOMOUS CATEGORY

FRANÇOIS LAMARCHE^a AND LUTZ STRASSBURGER^b

^a LORIA & INRIA-Lorraine, Projet Calligramme, 615, rue du Jardin Botanique, 54602 Villers-lès-Nancy, France
e-mail address: lamarche@loria.fr

^b INRIA-Futurs, Projet Parsifal, École Polytechnique, Laboratoire d’Informatique (LIX), Rue de Saclay, 91128 Palaiseau Cedex, France
e-mail address: lutz@lix.polytechnique.fr

ABSTRACT. In the first part of this paper we present a theory of proof nets for full multiplicative linear logic, including the two units. It naturally extends the well-known theory of unit-free multiplicative proof nets. A linking is no longer a set of axiom links but a tree in which the axiom links are subtrees. These trees will be identified according to an equivalence relation based on a simple form of graph rewriting. We show the standard results of sequentialization and strong normalization of cut elimination. In the second part of the paper we show that the identifications enforced on proofs are such that the class of two-conclusion proof nets defines the free *-autonomous category.

INTRODUCTION

The interplay between logic and category theory is fascinating because it is rich, bidirectional and non-trivial. There is more to this non-triviality than the fact that

a proof of a statement like “the logical system \mathcal{S} corresponds to the set of categorical axioms \mathcal{T} ” is always a non-trivial task.

In addition there will very often be discrepancies between the abstract categorical axiomatization and the actual properties of the syntactical objects that are used by proof theorists. And if a denotational semantics is found for \mathcal{S} , it is more likely to follow the categorical directives than the syntactical ones. These discrepancies are the source of creative tensions.

For instance many logical constructions can be expressed in terms of adjunctions, and ordinary adjunctions give rise to two “triangular” equations, which can be called (very roughly) unit and co-unit. But syntactical considerations often give a real significance to one of them but not to the other. A standard example is the lambda calculus, where the

2000 ACM Subject Classification: F.4.1.

Key words and phrases: multiplicative linear logic, proof nets, *-autonomous categories, coherence problems.

^b This research has been carried out while the second author had an INRIA post-doc position at the LORIA in Nancy, France.

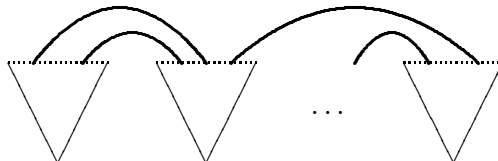
co-unit equation is β -reduction, and the unit one is the η -rule. Nobody would suggest that the latter is more important than the former, and proof theorists would most often rather not deal with the η -rule, because it makes normalization much harder, if not impossible. But it is not easy at all to construct a denotational semantics that does not obey the η -rule, although it can be done [Hay85].

As another example of tension, if a poset can be used to embody provability— $A \leq B$ means I can prove B if I assume A —replacing that poset by a category will allow us to *name* proofs, and to single one out by a map $f: A \rightarrow B$. But then composition of proofs (for syntacticians: cut-elimination) will have to be associative. This happens rather naturally with natural deduction systems, less so with the sequent calculus, where some quotienting has to be done. Thus category theory furnishes critical tools to test proof theory, a set of external ideals by which it can be judged. But if some categorical criterion is not obeyed by the syntax, this does not mean that syntax is automatically wrong. Perhaps it is the categorical formulation that needs to be refined. Tensions can be resolved in more than one way.

Naturally this idea of naming proofs “correctly” has been around long before categories were invented. For a long time logicians have been aware of the need to determine, given a formal system \mathcal{S} and two proofs of a formula A in that system, when these two proofs are, or name “the same” proof. As a matter of fact this was already a concern of Hilbert when he was preparing his famous lecture of 1900 [Thi03]. This problem has taken more importance during the last few years, because many logical systems permit a close correspondence between proofs and programs.

In a formalism like the sequent calculus (and to a lesser degree, natural deduction), it is oftentimes very easy to see that two derivations π_1 and π_2 should be identified because π_1 can be transformed in to π_2 by a sequence of rule permutations that are obviously trivial. It is less immediately clear *in general* what transformations can be effected on a proof without changing its essence. Here the categorical ideals are very helpful, providing criteria for the identification of proofs that are simple, general and unambiguous. But they are sometimes too strong, as happens [LS86, Gir91] for classical logic¹. . . another case of creative tension, which puts evolutionary pressure on both category theory and proof theory.

The advent of linear logic marked a significant advance in that quest for a good onomastics of proofs. In particular the multiplicative fragment of linear logic (MLL) comes equipped with an extremely successful theory of proof identification: not only do we know exactly when two sequent proofs should be identified (the allowed rule permutations are described in [Laf95]), but there is a class of simple formal objects that precisely represent these equivalence classes of sequent proofs. These objects are called proof nets, and they have a strong geometric character, corresponding to additional graph structure (“axiom links”) on the syntactical forest of the sequent. More precisely, given a sequent $\Gamma = A_1, \dots, A_n$ and a proof π of that sequent, then the proof net that represents π is simply given by the syntactical forest of Γ decorated with additional edges (shown in thick lines in the picture below) that represent the identity axioms that appeared in the proof:



¹perhaps it is better to say: the currently held conception of proofs in classical logic

Moreover proof nets are vindicated by category theory, since the category of two-formula sequents and proof nets is the free *-autonomous category [Bar79] (if we omit units from the definition of a *-autonomous category) on the set of generating atomic formulas. This first appeared in [Blu93], but the problem of precisely defining a *-autonomous category without units has given rise to recent developments [LS05a, HHS05, DP05].

As a matter of fact, axiom links were already visible, under the name of *Kelly-Mac Lane graphs* in the early work [KM71] that tried to describe free symmetric monoidal closed (also called *autonomous*) categories; Girard’s key insights [Gir87] here were in noticing that there was an inherent symmetry that could be formalized through a negation (thus the move from autonomous to *-autonomous), and that the addition of the axiom links to the sequent’s syntactic forest were enough to completely characterize the proof (if no units are present).

The theory of proof nets has been extended to larger fragments of linear logic. When judged from the point of view of their ability to identify proofs that should be identified, these extensions can be shown to have varying degrees of success. One of these extensions, which complies particularly well with the categorical ideal—and can be firmly put in the “successful” class even without appealing to categorical considerations—is the inclusion of additive connectives presented in [HvG03], in which the additives correspond exactly to categorical product and coproduct.

In this paper we give a theory of proof nets for the full multiplicative fragment, that is, including the multiplicative units. We then prove that it allows us to construct the free *-autonomous category with units on a given set of generating objects, thus getting full validation from the categorical imperative.

When this paper was submitted there were only two other treatments of multiplicative units that we were aware of. In [KO99], the authors provide an internal language for autonomous and *-autonomous categories based on the $\lambda\mu$ -calculus, and in [BCST96], several classes of free monoidal categories are constructed, by the means of a nonstandard version of two-sided proof nets, with a correctness criterion which is a version of the Danos contractibility criterion [Dan90]. Being based on the $\lambda\mu$ -calculus, the first of these papers is firmly in the tradition of natural deduction, in which the logical rules (introduction/elimination) mechanically generate the system of proof objects, which are ordinary terms with binders. Unsurprisingly, an equivalence relation on the terms is needed to construct the free *-autonomous categories. It is well-known that unless a calculus is “intuitionistic” (with one-conclusion sequents), it is not easy at all to establish a good correspondence between such systems of terms and the graphical proof objects that have become the tradition in linear logic; this is still research material.

As for the second of these papers, we think its approach is best summarized, despite its title, by the means of the sequent calculus. It starts with a core logic which is weaker than MLL, which can be related to it as follows: Given the usual primitives \otimes , \wp , 1 , and \perp of multiplicative linear logic, look at the following system of polarities, where \circ means “right side” and \bullet means “left side”:

$$\frac{\otimes}{\begin{array}{c|cc} \bullet & \circ & \\ \bullet & \bullet & \\ \circ & & \circ \end{array}} \quad \frac{\wp}{\begin{array}{c|cc} \bullet & \circ & \\ \bullet & \bullet & \\ \circ & & \circ \end{array}} \quad \frac{1}{\begin{array}{c} \bullet \\ \circ \end{array}} \quad \frac{\perp}{\begin{array}{c} \bullet \\ \circ \end{array}}$$

A constant may have either polarity, but you are only allowed to apply a tensor or a par on two formulas that have the same polarity, and the resulting formula has that same polarity. If we now add axioms of the form a°, a^\bullet then the main system in [BCST96] is

exactly equivalent to multiplicative linear logic with the usual rules, but where the introduction of connectives have to obey the polarity restrictions above. A polarized one-sided sequent of the form $\vdash A_1^\bullet, \dots, A_n^\bullet, B_1^\circ, \dots, B_m^\circ$ is translated back in the authors' notation as $A_n^\perp, \dots, A_1^\perp \vdash B_1, \dots, B_m$, where the $(-)^{\perp}$ operation here is the ordinary de Morgan dual, remembering that it inverts polarities. Thus it is indeed a weaker logic than classical multiplicative linear logic, since, for instance, ordinary axiom links always “straddle the left-right divide”. But there are no polarity restrictions on constants, and thus the constant-only fragment, suitably quotiented, should give back the free $*$ -autonomous category generated by the empty set. The addition of non-logical axioms allows the construction of the free such category generated by an ordinary category.

Later in [BCST96], a side-switching negation connective is introduced, as is also done in [Pui01], along with non-logical axioms for it, and the larger system is equivalent to classical multiplicative linear logic.

In the next section we will say how our approach to proof nets differs from the one which is used in [BCST96]. It should be obvious eventually that what we propose is considerably simpler. We have chosen to use only sets of objects (atomic variables) as generating sets. It would be easy to extend our work to construct the free $*$ -autonomous category generated by an arbitrary category, or an arbitrary structad [Lam01], but we are trying very hard to be read by both algebraists and proof theorists, and proof theorists are notoriously suspicious of non-logical axioms. In general they kill cut-elimination/normalization, but there are indeed classes of harmless non-logical axioms. Recently, general theoretical criteria [DW03] have been developed that allow the identification of such harmless classes.

Since this manuscript was submitted, yet another approach [Hug05a] has been proposed to the problem of constructing free $*$ -autonomous categories, which improves on its predecessors in that the cut-elimination process can be effected at the level of the *representatives* of the equivalence classes [Gir96b].

Outline of the paper. This paper consists of two parts. The first one is only concerned with syntax: the sequent calculus and the more modern syntax of proof nets, and our variation on ordinary multiplicative proof nets that permits the addition of units. The standard results—sequentialization and cut-elimination—are proved. The second part is concerned with algebra: after some introductory material on autonomous and $*$ -autonomous categories, we show that, given a set \mathcal{A} of atomic formulas, the set of proof nets constructed in the previous section do form a $*$ -autonomous category, which is easy, and then that it is actually the free one generated by \mathcal{A} , which is much harder. Both sections have discussion on history and motivation. We have taken pains to make the treatment as self contained as possible. All the proofs are complete; we have done the utmost to avoid any kind of hand-waving, and we tried hard to ensure that a reader with only a minimal background in multiplicative linear logic and/or category theory could read this.

The main results of this paper have already been presented at the CSL-conference 2004 in Karpacz, but it was impossible to give the complete story in fifteen pages [SL04].

1. PROOF NETS FOR MULTIPLICATIVE LINEAR LOGIC

We assume that the reader is familiar with the sequent calculus for classical multiplicative linear logic, and has some basic notions on proof nets, e.g., understands the idea of a correctness criterion. For an introduction, the reader is referred to [Laf95, Str06].

1.1. Why are the units problematic? The problem with the bottom rule is that it is very mobile. Suppose a proof contains a rule instance r which appears after a \perp -introduction rule and does not introduce a connective under that \perp . Then r can be pushed above that \perp -introduction:

$$\frac{\perp \frac{\Gamma}{\perp, \Gamma} \quad \dots}{r \frac{\perp, \Gamma'}{\perp, \Gamma'}} \quad \longleftrightarrow \quad r \frac{\Gamma \quad \dots}{\perp \frac{\Gamma'}{\perp, \Gamma'}}$$

It is very hard to find a good reason to decree that the difference between these two sequent proofs has some essential significance, which goes beyond mere notation, and that they ought to be distinguished. Asking for a distinction opens the door to a theory of proof identification which can only be slightly less bureaucratic than the sequent calculus itself. Moreover, the theory of *-autonomous categories tells us that they *should* be identified. But then accepting this seemingly trivial permutation as an equation has deep consequences. Supposing that rule r was a \otimes -introduction, there is now a choice of two branches on which to do the \perp -introduction.

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

Ordinary proof nets for multiplicative linear logic have successfully eliminated the bureaucracy introduced by the \otimes and \wp sequent rules. They are characterized by the presence of *links*, which connect the atoms of the syntactical forest of the sequent. When extending them to multiplicative units, the first impulse is probably to try to attach the \perp s that are present on the sequent forest on other atomic formulas. This corresponds to doing the \perp -introductions as early as possible, that is, as high up on the sequent tree as can be done. This approach has very recently been used in a most satisfying manner [Hug05b, Hug05a].

We see that an arbitrary choice has to be made because of tensor introductions: in a \otimes -introduction one of the two branches of the sequent proof tree has to be chosen for doing the \perp -introduction. In such a situation, correct identification of proofs can only be achieved by considering equivalence classes of graphs, and the theory of proof nets involves an equivalence relation on a set of “correct” graphs.

Another possibility is to attach the \perp s on *branches* of the proof forest. This is done in [BCST96], where an equivalence class is also used, built on the ability to slide the constants up and down the branches.

Yet another possibility is to attach the \perp s “as low as possible” on the forest, corresponding to the idea that in the sequent calculus deduction the \perp -introduction would be done as late as possible, for example just before the \perp instance gets a connective introduced under it. One way of implementing this is linking the \perp instance to the last connective that was introduced above it. This is not the only way of doing things, for example we could

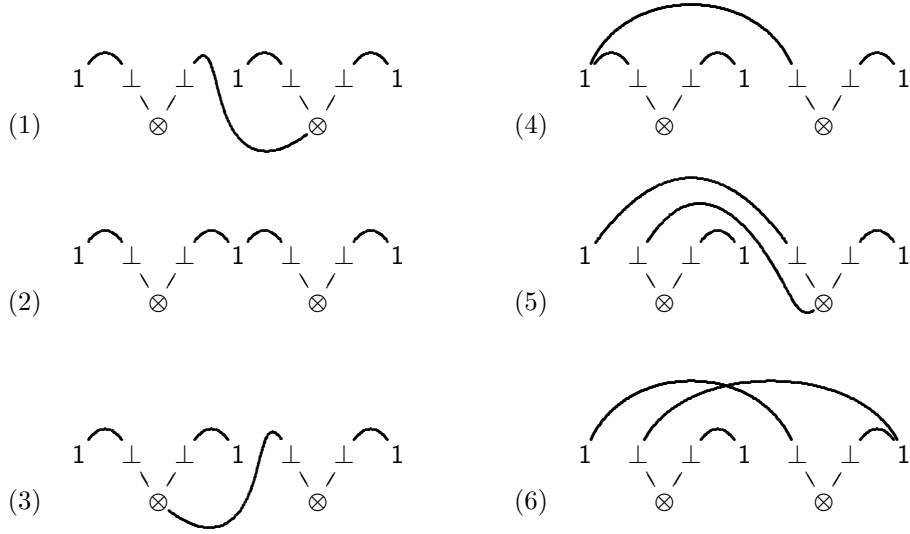


Figure 1: Different representations of the same proof

imagine links that attach that \perp instance to several subformulas of the sequent forest, corresponding to the several conclusions of the sequent that existed above the \perp -introduction.² But whatever way we choose to “normalize” proofs, we claim that if the conventional notion of “link” is used for \perp s,³ we still need to use equivalence classes of such graphs, and there is no hope of having a normal form in that universe of enriched sequent graphs. For instance, the six graphs in Figure 1 are easily seen to represent equivalent proofs, because going from an odd-numbered example to its successor is just sliding a \perp -intro up in one of the \otimes -intro branches, and going from an even-numbered example to its successor is just doing the reverse transformation. But notice that examples (3) and (5) are *distinct but isomorphic* graphs, since one can be exactly superposed on the other *by only using the Exchange rule*. Thus it is impossible, given the information at our disposal, to choose one instead of the other to represent the abstract proof they both denote. The only way this could be done would be by using arbitrary extra information, like the order of the formulas in the sequent, a strategy that only replaces the overdeterminism of the sequent calculus by another kind of overdeterminism.

The same can be said of Examples (2) and (6), which are also isomorphic modulo Exchange. But notice that these two comply to the “as early as possible” strategy, while the previous two were of the “as late as possible” kind. So for neither strategy can there be a hope a graphical normal form. The interested reader can verify that the six examples above are part of a “ring” of 24 graphs that are all equivalent from the point of view of category theory. Thus there are essentially only two proofs of that sequent, but 48 possible different graphs like these on it.

²Our approach is probably best seen as a version of this, where ordinary proof net technology is used to express this idea. We add extra “bunching” nodes, that act like ordinary pars from the point of view of correctness.

³I.e., if we consider a proof π on the sequent Γ as the sequent forest of Γ decorated with special edges that encode information about the essence of π .

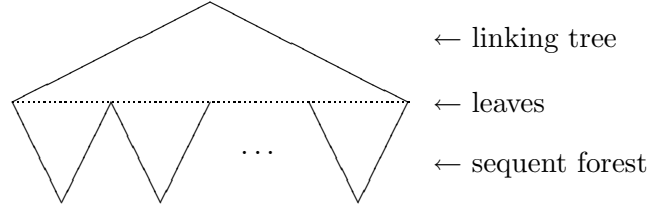


Figure 2: Linking tree and sequent forest sharing the same set of leaves

Thus there is one aspect of our work that does not differ from [BCST96], which is our presentation of abstract proofs as equivalence classes of graphs. But some related aspects are significantly different:

- The graphs that belong to our equivalence classes are *standard multiplicative proof nets*, where the usual notions, like correctness criteria and the empire of a tensor branch, will apply. It is just that some \wp and \otimes links are used in a particular fashion to deal with the units. (The readers can choose their favorite correctness criterion since they are all equivalent; in this paper we will use the one of [DR89] because of its popularity.)
- The equivalence relation we will present is based on a very simple set of rewriting rules on proof graphs. As a matter of fact, there is only *one* non-trivial rule, since the other rules have to do with commutativity and associativity of the connectives and can be dispensed with if we use, for example, n -ary connectives.

1.2. Cut free proof nets. Let $\mathcal{A} = \{a, b, \dots\}$ be an arbitrary set of atoms, and let $\mathcal{A}^\perp = \{a^\perp, b^\perp, \dots\}$. The set of MLL *formulas* is defined as follows:

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{A}^\perp \mid \mathbf{1} \mid \perp \mid \mathcal{F} \otimes \mathcal{F} \mid \mathcal{F} \wp \mathcal{F} \quad . \quad (1.1)$$

Additionally, we will define the set of MLL *linkings* (which can be seen as a special kind of formulas) as follows:

$$\mathcal{L} ::= \mathbf{1} \mid a \otimes a^\perp \mid a^\perp \otimes a \mid \perp \otimes \mathcal{L} \mid \mathcal{L} \otimes \perp \mid \mathcal{L} \wp \mathcal{L} \quad . \quad (1.2)$$

Here, a stands for any element of \mathcal{A} . We will use A, B, \dots to denote formulas, and P, Q, \dots to denote linkings. *Sequents* (denoted by Γ, Δ, \dots) are finite lists of formulas (separated by comma).

In the following, we will consider formulas and linkings always as binary trees (and sequents as forests), whose leaves are decorated by elements of $\mathcal{A} \cup \mathcal{A}^\perp \cup \{\mathbf{1}, \perp\}$, and whose inner nodes are decorated by \wp or \otimes . We can also think of the nodes being decorated by the whole subformula rooted at that node.

Definition 1.2.1. A *pre-proof graph* is a graph consisting of a linking P and a sequent Γ , such that the set of leaves of P coincides with the set of leaves of Γ (as depicted in Figure 2). It will be denoted by $P \triangleright \Gamma$ or by

$$\begin{array}{c} P \\ \nabla \\ \Gamma \end{array} .$$

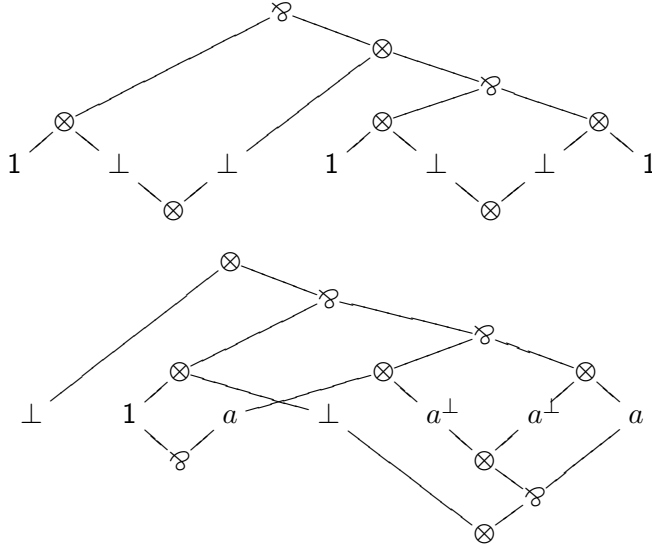


Figure 3: Two examples of proof graphs

Following the tradition, we will draw these graphs in such a way that the roots of the formula trees are at the bottom, the root of the linking tree is at the top, and the leaves are in between. Figure 3 shows two examples. The first of them corresponds to the first graph in Figure 1. In a more compact notation we will write them as

$$\begin{aligned} & (1_1 \otimes \perp_2) \wp (\perp_3 \otimes ((1_4 \otimes \perp_5) \wp (\perp_6 \otimes 1_7))) \\ & \quad \quad \quad \nabla \\ & 1_1, \perp_2 \otimes \perp_3, 1_4, \perp_5 \otimes \perp_6, 1_7 \end{aligned} \tag{1.3}$$

and

$$\begin{aligned} & \perp_1 \otimes ((1_2 \otimes \perp_4) \wp ((a_3 \otimes a_5^\perp) \wp (a_6^\perp \otimes a_7))) \\ & \quad \quad \quad \nabla \\ & \perp_1, 1_2 \wp a_3, \perp_4 \otimes ((a_5^\perp \otimes a_6^\perp) \wp a_7) \end{aligned} \tag{1.4}$$

Here, the indices are used to show how the leaves of the linking and the leaves of the sequent are identified. In this way we will, throughout this paper, use indices on atoms to distinguish between different occurrences of the same atom (i.e., a_3 and a_7 do not denote different elements of \mathcal{A}). In the same way, indices on the units 1 and \perp are used to distinguish different occurrences.

Remark 1.2.2. Since we are dealing with sets-and-structure, we should mention an additional bit of structure that pre-proof-graphs possess, which is a total ordering on the set of leaves, corresponding to the syntactic order in which formulas and sequents are written. Since this order is completely obvious in the notation, we will not mention it again. Thus we take the most standard approach to the sequent calculus in commutative logic, in which a sequent is a *sequence* of formulas, subject to the permutations that are induced by the Exchange rule.

Definition 1.2.3. A *switching* of a pre-proof graph $P \triangleright \Gamma$ is a graph G that is obtained from $P \triangleright \Gamma$ by omitting for each \wp -node one of the two edges that connect the node to its children. [DR89]

$$\begin{array}{ccc}
\text{id} \frac{}{a \otimes a^\perp \triangleright a, a^\perp} & & \text{ex} \frac{P \triangleright \Gamma, A, B, \Delta}{P \triangleright \Gamma, B, A, \Delta} \\
\\
1 \frac{}{1 \triangleright 1} & & \perp \frac{P \triangleright \Gamma}{\perp \otimes P \triangleright \perp, \Gamma} \\
\\
\wp \frac{P \triangleright A, B, \Gamma}{P \triangleright A \wp B, \Gamma} & & \otimes \frac{P \triangleright \Gamma, A \quad Q \triangleright B, \Delta}{P \wp Q \triangleright \Gamma, A \otimes B, \Delta}
\end{array}$$

Figure 4: Translation of cut free sequent calculus proofs into pre-proof graphs

Lemma 1.2.7 ensures that the notion of proof net is well-defined, in the sense that all its members are proof graphs, i.e., correct.

An alternative approach to the definition of a proof net would be to restrict the definition of \sim to proof graphs and not mention pre-proof graphs at all, i.e., to assume from the start that everything obeys the correctness criterion. This slight breach with tradition has the advantage of not requiring the side condition, which is asymmetrical. Such a point of view is used systematically in [Hug05a].

1.3. Sequentialization. In this section we will relate our proof nets to cut free sequent calculus proofs of MLL. For this, we will first show, how cut free sequent proofs of MLL can be inductively translated into pre-proof graphs. This is done in Figure 4. We will call a pre-proof net *sequentializable* if one of its representatives can be obtained from a sequent calculus proof via this translation.

Theorem 1.3.1. *A pre-proof net is sequentializable if and only if it is a proof net.*

For the proof we will need the observation that any proof graph is an ordinary proof net (in the sense of [DR89]), and the well-known fact that there is always a splitting tensor in such a net.

1.3.2. Ordinary proof nets. An *ordinary axiom linking* for a sequent Γ is a perfect matching of the leaves of Γ (i.e., the atoms and units) such that only dual atoms or units are matched. Of course, there are sequents for which no ordinary axiom linking exists, for example $a, b^\perp \wp c, a^\perp \otimes c$, and there are sequents with more than one possible ordinary axiom linking, for example $a \wp a, a^\perp \otimes a^\perp$. An *ordinary pre-proof net* is a sequent Γ equipped with an ordinary axiom linking. Switchings and correctness are defined as in Definitions 1.2.3 and 1.2.4. An *ordinary proof net* is a correct ordinary pre-proof net. In other words, what we call ordinary proof nets are the nets that are thoroughly studied in the literature, with the only difference that we also allow 1 and \perp at the places of the leaves.

Observation 1.3.3. Every pre-proof graph $P \triangleright \Gamma$ is an ordinary pre-proof net. To make this precise, define for the linking P the *linking formula* P^* inductively as follows:

$$\begin{array}{ccc}
a^{\perp*} = a & 1^* = \perp & (A \otimes B)^* = B^* \otimes A^* \\
a^* = a^\perp & \perp^* = 1 & (A \wp B)^* = B^* \wp A^*
\end{array} \quad .$$

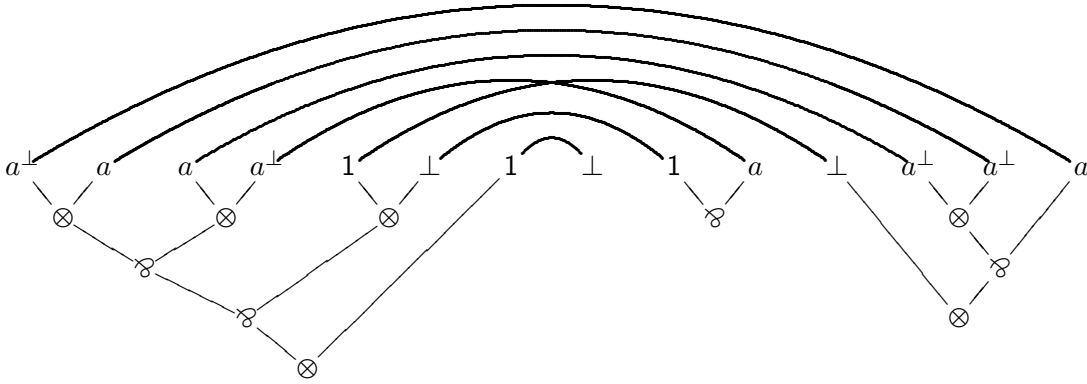


Figure 5: Example of an ordinary proof net

In other words, P^* is obtained from P by first replacing each leaf by its dual and by leaving all inner nodes unchanged, and then taking the mirror image of the tree⁵. We now connect the leaves of P^* and Γ by ordinary axiom links according to the leaf identification in $P \triangleright \Gamma$. Here we forget the fact that \perp and 1 are the units and think of them as ordinary dual atoms. We get an ordinary pre-proof net, which we will denote by $\pi_{P \triangleright \Gamma}$. Figure 5 shows as example the ordinary proof net obtained from the second proof graph in Figure 3.⁶ Observe that $\pi_{P \triangleright \Gamma}$ is correct if and only if $P \triangleright \Gamma$ is correct.

Lemma 1.3.4. *If in a ordinary proof net all roots are \otimes -nodes, then one of them is splitting, i.e., by removing it the net becomes disconnected.* [Gir87]

There are several proofs available for this lemma—one example is Girard’s original paper [Gir87], and another (very instructive) paper is Retoré’s [Ret03]. For this reason we do not show the proof here and concentrate on the

Proof of Theorem 1.3.1. It is easy to see that the rules 1 and id give proof graphs and that the rules \perp , \wp , and \otimes preserve the correctness. Therefore every sequentializable pre-proof net is correct.

For the other direction pick one representative $P \triangleright \Gamma$ of the proof net and proceed by induction on the sum of the number of \otimes -nodes in the graph and the number of \wp -nodes in Γ . In other words, the number of \wp -nodes in P is not relevant. (We will end up by exhibiting a sequentialization of an equivalent proof graph $Q \triangleright \Gamma$, obtained from $P \triangleright \Gamma$ by only applying associativity and commutativity of \wp , i.e., the first two equations in 1.2.5.)

The base case is trivial (the graph consists of a single node which is labeled by 1). For the inductive case look at the root nodes in Γ . If one of them is a \wp , we can remove it by applying the \wp -rule and proceed by induction hypothesis. If all roots in Γ are \otimes nodes, we interpret $P \triangleright \Gamma$ as an ordinary ordinary proof net (according to Observation 1.3.3), which remains correct if we remove in P all \wp -nodes that do not have a \otimes -node as ancestor. Now

⁵Since we are dealing only with the commutative case, taking the mirror image is unnecessary. However it simplifies many of the diagrams, avoiding unnecessary crossings, and this as much for proof nets as for commutative diagrams.

⁶If Γ consists of only one formula, then we have an object which is in [BC99] called a *bipartite proof net*. In fact, two proof graphs (in our sense) are equivalent if and only if the two linkings (seen as formulas) are isomorphic (in the sense of [BC99]).

all roots are \otimes -nodes and one of them is splitting (by Lemma 1.3.4). If it belongs to Γ , we restore the \wp -structure above the \otimes -nodes in P such that each of the two subtrees of the root- \wp covers one of the two components in which the graph is divided by removing the splitting \otimes . We can now apply the \otimes -rule and proceed by induction hypothesis on the two premises. If the splitting \otimes belongs to P , there are two possibilities. Either it comes from an axiom link (i.e., both children are dual atoms), or it comes from a bottom link (i.e., one child is a \perp). In the first case, we have that the graph is of the shape $P', a_i \otimes a_j^\perp, P'' \triangleright \Gamma'\{a_i\}, \Gamma''\{a_j^\perp\}$, where the linking is written as sequent because the \wp -roots are removed, and where $\Gamma'\{a_i\}$ denotes a sequent where one formula contains the atom a , indexed as a_i , such that P', a_i and $\Gamma'\{a_i\}$ share the same atoms and units, as well as a_j^\perp, P'' and $\Gamma''\{a_j^\perp\}$. If we replace a_i by 1_i and a_j^\perp by 1_j , we obtain two proof graphs $P', 1_i \triangleright \Gamma'\{1_i\}$ and $1_j, P'' \triangleright \Gamma''\{1_j\}$ of strictly smaller size. Therefore, by induction hypothesis, we have two sequent proofs

$$\begin{array}{ccc} \begin{array}{c} \overline{1} \\ 1_i \\ \pi_1 \\ \Gamma'\{1_i\} \end{array} & \text{and} & \begin{array}{c} \overline{1} \\ 1_j \\ \pi_2 \\ \Gamma''\{1_j\} \end{array} \end{array} .$$

From π_1 and π_2 we can construct the following sequent proof:

$$\begin{array}{c} \text{id} \overline{\quad} \\ a_i, a_j^\perp \\ \pi'_2 \\ a_i, \Gamma''\{a_j^\perp\} \\ \pi'_1 \\ \Gamma'\{a_i\}, \Gamma''\{a_j^\perp\} \end{array} ,$$

where π'_1 is obtained from π_1 by replacing 1_i everywhere by a_i and by adding $\Gamma''\{a_j^\perp\}$ everywhere to the sequent that contains the a_i . Similarly π'_2 is obtained from π_2 . It is easy to see that this proof translates into $[P \triangleright \Gamma]$. The case where the splitting tensor in P has a \perp as child is similar and left to the reader. \square

Remark 1.3.5. It is well known that ordinary proof nets also have a sequentialization theorem, i.e., they are correct if and only they are obtained from a (unit-free) sequent calculus proof in the obvious way. This has been studied thoroughly in the literature (e.g., [Gir87, DR89, Ret03]).

1.4. Proof nets with cuts. In this section we will introduce cuts in our proof nets. A *cut* is a formula $A \oplus A^\perp$, where \oplus is called the *cut connective*, and where the function $(-)^{\perp}$ is defined on formulas as follows (with an obvious abuse of notation):

$$\begin{array}{lll} a^{\perp\perp} = a & 1^\perp = \perp & (A \otimes B)^\perp = B^\perp \wp A^\perp \\ a^\perp = a^\perp & \perp^\perp = 1 & (A \wp B)^\perp = B^\perp \otimes A^\perp \end{array} \quad (1.5)$$

Notice that we invert the order under a negation, as if the logic were not commutative. This considerably simplifies many proof nets and categorical diagrams.

A *sequent with cuts* is a sequent where some of the formulas are cuts. But cuts are not allowed to occur inside formulas, i.e., all \oplus -nodes are roots. A *pre-proof graph with cuts* is a pre-proof graph $P \triangleright \Gamma$, where Γ may contain cuts. The \oplus -nodes have the same geometric behavior as the \otimes -nodes. Therefore the correctness criterion stays literally the same, and we can define *proof graphs with cuts* and *proof nets with cuts* accordingly. In the translation from sequent proofs containing the cut rule into pre-proof graphs with cuts, the cut is treated as follows:

$$\text{cut} \frac{\Gamma, A \quad A^\perp, \Delta}{\Gamma, \Delta} \quad \rightsquigarrow \quad \text{cut} \frac{P \triangleright \Gamma, A \quad Q \triangleright A^\perp, \Delta}{P \wp Q \triangleright \Gamma, A \oplus A^\perp, \Delta} .$$

Since the \oplus behaves in the same way as the \otimes , we immediately have the generalization of the sequentialization theorem:

Theorem 1.4.1. *A pre-proof net with cuts is sequentializable if and only if it is correct, i.e., it is a proof net with cuts.*

Proof. This proof is literally the same as the proof of Theorem 1.3.1, with the only difference, that there are now also \oplus -nodes, which are treated as \otimes -nodes. \square

Remark 1.4.2. In the same way, we can add the cut to ordinary proof nets, as defined in 1.3.2. Of course, this does not affect the sequentialization.

1.5. Cut elimination. The famous cut elimination theorem says that for any proof containing cuts there is a cut-free proof of the same conclusion. For MLL sequent calculus proofs this is a well-known fact. Since we have sequentialization for cut-free proof nets, as well as for proof nets with cuts (Theorems 1.3.1 and 1.4.1), we can immediately conclude a cut elimination result for proof nets.

In this section we will present a procedure that will eliminate the cuts directly on the proof nets. More precisely, we will present a strongly normalizing cut reduction relation. This means that to every proof net with cuts a unique cut free proof net is assigned.

On the set of cut pre-proof graphs we can define the cut reduction relation \rightarrow as follows:

$$\begin{array}{ccc} \frac{P}{\nabla} & \rightarrow & \frac{P}{\nabla} \\ (A \wp B) \oplus (B^\perp \otimes A^\perp), \Gamma & & A \oplus A^\perp, B \oplus B^\perp, \Gamma \\ \\ \frac{P\{(a_h^\perp \otimes a_i) \wp (a_j^\perp \otimes a_k)\}}{\nabla} & \rightarrow & \frac{P\{a_h^\perp \otimes a_k\}}{\nabla} \\ a_i \oplus a_j^\perp, \Gamma & & \Gamma \\ \\ \frac{P\{(Q \otimes \perp_i) \wp 1_j\}}{\nabla} & \rightarrow & \frac{P\{Q\}}{\nabla} \\ \perp_i \oplus 1_j, \Gamma & & \Gamma \end{array}$$

These reduction steps are shown in graphical notation in Figure 6.

We have the following immediate lemma, which ensures that correctness is preserved during the reduction.

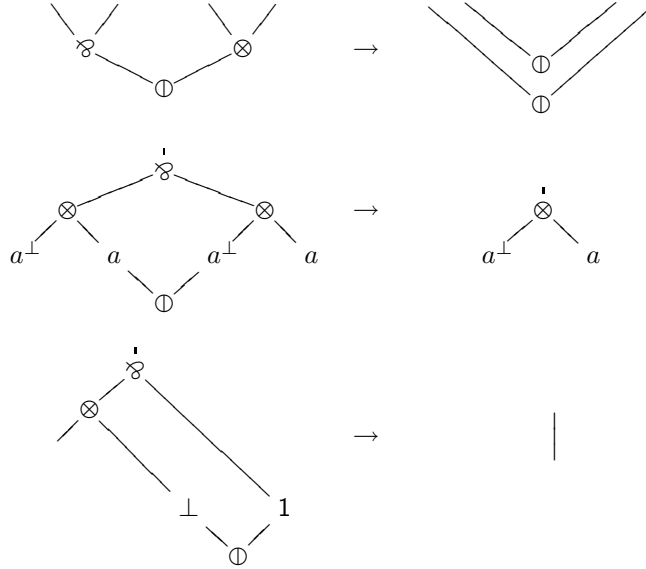
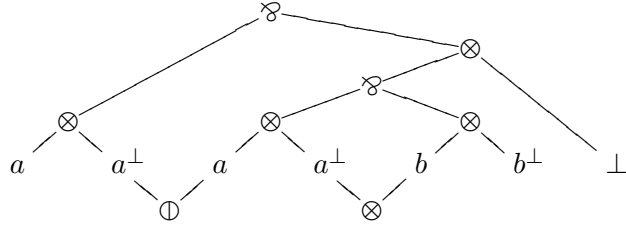


Figure 6: Cut elimination reduction steps

Lemma 1.5.1. *If $P \triangleright \Gamma$ is correct and $P \triangleright \Gamma \rightarrow P' \triangleright \Gamma'$, then $P' \triangleright \Gamma'$ is also correct.*

Proof. It is impossible that a cut reduction step introduce a cycle in a switching or make it disconnected. \square

Observe that it can happen that in a proof graph no reduction is possible, although there are cuts present in the sequent. For example, in



the cut cannot be reduced.

In a given proof graph $P \triangleright \Gamma$, a \oplus -node that can be reduced will be called *ready*. Obviously, a cut on a \otimes - \wp -pair is always ready, but for a cut on atoms or units this is not necessarily the case, as the example above shows. However, we have the following theorem.

Theorem 1.5.2. *Given a proof graph $P \triangleright \Gamma$ and a \oplus -node in Γ , there is an equivalent proof graph $P' \triangleright \Gamma$, in which that \oplus -node is ready, i.e., can be reduced.*

This is an immediate consequence of the following two lemmas.

Lemma 1.5.3. *For every proof graph $P \triangleright a_i \oplus a_j^\perp, \Gamma$ that contains an atomic cut, there is an equivalent proof graph $P' \{(a_h^\perp \otimes a_i) \wp (a_j^\perp \otimes a_k)\} \triangleright a_i \oplus a_j^\perp, \Gamma$.*

Lemma 1.5.4. *For every proof graph $P \triangleright \perp_i \oplus 1_j, \Gamma$ that contains a cut on the units, there is an equivalent proof graph $P' \{(Q \otimes \perp_i) \wp 1_j\} \triangleright \perp_i \oplus 1_j, \Gamma$.*

For proving them, we will use the following three lemmas (1.5.5–1.5.7).

Lemma 1.5.5. *Let $P\{(\perp_k \otimes R\{x_i\}) \wp (S\{x_j^\perp\} \otimes \perp_h)\} \triangleright x_i \oplus x_j^\perp, \Gamma$ be a proof graph, where x is an arbitrary atom or a unit, and x^\perp its dual.*

Then at least one of $P\{\perp_k \otimes (R\{x_i\} \wp (S\{x_j^\perp\} \otimes \perp_h))\} \triangleright x_i \oplus x_j^\perp, \Gamma$ and $P\{((\perp_k \otimes R\{x_i\}) \wp S\{x_j^\perp\}) \otimes \perp_h\} \triangleright x_i \oplus x_j^\perp, \Gamma$ is equivalent to it.

Proof. By way of contradiction, assume that both are not equivalent to the original proof graph. This means that in both cases the side condition (*) of 1.2.5 is not fulfilled, which means that in the original proof graph we have

- an extended switching wrt. \perp_k such that one node of $S\{x_j^\perp\} \otimes \perp_h$ is connected to it, and
- an extended switching wrt. \perp_h such that one node of $\perp_k \otimes R\{x_i\}$ is connected to it.

Without loss of generality, we can assume that in both extended switching the \perp_k (respectively \perp_h) is connected to the \otimes -root of $S\{x_j^\perp\} \otimes \perp_h$ (respectively $\perp_k \otimes R\{x_i\}$). If the two paths do not have a common node, then we can immediately construct a (normal, non-extended) switching in which both are present. But this switching contains a cycle in which the two paths are connected by the \otimes -roots of $\perp_k \otimes R\{x_i\}$ and $S\{x_j^\perp\} \otimes \perp_h$, contradicting the assumption of correctness. If the two paths have at least one common node, we can also construct a switching with a cycle as follows. We can make sure that it contains the first path between \perp_k and the first intersection node with the second path. Note that is path must go “downwards” from the \perp_k because the edge between the \perp_k and the root of $\perp_k \otimes R\{x_i\}$ is not present in the extended switching. Then the next node of the first path does also belong to the second path (otherwise we would have a graph node with four edges attached to it). We can now make sure that that part of the second path, which is determined by the direction established by these two nodes, is contained in the switching. There are now two possibilities:

- We get a path between \perp_k and the \otimes -root of $\perp_k \otimes R\{x_i\}$, which yields a cycle immediately because now the edge between the \perp_k and its \otimes -parent is present.
- We get a path between \perp_k and \perp_h . There are two subcases:
 - The path does not contain nodes from $S\{x_j^\perp\}$. In this case we can extend the path to a cycle using the two \otimes -roots of $\perp_k \otimes R\{x_i\}$ and $S\{x_j^\perp\} \otimes \perp_h$, as well as the \oplus -node between x_i and x_j^\perp .
 - The path does contain nodes from $S\{x_j^\perp\}$. We consider the last part of this path which connects a leaf of $S\{x_j^\perp\}$ with \perp_h , without touching any other node of $S\{x_j^\perp\}$, and which does not contain the \otimes -root of $S\{x_j^\perp\} \otimes \perp_h$ because the edge between \perp_h and its \otimes -parent is not present in the extended switching. This path can now be extended to a cycle that contains \otimes -parent of \perp_h .

This contradicts the assumption of correctness of the original graph. \square

Lemma 1.5.6. *Let $P\{(\perp_k \otimes R\{x_i\}) \wp (x_j^\perp \otimes Q)\} \triangleright x_i \oplus x_j^\perp, \Gamma$ be a proof graph, where x is an arbitrary atom or a unit, and x^\perp its dual.*

Then $P\{\perp_k \otimes (R\{x_i\} \wp (x_j^\perp \otimes Q))\} \triangleright x_i \oplus x_j^\perp, \Gamma$ is equivalent to it.

Proof. By way of contradiction, assume this is not the case, i.e., the side condition (*) of 1.2.5 is not fulfilled, which means that we have in the original proof graph an extended switching wrt. \perp_k such that a node of $x_j^\perp \otimes Q$ is connected to it. If this path goes through $R\{x_i\}$, we have a cycle immediately. If this is not the case, it has to enter $x_j^\perp \otimes Q$ either from

the \wp -node above or through a leaf of Q . In both cases we can extend the path through x_j^\perp and the \oplus -node to x_i and the root of $R\{x_i\}$, which yields a cycle. \square

Lemma 1.5.7. *Let $P\{(\perp_k \otimes R\{x_i\}) \wp x_j^\perp\} \triangleright x_i \oplus x_j^\perp, \Gamma$ be a proof graph, where x is an arbitrary atom or a unit, and x^\perp its dual.⁷ Then $P\{\perp_k \otimes (R\{x_i\} \wp x_j^\perp)\} \triangleright x_i \oplus x_j^\perp, \Gamma$ is equivalent to it.*

Proof. Similar to Lemma 1.5.6. \square

We can now proceed with the proofs of Lemmas 1.5.3 and 1.5.4.

Proof of Lemma 1.5.3. By the definition of proof graph, the linking P must contain two subtrees $a_h^\perp \otimes a_i$ and $a_j^\perp \otimes a_k$. By the correctness criterion, they must be in a \wp -relation, i.e., $P = P''\{R\{a_h^\perp \otimes a_i\} \wp S\{a_j^\perp \otimes a_k\}\}$ for some contexts $P''\{ \}$, $R\{ \}$ and $S\{ \}$. We will proceed by induction on the size of $R\{ \}$ and $S\{ \}$, i.e., the sum of the number of \wp - and \otimes -nodes in them. We have the following cases:

- Both are empty. In this case we are done.
- $R\{ \}$ has a \otimes as root and $S\{ \}$ is empty. In this case $R\{ \} = \perp \otimes R'\{ \}$ for some $R'\{ \}$, and we can apply Lemma 1.5.6 with $Q = a_k$.
- $R\{ \}$ is empty and $S\{ \}$ has a \otimes as root. This case is symmetrical to the previous one, and we can apply Lemma 1.5.6 with $Q = a_h^\perp$.
- Both $R\{ \}$ and $S\{ \}$ have a \otimes as root. In this case, we can apply Lemma 1.5.5, and proceed by induction hypothesis.
- One of $R\{ \}$ and $S\{ \}$ has a \wp as root. In this case we apply the associativity of the \wp (which is not subject to a side condition), and proceed by induction hypothesis. \square

Proof of Lemma 1.5.4. This proof is very similar to the previous one. Since $P \triangleright \perp_i \oplus 1_j, \Gamma$ is correct, it is of the shape $P''\{R\{Q \otimes \perp_i\} \wp S\{1_j\}\} \triangleright \perp_i \oplus 1_j, \Gamma$. Again, we will proceed by induction on the size of $R\{ \}$ and $S\{ \}$, with almost identical cases. The only difference is:

- $R\{ \}$ has a \otimes as root and $S\{ \}$ is empty. In this case we apply Lemma 1.5.7 (instead of Lemma 1.5.6). \square

This completes the proof of Theorem 1.5.2.

Let us now extend the relation \rightarrow to proof nets as follows: We say

$$[P \triangleright \Gamma] \rightarrow [Q \triangleright \Delta]$$

if and only if there are proof graphs $P' \triangleright \Gamma$ and $Q' \triangleright \Delta$ such that

$$P \triangleright \Gamma \sim P' \triangleright \Gamma \rightarrow Q' \triangleright \Delta \sim Q \triangleright \Delta \quad .$$

Let us first show that this is well-defined, in the sense that if the same cut is reduced in two different representatives of the same net, then the two results do also represent the same net.

Lemma 1.5.8. *Let $P \triangleright \Gamma \sim P' \triangleright \Gamma$, and let $P \triangleright \Gamma \rightarrow Q \triangleright \Delta$ and $P' \triangleright \Gamma \rightarrow Q' \triangleright \Delta$, i.e., in both reductions the same cut is reduced. Then we have $Q \triangleright \Delta \sim Q' \triangleright \Delta$.*

⁷In fact, according to the definition of proof graph, the only possibility here is that $x = \perp$ and $x^\perp = 1$. However, in order to emphasize a certain uniformity in all three lemmas, we use x .

Proof. Since $P \triangleright \Gamma \sim P' \triangleright \Gamma$, we have

$$P \triangleright \Gamma = P_0 \triangleright \Gamma \sim P_1 \triangleright \Gamma \sim \dots \sim P_n \triangleright \Gamma = P' \triangleright \Gamma,$$

for some linkings P_0, P_1, \dots, P_n , where for each $i = 1, \dots, n$ the equivalence $P_{i-1} \triangleright \Gamma \sim P_i \triangleright \Gamma$ is a direct application of the equations in 1.2.5. We can now distinguish three cases.

First, the reduced cut is on binary connectives. Then in each of the proof graphs $P_i \triangleright \Gamma$ the cut is ready and we have $Q \triangleright \Delta = Q_0 \triangleright \Delta \sim \dots \sim Q_n \triangleright \Delta = Q' \triangleright \Delta$, where each $Q_i \triangleright \Delta$ is obtained from reducing the cut in $P_i \triangleright \Gamma$.

In the second case the reduced cut is an atomic one, say $a_i \oplus a_j^\perp$. Here it might happen that in some of the $P_i \triangleright \Gamma$ the cut is not ready because of unnecessary applications of associativity. But it is easy to see that there is a transformation from $P \triangleright \Gamma$ to $P' \triangleright \Gamma$ in which the readiness of the cut is not destroyed, i.e., the sublinking $(a_h^\perp \otimes a_i) \wp (a_j^\perp \otimes a_k)$ of P and P' is not touched. We can therefore proceed as in the first case.

The most difficult case occurs if the cut is on the units, say $\perp_i \oplus 1_j$. Although $P = R\{(S \otimes \perp_i) \wp 1_j\}$ and $P' = R'\{(S' \otimes \perp_i) \wp 1_j\}$, the sublinking $(- \otimes \perp_i) \wp 1_j$ might be destroyed in the transformation because other subtrees might leave or enter the scope of the \perp_i , and can therefore occur “between” \perp_i and 1_j . However, in the reduction \perp_i and 1_j disappear. Hence these intermediate steps become vacuous. We can therefore proceed similarly to the other two cases. \square

The next thing to check is termination:

Lemma 1.5.9. *There is no infinite sequence*

$$[P \triangleright \Gamma] \rightarrow [P' \triangleright \Gamma'] \rightarrow [P'' \triangleright \Gamma''] \rightarrow \dots$$

Proof. In each reduction step the size of the sequent (i.e., the number of \wp , \otimes and \oplus -nodes) is reduced. \square

For showing confluence of the reduction relation, we will proceed in two steps. First we will show that the reduction relation on (pre-)proof graphs is confluent, and then we will extend the result to (pre-)proof nets, by employing Lemma 1.5.8.

Lemma 1.5.10. *If $Q \triangleright \Delta \leftarrow P \triangleright \Gamma \rightarrow R \triangleright \Sigma$, then either $Q \triangleright \Delta = R \triangleright \Sigma$, or there is a proof graph $S \triangleright \Phi$ such that $Q \triangleright \Delta \rightarrow S \triangleright \Phi \leftarrow R \triangleright \Sigma$.*

Proof. If $Q \triangleright \Delta$ and $R \triangleright \Sigma$ are obtained from $P \triangleright \Gamma$ by reducing the same \oplus -node in Γ then they must be equal. If different \oplus -nodes have been reduced in the two reductions, then both \oplus -nodes must have been ready in $P \triangleright \Gamma$. But reducing one of the two \oplus -nodes does not destroy the readiness of the other, which can therefore be reduced afterwards. Since the redexes of the reductions cannot “overlap”, the result $S \triangleright \Phi$ is independent of the order of the two reductions. \square

Lemma 1.5.11. *If $[Q \triangleright \Delta] \leftarrow [P \triangleright \Gamma] \rightarrow [R \triangleright \Sigma]$, then either $[Q \triangleright \Delta] = [R \triangleright \Sigma]$, or there is a proof net $[S \triangleright \Phi]$ such that $[Q \triangleright \Delta] \rightarrow [S \triangleright \Phi] \leftarrow [R \triangleright \Sigma]$.*

Proof. The problem is that the two reduction might take place in two different presentations of the proof net $[P \triangleright \Gamma]$. (Otherwise we could immediately apply Lemma 1.5.10.) The main idea of this proof is therefore to exhibit a presentation of $[P \triangleright \Gamma]$ in which both cuts are ready, and then apply Lemma 1.5.10 together with Lemma 1.5.8. Let \oplus_1 denote the cut that is reduced in Γ to obtain Δ and \oplus_2 the one that is reduced to obtain Σ . If they are identical, we immediately have (by Lemma 1.5.8) that $[Q \triangleright \Delta] = [R \triangleright \Sigma]$. If not, we distinguish the following cases:

- One of the two cuts is on binary connectives, i.e., it is ready in each presentation of the proof net. We can therefore choose a presentation in which the other cut is also ready and apply Lemma 1.5.10 and Lemma 1.5.8.
- One of the two cuts is on units, say \oplus_1 . Then we can first make \oplus_2 ready by applying Lemma 1.5.3 or Lemma 1.5.4. Then we apply Lemma 1.5.4 to also make \oplus_1 ready. This does not affect the readiness of \oplus_2 . We can therefore obtain a presentation of $[P \triangleright \Gamma]$ in which both cuts are ready, and proceed as in the previous case.
- Both cuts are atomic, but are not directly connected to each other via a “real” axiom link. Then we can proceed as in the previous case to obtain a presentation of $[P \triangleright \Gamma]$ in which both cuts are ready.
- Both cuts are atomic and share a common “real” axiom link. In other words, $P \triangleright \Gamma$ is of the following shape:

$$P' \{ (P'' \{ a_h^\perp \otimes a_i \} \wp P''' \{ a_j^\perp \otimes a_k \} \wp P'''' \{ a_l^\perp \otimes a_m \}) \}$$

$$\quad \quad \quad \nabla$$

$$a_i \oplus_1 a_j^\perp, a_k \oplus_2 a_l^\perp, \Phi$$

In this case it is not possible to make both cuts ready at the same time. But we can transform the above graph into

$$S' \{ ((a_h^\perp \otimes a_i) \wp (a_j^\perp \otimes a_k)) \wp (a_l^\perp \otimes a_m) \}$$

$$\quad \quad \quad \nabla$$

$$a_i \oplus_1 a_j^\perp, a_k \oplus_2 a_l^\perp, \Phi \quad ,$$

as well as into

$$S' \{ (a_h^\perp \otimes a_i) \wp ((a_j^\perp \otimes a_k) \wp (a_l^\perp \otimes a_m)) \}$$

$$\quad \quad \quad \nabla$$

$$a_i \oplus_1 a_j^\perp, a_k \oplus_2 a_l^\perp, \Phi \quad .$$

In the first case \oplus_1 is ready and in the second \oplus_2 . In both cases, after the reduction of one cut, the other becomes ready. After the second reduction, the result is in both cases $S' \{ a_h^\perp \otimes a_m \} \triangleright \Phi$. \square

Theorem 1.5.12. *The cut elimination reduction \rightarrow on proof nets is strongly normalizing. The normal forms are cut free proof nets.*

Proof. Termination is provided by Lemma 1.5.9, confluence follows from Lemma 1.5.11, and that the normal form is cut free is ensured by Theorem 1.5.2. \square

1.5.13. Cut elimination for ordinary proof nets. The following is very well known (see e.g., [Gir87, DR89, Ret03]), but we add it for the sake of completeness. Define the cut reduction relation on the set of ordinary pre-proof nets as shown in Figure 7. There are only two cases: the cut on binary connectives and the cut on atoms. A cut on binary connectives is replaced by two cuts on the corresponding subformulas (as in the case with units), and a cut on atoms is removed by melting the two attached axiom links to a single axiom link. It is easy to see that this reduction preserves correctness, and is terminating and confluent. Therefore Theorem 1.5.12 does also hold for ordinary proof nets.

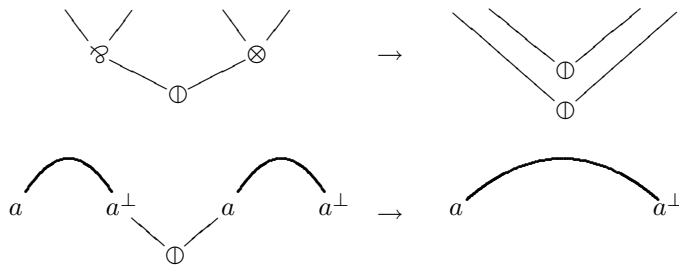


Figure 7: Cut elimination reduction steps for ordinary proof nets

2. *-AUTONOMOUS CATEGORIES

In the great majority of cases, a category with additional structure turns out to be a category that obeys a certain class of universal properties: products, coproducts, right adjoint to products, equalizers, etc. That is, the structure in question ends up being a *property* of the category, from which operations, in a more standard algebraic sense, can be extracted via the axiom of choice. There are exceptions to this, and the most important one, by far, is the concept of monoidal structure, which cannot be defined abstractly without recourse to an explicit binary operation, i.e., a bifunctor.

On this operation a form of associativity holds, which is not an equation but a natural isomorphism. A unit object for it is almost always present (hence the adjective *monoidal*), and very often there is additional structure like a braiding or a symmetry, that corresponds to a suitably generalized form of commutativity.

Monoidal categories abound in nature, and the first examples were in the world of rings and modules (and in the closely related world of topological vector spaces): the category of modules over a commutative ring—thus Abelian groups form an important special case of this—has a symmetric monoidal structure given by the operation of tensoring. The category of left-right bimodules over an arbitrary ring has a monoidal structure, not symmetrical in general. But bimodules have another interesting binary operation, the bimodule of functions, which obeys a relation of adjointness with the tensor. This additional structure on modules led to the axiomatization of monoidal closed categories. Even before these abstract concepts of monoidal and monoidal closed categories had been formulated, Lambek [Lam61] had noticed the strong logical flavor of these operations, the function module operation being a form of implication, whose associated conjunction, tensoring, was not necessarily commutative.

Monoidal closed categories first appeared in [EK66], and they give the necessary axiomatic treatment for Lambek’s implication; as a matter of fact they were formulated in such a way that in some cases an implication/function object operator can be defined without the presence of a conjunction/tensor. Thus there can be closed categories that are not monoidal.

Constructing free categories-with-structure is an interesting problem by itself; for historical reasons it is called “solving a coherence problem”. So as soon as the concept of closed category was formulated, there was the question of describing free ones. Lambek immediately saw the relationship between monoidal closed categories and logic, and produced a specific cut-elimination theorem soon after they were introduced [Lam68, Lam69]. Thus, from the start the relationship between logic and categories was bidirectional. Abstract

properties of semantical categories could give a way of formulating semantics for logical systems, as well as suggesting new such systems. And logical tools like cut-elimination could help the construction of free categories-with-structure. As a matter of fact the problem of getting a complete description of the free monoidal closed category, by means logical or not, has led to a sizable lot of publications over the last 35 years.

But some categories of modules have even more structure. First, for any commutative ring R there is always a notion of dual: the dual M^* of an R -module M is given by taking the module of functions of M into R . This defines a contravariant endofunctor; but, moreover, if R is a field and if we restrict ourselves to finitely generated modules (= finite dimensional vector spaces) over that field, then we get that M and the bidual M^{**} are related by a natural isomorphism. Thus we have something very much like an involutive negation in logic... except that in this particular case the “false” object R also takes the role of “true” in Lambek’s logical interpretation.

Driven by purely algebraic considerations M. Barr [Bar79] started looking for more examples of symmetric monoidal closed categories that have such a “dualizing” object, i.e., where every object is naturally isomorphic to its bidual. This led him to the formulation of *-autonomous categories, and to examples where the dualizer was not necessarily the unit to the tensor. He also found a general technique for producing such categories out of any ordinary monoidal closed category that has pullbacks; this is now called the Chu construction [Chu79]. The main inspiration for the Chu construction had been around since Mackey’s thesis of 1942, published in 1945 [Mac45], and can be summarized as follows. In the realm of topological vector spaces, it is very desirable to have a notion of duality which is naturally involutive, as above; the canonical example is the category of Hilbert spaces, which is unfortunately a very restricted case. Mackey’s idea was to decide that a topology on a vector space over a complete normed field could be replaced by an abstract notion of dual: another vector space whose elements are to be seen as continuous linear functionals for the other space. The relationship between these two spaces is quite symmetrical, and so the operation of taking the dual simply becomes the exchange of the two spaces.

The discovery of linear logic by Girard was completely independent from this, but came from the observation of a particular *-autonomous category, that of coherence spaces and linear maps. Coherence spaces are more closely related to the category of sets and relations than to the category of Abelian groups, but from the beginning Girard was aware that there were numerous points of contact between linear algebra and the improved logic he was seeking to create. Hence his choice of the name “linear logic”. It did not take long to establish that the categorical framework for axiomatizing (multiplicative) linear logic was *-autonomous categories; that was worked out in [Laf88, See89]. Particular cases of the Chu construction were then rediscovered by the linear logicians: applying Chu to sets and functions gives the category of Lafont-Streicher games [LS91] (now better known as “Chu spaces”), and applying it to Banach spaces gives Girard’s “coherent Banach spaces” [Gir96a], thus closing a fifty-year loop.

2.1. Basic definitions and properties. In this section we will recall the definition of a *-autonomous category and show some properties that they have and that we will use. We assume that the reader is familiar with the basic notions of category theory. Given a category \mathcal{C} and maps $f: A \rightarrow B$ and $g: B \rightarrow C$ in \mathcal{C} we will write the composition of f, g either as gf or $g \circ f$, depending on the needs of readability. We will write the identity map

on, say A as 1_A , although there is much to be said for writing it as simply A . We do not do this here because it tends to confuse beginners.

Definition 2.1.1. A *monoidal category* is a category \mathcal{C} , equipped with a bifunctor $- \otimes - : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$, a distinguished object $1_{\mathcal{C}}$, called the *unit object*⁸, and for all objects A, B, C natural isomorphisms

$$\begin{aligned} \alpha_{A,B,C} &: A \otimes (B \otimes C) \rightarrow (A \otimes B) \otimes C \quad , \\ \rho_A &: A \otimes 1_{\mathcal{C}} \rightarrow A \quad , \\ \lambda_A &: 1_{\mathcal{C}} \otimes A \rightarrow A \quad , \end{aligned}$$

such that the following diagrams commute:

$$\begin{array}{ccc} A \otimes (1_{\mathcal{C}} \otimes B) & \xrightarrow{\alpha_{A,1_{\mathcal{C}},B}} & (A \otimes 1_{\mathcal{C}}) \otimes B \\ & \searrow^{1_A \otimes \lambda_B} & \swarrow_{\rho_A \otimes 1_B} \\ & A \otimes B & \\ \\ & (A \otimes B) \otimes (C \otimes D) & \\ \alpha_{A,B,C \otimes D} \nearrow & & \searrow \alpha_{A \otimes B,C,D} \\ A \otimes (B \otimes (C \otimes D)) & & ((A \otimes B) \otimes C) \otimes D \\ & \searrow^{1_A \otimes \alpha_{B,C,D}} & \swarrow_{\alpha_{A,B,C} \otimes 1_D} \\ & A \otimes ((B \otimes C) \otimes D) & \xrightarrow{\alpha_{A,B \otimes C,D}} (A \otimes (B \otimes C)) \otimes D \end{array}$$

Let us introduce notation that will be very useful. Let I be a finite index set. A *bracketing* of I is given by a total order on $I = \{i_1, \dots, i_k\}$ and a binary tree with k leaves indexed by I such that the order is respected. We will denote bracketings of I also by I . Thus, given an I -indexed family $(C_i)_{i \in I}$ of objects of \mathcal{C} , we can use the notation $\otimes_I \{C_{i_1}, \dots, C_{i_k}\}$ to denote the object of \mathcal{C} that is obtained by applying the functor $- \otimes -$ according to the bracketing I . For empty I , let $\otimes_{\emptyset} \emptyset = 1_{\otimes}$. This allows us to state the following proposition, which we will need later.

Proposition 2.1.2. *Let I be a finite index set, and let I' and I'' be two bracketings on I that share the same order on $I = \{i_1, \dots, i_k\}$. Then for every I -indexed family $(C_i)_{i \in I}$ of objects of a monoidal category \mathcal{C} , there is a uniquely determined natural isomorphism*

$$\phi : \otimes_{I'} \{C_i \mid i \in I\} \rightarrow \otimes_{I''} \{C_i \mid i \in I\} \quad ,$$

constructed only with the available data.

Proof. This is an immediate consequence of the well-known coherence theorem for monoidal categories. (See, e.g., [Mac71, Chapter VII.2]) \square

⁸The choice of typeface should prevent confusion with identity maps.

As a matter of fact, we only have used *part* of the coherence theorem, the part that deals only with the tensor. What the above says is that we can drop parentheses when we write an expression involving only the tensor operation and an arbitrary family of objects. We just have to make sure the objects are in the same order in both expressions. But the full monoidal coherence theorem says more: not only can we drop parentheses, and write a tensor of arbitrary objects as a list/sequence, we also have the right to insert units anywhere we want in that list. There will always be a *unique* way to go from one to the other via a coherent isomorphism.

Definition 2.1.3. A *symmetric monoidal category* is a monoidal category \mathcal{C} in which for all objects A and B , there is a natural isomorphism (the *symmetry*)

$$\sigma_{A,B} : A \otimes B \rightarrow B \otimes A \quad ,$$

such that the following diagrams commute:

$$\begin{array}{ccc}
 A \otimes B & \xrightarrow{1_{A \otimes B}} & A \otimes B \\
 \searrow \sigma_{A,B} & & \nearrow \sigma_{B,A} \\
 & B \otimes A &
 \end{array}$$

$$\begin{array}{ccc}
 A \otimes 1_{\mathcal{C}} & \xrightarrow{\sigma_{A,1_{\mathcal{C}}}} & 1_{\mathcal{C}} \otimes A \\
 \searrow \rho_A & & \nearrow \lambda_A \\
 & A &
 \end{array}$$

$$\begin{array}{ccccc}
 & & (A \otimes B) \otimes C & \xrightarrow{\sigma_{A \otimes B, C}} & C \otimes (A \otimes B) & & \\
 & \nearrow \alpha_{A,B,C} & & & & \searrow \alpha_{C,A,B} & \\
 A \otimes (B \otimes C) & & & & & & (C \otimes A) \otimes B \\
 \searrow 1_A \otimes \sigma_{B,C} & & & & & \nearrow \sigma_{A,C} \otimes 1_B & \\
 & & A \otimes (C \otimes B) & \xrightarrow{\alpha_{A,C,B}} & (A \otimes C) \otimes B & &
 \end{array}$$

Proposition 2.1.2 can be generalized to symmetric monoidal categories, where we now drop the additional conditions that the two bracketings I' and I'' on I have to share the same order.

Proposition 2.1.4. *Let I be a finite index set, and let I' and I'' be two bracketings on I . Then for every I -indexed family $(C_i)_{i \in I}$ of objects of a symmetric monoidal category \mathcal{C} , there is a uniquely determined natural isomorphism*

$$\phi : \bigotimes_{I'} \{C_i \mid i \in I\} \rightarrow \bigotimes_{I''} \{C_i \mid i \in I\} \quad .$$

Proof. As in the nonsymmetric case, the proposition is an immediate consequence of the coherence theorem, which for symmetric monoidal categories has first been proved in [Mac63]. \square

Thus, not only can we drop the parentheses in a tensor-unit expression, now we can also change the order in which things are written. But we have to be a bit more careful in the symmetrical case. Given two expressions that involve the same family of objects, when one of these objects appears more than once in the family, we have to explicitly state how its instances are permuted between the two expressions. An expression like $A \otimes A \otimes A$ is used in practice, but it should be more like $A_2 \otimes A_1 \otimes A_3$ or $A_3 \otimes A_1 \otimes A_2$ or whatever.

Definition 2.1.5. A **-autonomous category* is a symmetric monoidal category \mathcal{C} equipped with a contravariant functor $(-)^{\perp} : \mathcal{C} \rightarrow \mathcal{C}$, such that for any object A , we have a natural isomorphism $A^{\perp\perp} \cong A$, and for any three objects A, B, C there is a natural bijection between

$$\mathrm{Hom}_{\mathcal{C}}(A \otimes B, C) \quad \text{and} \quad \mathrm{Hom}_{\mathcal{C}}(A, C \wp B^{\perp}) \quad ,$$

where the bifunctor $- \wp - : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ is defined by $A \wp B = (B^{\perp} \otimes A)^{\perp}$.⁹ We write $\perp_{\mathcal{C}}$ for $1_{\mathcal{C}}^{\perp}$, and call it the *dualizing object* of \mathcal{C} . We say a *-autonomous category is *strict* if the isomorphism $A^{\perp\perp} \rightarrow A$ is an identity for all objects A .¹⁰

From now on \mathcal{C} denotes a *-autonomous category. A first immediate consequence of this definition is that on every *-autonomous category \mathcal{C} we have a second symmetric monoidal structure imposed by the bifunctor $- \wp - : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ and its unit object $\perp_{\mathcal{C}}$. We will use the same notation (i.e., $\alpha, \lambda, \rho, \sigma, \dots$) for the natural isos associated to that new symmetric monoidal structure, and we will also use $\wp_I\{C_i \mid i \in I\}$ (where I is a bracketing on an index set and $(C_i)_{i \in I}$ an I -indexed family) in the same way as it has been done before for the bifunctor $- \otimes -$. For empty I , let $\wp_{\emptyset} \emptyset = \perp_{\mathcal{C}} = 1_{\mathcal{C}}^{\perp}$. Since there is no risk of confusion we tend to denote the objects $1_{\mathcal{C}}$ and $\perp_{\mathcal{C}}$ by 1 and \perp , respectively.

Let us extract some standard consequences of that natural bijection

$$\mathrm{Hom}_{\mathcal{C}}(A \otimes B^{\perp}, C) \cong \mathrm{Hom}_{\mathcal{C}}(A, C \wp B) \quad . \quad (2.1)$$

Notice that we have swapped B and B^{\perp} , which is perfectly legal given the involutory property of $(-)^{\perp}$; this way of writing things is often more convenient for us. In the above, replace A by $C \wp B$. In the right half plug the identity $1_{C \wp B}$, so at the left we get $\epsilon_{B,C} : (C \wp B) \otimes B^{\perp} \rightarrow C$, the *evaluation map* which obeys the well-known universal property:

Proposition 2.1.6. *Let A, B, C be any objects of \mathcal{C} , and let $f \in \mathrm{Hom}_{\mathcal{C}}(A \otimes B^{\perp}, C)$ and $g \in \mathrm{Hom}_{\mathcal{C}}(A, C \wp B)$ be related by the bijection (2.1). Then g is the unique map such that $f = \epsilon_{B,C} \circ (g \otimes 1_{B^{\perp}})$:*

$$A \otimes B^{\perp} \xrightarrow{g \otimes 1_{B^{\perp}}} (C \wp B) \otimes B^{\perp} \xrightarrow{\epsilon_{B,C}} C$$

The proof can found in any textbook on category theory applied to computer science, although the reader will probably see things like $A \multimap B$ or $A \Rightarrow B$ when we would write $A^{\perp} \wp B$.

It is standard to say that one map is the *transpose* of the other (an even more standard term is “exponential transpose” but in linear logic the first adjective is dropped, for obvious

⁹Observe that we stick to our notational convention of making negation switch the arguments; this is not strictly necessary but makes many formulas simpler to write.

¹⁰Note that the symmetric monoidal structure is not necessarily strict (in the usual sense that associativity and symmetry are identities).

reasons). The map g is also often called the *curryfication* of f . We will call the reverse process *de-curryfication* and we will use the term “transpose” in a generic, non-directional way.

We can apply the symmetry isomorphism to the evaluation map, and get a map $B^\perp \otimes (B \wp C) \rightarrow C$, which is also “the” evaluation map. Both versions of the evaluation map will be simply denoted by ϵ_B , since there is little chance of confusion. In the same way, we allow ourselves to write the fundamental natural bijection as $\text{Hom}_{\mathcal{C}}(A, C \wp B) \cong \text{Hom}_{\mathcal{C}}(C^\perp \otimes A, B)$. This is obtained by using symmetry, but it could be true even if we didn’t have the symmetry, for example in the case of a cyclic $*$ -autonomous category [Bar95, BLR02]. Thus we can say we have “left curryfying” and “right curryfying”.

Proposition 2.1.7. *Let $f: A \otimes B^\perp \rightarrow C$ and $g: A \rightarrow C \wp B$ be just as above, and let $h: A' \rightarrow A$, $k: B \rightarrow B'$, $l: C \rightarrow C'$, $m: B'' \rightarrow B^\perp$. Then:*

- (1) *the curryfication of $f \circ (h \otimes m)$: $A' \otimes B'' \rightarrow C$ is $(1_C \wp m^\perp) \circ g \circ h: A' \rightarrow C \wp B''^\perp$,*
- (2) *the curryfication of $f \circ (h \otimes 1_{B^\perp})$: $A' \otimes B^\perp \rightarrow C$ is $g \circ h: A' \rightarrow C \wp B$,*
- (3) *the de-curryfication of $(l \wp k) \circ g: A \rightarrow C' \wp B'$ is $l \circ f \circ (1_A \otimes k^\perp): A \otimes B'^\perp \rightarrow C'$,*
- (4) *the de-curryfication of $(l \wp 1_B) \circ g: A \rightarrow C' \wp B$ is $l \circ f: A \otimes B^\perp \rightarrow C'$.*

Proof. We can observe that (1) is just a restatement of the naturality of the defining natural isomorphism of $*$ -autonomous categories. Then (2) is obtained by replacing m by 1_{B^\perp} . Or we first could prove (2) by applying Proposition 2.1.6 to $g \circ h$ and seeing that it gives us exactly $f \circ (h \otimes 1_{B^\perp})$, and then apply that proposition again, using duality and some exchange of left and right.

The last two statements are just the duals of the first two. \square

Proposition 2.1.8. *The following diagram always commutes:*

$$\begin{array}{ccc} A^\perp \otimes (A \wp B \wp C) \otimes C^\perp & \xrightarrow{1_{A^\perp} \otimes \epsilon_C} & A^\perp \otimes (A \wp B) \\ \epsilon_A \otimes 1_{C^\perp} \downarrow & & \downarrow \epsilon_A \\ (B \wp C) \otimes C^\perp & \xrightarrow{\epsilon_C} & B \end{array}$$

Proof. The previous proposition tells us that if we right- (or C -) curryfy $\epsilon_C \circ (\epsilon_A \otimes 1_{C^\perp})$ we get

$$A^\perp \otimes (A \wp B \wp C) \xrightarrow{\epsilon_A} B \wp C \xrightarrow{\tilde{\epsilon}} B \wp C \quad ,$$

where the map $\tilde{\epsilon}$ is the curryfication of $\epsilon_C: (B \wp C) \otimes C^\perp \rightarrow B$, i.e., the identity $1_{B \wp C}$. If we then A -curryfy this composite, which is just ϵ_A we get the identity on $A \wp B \wp C$. We can do the same to $\epsilon_A \circ (1_{A^\perp} \otimes \epsilon_C)$, this time A -curryfying before we C -curryfy, and we will also get the identity on $A \wp B \wp C$. Thus the square commutes by uniqueness of transposes. \square

We could interpret this by saying that the operations of left and right curryfying commute with each other. But because we have a symmetry, it is better to say that any two successive applications of curryfication will commute with each other, the left-right distinction being purely for readability.

By curryfying the isomorphism $\rho_A: A \otimes 1 \rightarrow A$ we get an arrow $\hat{1}_A: 1 \rightarrow A^\perp \wp A$, which is often called the *name of the identity*. Its dual is $\hat{1}_A^\perp: A^\perp \otimes A \rightarrow \perp$. In general any map $f: A \rightarrow B$ has a *name* $\hat{f}: 1 \rightarrow A^\perp \wp B$, obtained by curryfying $f \circ \rho_A: A \otimes 1 \rightarrow B$.

Proposition 2.1.9. *Let \mathcal{C} be a *-autonomous category, and let C_1, \dots, C_n be objects of \mathcal{C} . Let $I, J \subseteq \{1, \dots, n\}$, and let $\mathbb{C}I = \{1, \dots, n\} \setminus I$ and $\mathbb{C}J = \{1, \dots, n\} \setminus J$ be their complements. Then for all bracketings of $I, J, \mathbb{C}I$, and $\mathbb{C}J$, we have a natural bijection between*

$$\mathrm{Hom}_{\mathcal{C}} \left(\otimes_I \{C_i^\perp \mid i \in I\}, \wp_{\mathbb{C}I} \{C_i \mid i \in \mathbb{C}I\} \right)$$

and

$$\mathrm{Hom}_{\mathcal{C}} \left(\otimes_J \{C_j^\perp \mid j \in J\}, \wp_{\mathbb{C}J} \{C_j \mid j \in \mathbb{C}J\} \right) .$$

Proof. This should be obvious in view of the previous results: one map is always obtained from the other by applying the “transpose” operator as needed. \square

In other words, any

$$f : \otimes_I \{C_i^\perp \mid i \in I\} \rightarrow \wp_{\mathbb{C}I} \{C_i \mid i \in \mathbb{C}I\}$$

uniquely determines an arrow

$$f' : \otimes_J \{C_j^\perp \mid j \in J\} \rightarrow \wp_{\mathbb{C}J} \{C_j \mid j \in \mathbb{C}J\} ,$$

and vice versa.

This means that every

$$f : \otimes_I \{C_i^\perp \mid i \in I\} \rightarrow \wp_{\mathbb{C}I} \{C_i \mid i \in \mathbb{C}I\}$$

uniquely determines a whole family of arrows, indexed by the set of bracketings on I . We will call such a family an *equivariant family*, and a member of it a *representative*. Note that if we put $I = \emptyset$ we get representatives that are more canonical than others, with source 1 ; this is also the case when we put $\mathbb{C}I = \emptyset$, where the representatives have target \perp . Proposition 2.1.9 ensures that every representative of such a family of morphisms uniquely determines the whole family. This will turn out to be very helpful for the construction of the free *-autonomous category in Section 2.3; it allows us to avoid the notion of poly- or multicategory [Lam69, Sza75, CS97b].

For example, any map, say $f : A \rightarrow B$ will have (at least) six members in its equivariant family. There will be f , with $f^\perp : B^\perp \rightarrow A^\perp$, the name $\hat{f} : 1 \rightarrow A^\perp \wp B$ and its dual $\hat{f}^\perp : B^\perp \otimes A \rightarrow \perp$, along with the twisted versions $1 \rightarrow B \wp A^\perp$ and $A \otimes B^\perp \rightarrow \perp$ of \hat{f} and \hat{f}^\perp , respectively. These two do not deserve their own special notation, and we will sometimes call them \hat{f} and \hat{f}^\perp , even if something like $\sigma_{A,B} \circ \hat{f}$ is really the correct notation.

Let us give some more standard constructions on *-autonomous categories and their relatives. Given arbitrary objects A, B, C , and D , take the tensor

$$A^\perp \otimes (A \wp B) \otimes (C \wp D) \otimes D^\perp \xrightarrow{\epsilon_A \otimes \epsilon_D} B \otimes C \quad (2.2)$$

and then curryfy twice, left and right. We get a natural¹¹ map

$$\tau_{A,B,C,D} : (A \wp B) \otimes (C \wp D) \longrightarrow A \wp (B \otimes C) \wp D ,$$

the *internal tensor*. A particular case of this is when $A = \perp$. Thus we can form

$$B \otimes (C \wp D) \xrightarrow{\lambda_B^{-1} \otimes 1_{C \wp D}} (\perp \wp B) \otimes (C \wp D) \xrightarrow{\tau_{\perp,B,C,D}} \perp \wp (B \otimes C) \wp D \xrightarrow{\lambda_{(B \otimes C) \wp D}} (B \otimes C) \wp D$$

¹¹This fact will not be used afterwards and we won't prove it.

and get an arrow, that we call *switch* [GS01, BT01] but is more traditionally known as *weak distributivity* [HdP93, CS97b] or *linear distributivity*¹², and that we denote by $\tau_{\emptyset,B,C,D}$. There is another version of switch, $\tau_{A,B,C,\emptyset}: (A \wp B) \otimes C \rightarrow A \wp (B \otimes C)$ obtained by replacing D by \perp . An interesting property of switch is that it is self-dual, i.e.,

$$\tau_{\emptyset,B,C,D}^{\perp} = \tau_{\emptyset,D^{\perp},C^{\perp},B^{\perp}} \quad \text{and} \quad \tau_{A,B,C,\emptyset}^{\perp} = \tau_{C^{\perp},B^{\perp},A^{\perp},\emptyset} \quad (2.3)$$

as the reader can show.

Proposition 2.1.10. *Let $f: A \rightarrow B$ and $g: D \rightarrow C$. Then the following holds:*

$$\begin{array}{ccc} 1 \otimes 1 & \xrightarrow{\widehat{f \otimes g}} & (A^{\perp} \wp B) \otimes (C \wp D^{\perp}) \\ \cong \downarrow & & \downarrow \tau_{A^{\perp},B,C,D^{\perp}} \\ & & A^{\perp} \wp (B \otimes C) \wp D^{\perp} \\ & & \downarrow \cong \\ 1 & \xrightarrow{\widehat{f \otimes g}} & (D^{\perp} \wp A^{\perp}) \wp (B \otimes C) \end{array} \quad (2.4)$$

Proof. Do left-right de-curryfication on $\tau_{A^{\perp},B,C,D^{\perp}} \circ (\widehat{f \otimes g})$. We get a map $(A \otimes 1) \otimes (1 \otimes D) \rightarrow B \otimes C$ and we can precompose it with $\rho_A^{-1} \otimes \lambda_D^{-1}$:

$$\begin{array}{c} A \otimes D \\ \cong \uparrow \quad \searrow \rho_A^{-1} \otimes \lambda_D^{-1} \\ (A \otimes 1) \otimes (1 \otimes D) \xrightarrow{1_A \otimes \widehat{f \otimes g} \otimes 1_D} A \otimes (A^{\perp} \wp B) \otimes (C^{\perp} \wp D) \otimes D^{\perp} \xrightarrow{\epsilon_{A^{\perp}} \otimes \epsilon_D} B \otimes C \\ \cong \downarrow \quad \nearrow 1_A \otimes \delta \otimes 1_D \\ A \otimes 1 \otimes D \end{array}$$

(δ being the obvious $\rho_1^{-1} = \lambda_1^{-1}$). It should be clear that this is $A \otimes D \xrightarrow{f \otimes g} B \otimes C$. The bottom part of the triangle adds details on how the name $\widehat{g \otimes f}$ fits in equation (2.4). \square

This gives an explanation for the name internal tensor. The following is actually more general, but we will let the reader check that.

Proposition 2.1.11. *Let $f: X \otimes Y \rightarrow A \wp B$ and $g: Z \otimes W \rightarrow C \wp D$ be maps, and let $\tilde{f}: Y \rightarrow X^{\perp} \wp A \wp B$ and $\tilde{g}: Z \rightarrow C \wp D \wp W^{\perp}$ be their curryfications. Then the left-right curryfication of*

$$X \otimes Y \otimes Z \otimes W \xrightarrow{f \otimes g} (A \wp B) \otimes (C \wp D) \xrightarrow{\tau_{A,B,C,D}} A \wp (B \otimes C) \wp D$$

is

$$Y \otimes Z \xrightarrow{\tilde{f} \otimes \tilde{g}} (X^{\perp} \wp A \wp B) \otimes (C \wp D \wp W^{\perp}) \xrightarrow{\tau_{X^{\perp} \wp A, B, C, D \wp W^{\perp}}} X^{\perp} \wp A \wp (B \otimes C) \wp D \wp W^{\perp} .$$

¹²We would like to add that this law is much more an artifact of associative logics than a form of distributivity, and that Došen's coinage *dissociativity* [DP04, DP05] for it should be considered seriously.

Proof. If we de-curryfy the first map on A, D we get (by using Proposition 2.1.6 twice)

$$\begin{array}{c} A^\perp \otimes X \otimes Y \otimes Z \otimes W \otimes D^\perp \\ \downarrow I_A \otimes f \otimes g \otimes I_D \\ A^\perp \otimes (A \wp B) \otimes (C \wp D) \otimes D^\perp \\ \downarrow \epsilon_A \otimes \epsilon_B \\ B \otimes C \end{array}$$

while if we de-curryfy the second map on $X^\perp \wp A$ and $D \wp W^\perp$ we get

$$\begin{array}{c} (A^\perp \otimes X) \otimes Y \otimes Z \otimes (W \otimes D^\perp) \\ \downarrow I_{A \otimes X^\perp} \otimes \tilde{f} \otimes \tilde{g} \otimes I_{W \otimes D^\perp} \\ (A^\perp \otimes X) \otimes (X^\perp \wp A \wp B) \otimes (C \wp D \wp W^\perp) \otimes (W \otimes D^\perp) \\ \downarrow \epsilon_{X^\perp \wp A} \otimes \epsilon_{D \wp W^\perp} \\ B \otimes C \end{array}$$

But these two are equal: just apply Proposition 2.1.8 (with the remark right after it) twice, along with the defining universal property of the transpose operation, i.e., Proposition 2.1.6. \square

Proposition 2.1.12. *The following commutes*

$$\begin{array}{ccc} (A \wp B) \otimes (C \wp D) & \xrightarrow{\tau_{A,B,C,D}} & A \wp (B \otimes C) \wp D \\ & \searrow \tau_{\emptyset, A \wp B, C, D} & \nearrow \tau_{A, B, C, \emptyset} \wp I_D \\ & & ((A \wp B) \otimes C) \wp D . \end{array}$$

Proof. We know we have proved this if we can show that the left-right de-curryfication of the composite map gives us $\epsilon_A \otimes \epsilon_D$; this is just by definition of τ . It should also be clear that the right de-curryfication of $\tau_{\emptyset, A \wp B, C, D}$ is $I_{A \wp B} \otimes \epsilon_D$, and that the left de-curryfication of $\tau_{A, B, C, \emptyset}$ is $\epsilon_A \otimes I_C$. From the first fact it follows that the right de-curryfication of the composite map is

$$(A \wp B) \otimes (C \wp D) \otimes D^\perp \xrightarrow{I_{A \wp B} \otimes \epsilon_D} (A \wp B) \otimes C \xrightarrow{\tau_{A, B, C, \emptyset}} A \wp (B \otimes C) \quad ,$$

and using the second fact it is easy to see that the left de-curryfication of this is $\epsilon_A \otimes \epsilon_D$. \square

We can also construct the composite

$$(A \wp B) \otimes (B^\perp \wp C) \xrightarrow{\tau_{A, B, B^\perp, C}} A \wp (B \otimes B^\perp) \wp C \xrightarrow{I_{A^\perp} \wp \hat{i}_B^\perp \wp I_C} A^\perp \wp \perp \wp C \xrightarrow{\cong} A^\perp \wp C$$

We call this arrow $\gamma_{A, B, C} : (A^\perp \wp B) \otimes (B^\perp \wp C) \rightarrow A^\perp \wp C$ the *internalized composition*. It should be clear (apply Proposition 2.1.7 twice) that it is obtained by applying left-right curryfication on

$$A \otimes (A^\perp \wp B) \otimes (B^\perp \wp C) \otimes C^\perp \xrightarrow{\epsilon_{A^\perp} \otimes \epsilon_C} B \otimes B^\perp \xrightarrow{\hat{i}_B^\perp} \perp$$

The name internal composition can also be explained:

Proposition 2.1.13. *Let $f: A \rightarrow B$ and $g: B \rightarrow C$. Then the following always commutes:*

$$\begin{array}{ccc} 1 \otimes 1 & \xrightarrow{\hat{f} \otimes \hat{g}} & (A^\perp \wp B) \otimes (B^\perp \wp C) \\ \cong \downarrow & & \downarrow \gamma_{A,B,C} \\ 1 & \xrightarrow{\widehat{f \circ g}} & A^\perp \wp C \end{array} \quad (2.5)$$

Proof. Repeat the proof of 2.1.10, starting by a left-right de-curryfication on $\gamma_{A,B,C} \circ (\hat{f} \otimes \hat{g})$, then doing precomposition with $\rho_A^{-1} \otimes \lambda_{C^\perp}^{-1}$. When reaching the sentence ‘‘It should be clear that...’’, what should be clear now is that we are looking at

$$A \otimes C^\perp \xrightarrow{f \otimes g^\perp} B \otimes B^\perp \xrightarrow{\hat{1}_B^\perp} \perp.$$

Seen as an equivariant family, one representative is $g \circ f$ and another is $\widehat{g \circ f}$. \square

We have never seen the following in the literature. Perhaps it can be considered trivial for a seasoned category theorist, but we think it is worthwhile proving in full.

Proposition 2.1.14 (Two-Tensor Lemma). *The following always commutes:*

$$\begin{array}{ccc} (X \wp A) \otimes (B \wp Y \wp C) \otimes (D \wp Z) & \xrightarrow{1_{X \wp A} \otimes \tau_{B \wp Y, C, D, Z}} & (X \wp A) \otimes (B \wp Y \wp (C \otimes D) \wp Z) \\ \tau_{X, A, B, Y \wp C} \otimes 1_{D \wp Z} \downarrow & & \downarrow \tau_{X, A, B, Y \wp (C \otimes D) \wp Z} \\ (X \wp (A \otimes B) \wp Y \wp C) \otimes (D \wp Z) & \xrightarrow{\tau_{X \wp (A \otimes B) \wp Y, C, D, Z}} & X \wp (A \otimes B) \wp Y \wp (C \otimes D) \wp Z \end{array} \quad (2.6)$$

Proof. Let $M = X^\perp \otimes (X \wp A) \otimes (A^\perp \wp B^\perp)$ and $N = (C^\perp \wp D^\perp) \otimes (D \wp Z) \otimes Z^\perp$. There are obvious $m: M \rightarrow B^\perp$ and $n: N \rightarrow C^\perp$, which are just sequences of evaluations. We want to show that the two ways of computing the diagonal above are equal. If we de-curryfy these maps left and right enough times, they both can be considered as maps:

$$X^\perp \otimes (A \otimes B)^\perp \otimes (X \wp A) \otimes (B \wp Y \wp C) \otimes (D \wp Z) \otimes (C \otimes D)^\perp \otimes Z^\perp \longrightarrow Y$$

whose source is isomorphic to $M \otimes (B \wp Y \wp C) \otimes N$, modulo some symmetries. Now look at

$$\begin{array}{ccccc} M \otimes (B \wp Y \wp C) \otimes N & \xrightarrow{1_M \otimes 1 \otimes n} & M \otimes (B \wp Y \wp C) \otimes C^\perp & \xrightarrow{1_M \otimes \epsilon_C} & M \otimes (B \wp Y) \\ m \otimes 1 \otimes 1_N \downarrow & & m \otimes 1 \otimes 1_{C^\perp} \downarrow & & \downarrow m \otimes 1_{B \wp Y} \\ B^\perp \otimes (B \wp Y \wp C) \otimes N & \xrightarrow{1_{B^\perp} \otimes 1 \otimes n} & B^\perp \otimes (B \wp Y \wp C) \otimes C^\perp & \xrightarrow{1_{B^\perp} \otimes \epsilon_C} & B^\perp \otimes (B \wp Y) \\ \epsilon_B \otimes 1_N \downarrow & & \epsilon_B \otimes 1_{C^\perp} \downarrow & & \downarrow \epsilon_B \\ (Y \wp C) \otimes N & \xrightarrow{1_{Y \wp C} \otimes n} & (Y \wp C) \otimes C^\perp & \xrightarrow{\epsilon_C} & Y \end{array} \quad (2.7)$$

The only small square in there that does not commute trivially is the bottom right one, and it commutes because of Proposition 2.1.8. But compare the outer square above with the previous one. Take one path of (2.6), say, first right, then down. We get a map of the form $\tau \circ (1 \otimes \tau)$. If we curryfy it twice, we get exactly the corresponding (right-right-down-down) path in (2.7). The same argument applies to the down-down-right-right path, and then since (2.7) commutes we get the result by uniqueness of transposes. \square

2.2. Proof nets form a *-autonomous category. The first basic observation of this section is that the proof nets that we have defined in Section 1 form a category. For making this precise, we provide for every formula A an identity proof net $I_A = [I_A \triangleright A^\perp, A]$, where I_A is called the *identity linking* which is defined inductively on A as follows:

$$\begin{aligned} I_a &= I_{a^\perp} = a \otimes a^\perp \\ I_\perp &= I_1 = \perp \otimes 1 \\ I_{A \wp B} &= I_{A \otimes B} = I_A \wp I_B \end{aligned}$$

Observe that we can have that $I_A = I_{A^\perp}$ because changing the order of the arguments of a \otimes or \wp in the linking of a proof graph does not change the proof net (see 1.2.5).

Furthermore, for any two proof nets $f = [P \triangleright A^\perp, B]$ and $g = [Q \triangleright B^\perp, C]$, we can define their composition $g \circ f$ to be the result of applying the cut elimination procedure to $[P \wp Q \triangleright A^\perp, B \otimes B^\perp, C]$. That this is well-defined and associative follows immediately from the strong normalization of cut elimination. We also have that $f \circ I_A = f = I_B \circ f$.

This gives rise to a category $\mathbf{PN}(\mathcal{A})$ whose objects are the formulas built from $\mathcal{A} \cup \mathcal{A}^\perp \cup \{\perp, 1\}$ via \otimes and \wp (cf. (1.1) on page 7), and whose arrows are the proof nets. More precisely, the arrows between two objects A and B are the (cut-free) proof nets $[P \triangleright A^\perp, B]$ (see Definition 1.2.6 on page 9).

The main result of this section is the following:

Proposition 2.2.1. *For every set \mathcal{A} , the category $\mathbf{PN}(\mathcal{A})$ is a (strict) *-autonomous category.*

Proof. The unit object is given by the formula 1 , and the bifunctor $- \otimes - : \mathbf{PN}(\mathcal{A}) \times \mathbf{PN}(\mathcal{A}) \rightarrow \mathbf{PN}(\mathcal{A})$ is determined by the operation \otimes on formulas, because for any two proof nets $f = [P \triangleright A^\perp, B]$ and $g = [Q \triangleright C^\perp, D]$ we have the proof net $f \otimes g = [P \wp Q \triangleright C^\perp \wp A^\perp, B \otimes D]$. We can exhibit the natural isomorphisms α , σ , ρ and λ , which are required by the definition of symmetric monoidal categories as follows:

$$\begin{aligned} \alpha_{A,B,C} &= [I_A \wp I_B \wp I_C \triangleright (C^\perp \wp B^\perp) \wp A^\perp, (A \otimes B) \otimes C] : A \otimes (B \otimes C) \rightarrow (A \otimes B) \otimes C \\ \rho_A &= [\perp \otimes I_A \triangleright \perp \wp A^\perp, A] && : A \otimes 1 \rightarrow A \\ \lambda_A &= [\perp \otimes I_A \triangleright A^\perp \wp \perp, A] && : 1 \otimes A \rightarrow A \\ \sigma_{A,B} &= [I_A \wp I_B \triangleright B^\perp \wp A^\perp, B \otimes A] && : A \otimes B \rightarrow B \otimes A \end{aligned}$$

It is easy to check that these are indeed proof nets, that they are natural isomorphisms, and that the diagrams given in Definitions 2.1.1 and 2.1.3 do indeed commute. The duality functor $(-)^\perp$ is defined on the objects as in (1.5) on page 12, and on arrows by assigning to $f = [P \triangleright A^\perp, B] : A \rightarrow B$ the arrow $f^\perp = [P \triangleright B, A^\perp] : B^\perp \rightarrow A^\perp$. Observe that in this particular case we have that $A^{\perp\perp} \rightarrow A$ is the identity, and not just an isomorphism. This will be discussed in detail in the next section. For now, it only remains to check that we have our natural bijection

$$\begin{aligned} \mathrm{Hom}(A \otimes B, C) &\cong \mathrm{Hom}(B, A^\perp \wp C) \\ [P \triangleright B^\perp \wp A^\perp, C] &\mapsto [P \triangleright B^\perp, A^\perp \wp C] \quad . \end{aligned}$$

That we have $A \wp B = (B^\perp \otimes A^\perp)^\perp$ and $\perp = 1^\perp$ does not come as a surprise. \square

2.3. The free *-autonomous category. In this section we will show that the category of proof nets is the free strict *-autonomous category: we have already observed that our $\mathbf{PN}(\mathcal{A})$ is strict, in the sense that $A \rightarrow A^{\perp\perp}$ is always the identity for every object A . So let \mathcal{A} be any set and let $\eta_{\mathcal{A}} : \mathcal{A} \rightarrow \text{Obj}(\mathbf{PN}(\mathcal{A}))$ be the function that maps every element of \mathcal{A} to itself seen as atomic formula. To say that $\mathbf{PN}(\mathcal{A})$ is the *free (strict) *-autonomous category generated by \mathcal{A}* amounts to saying that

Theorem 2.3.1. *Given a strict *-autonomous category $(\mathcal{C}, \otimes, \mathbf{1}_{\mathcal{C}}, (-)^{\perp})$ and a map $G^{\circ} : \mathcal{A} \rightarrow \text{Obj}(\mathcal{C})$, there is a unique functor $G : \mathbf{PN}(\mathcal{A}) \rightarrow \mathcal{C}$, preserving the *-autonomous structure, such that $G^{\circ} = \text{Obj}(G) \circ \eta_{\mathcal{A}}$, where $\text{Obj}(G)$ is the restriction of G on objects.*

The remainder of this section is devoted to the proof of this theorem.

Let the *-autonomous category \mathcal{C} and the embedding $G^{\circ} : \mathcal{A} \rightarrow \text{Obj}(\mathcal{C})$ be given. We will exhibit the functor $G : \mathbf{PN}(\mathcal{A}) \rightarrow \mathcal{C}$ which has the desired properties. On the objects, this functor is uniquely determined as follows:

$$\begin{aligned} G(a) &= G^{\circ}(a) & G(\perp) &= \perp_{\otimes} & G(A \wp B) &= G(A) \wp G(B) \\ G(a^{\perp}) &= G^{\circ}(a)^{\perp} & G(\mathbf{1}) &= \mathbf{1}_{\otimes} & G(A \otimes B) &= G(A) \otimes G(B) \end{aligned}$$

There is no other choice, since the objects $\mathbf{1}_{\otimes}$ and \perp_{\otimes} along with the functors $(-)^{\perp}$, $- \otimes -$, and $- \wp -$ are uniquely determined by the *-autonomous structure on \mathcal{C} .

For defining G on the morphisms, the situation is not as simple. In fact, before saying how G acts on proof nets, we will first define a mapping G^{\flat} that assigns to each ordinary proof net with cuts (see 1.3.2 and 1.4.2) an equivariant family of arrows in \mathcal{C} (as defined in Section 2.1). More precisely, let π be an ordinary proof net with conclusions $A_0, \dots, A_n, B_1 \oplus B_1^{\perp}, \dots, B_m \oplus B_m^{\perp}$ (for some $n, m \geq 0$), where A_0, \dots, A_n are the formulas in the sequent that are not cuts, and $B_1 \oplus B_1^{\perp}, \dots, B_m \oplus B_m^{\perp}$ are the cuts. For π we will construct a uniquely defined equivariant family $G^{\flat}(\pi)$ of arrows

$$\otimes_I \{G(A_i)^{\perp} \mid i \in I\} \rightarrow \wp_{\mathbb{C}I} \{G(A_i) \mid i \in \mathbb{C}I\}$$

indexed by the bracketings on the subsets $I \subseteq \{0, \dots, n\}$ and their complements. We begin with the cut-free case, i.e., the case where $m = 0$. We proceed by induction on the size of π (i.e., the sum of the numbers of \otimes - and \wp -nodes). We again make crucial use of Lemma 1.3.4, the existence of a splitting tensor.

- If the net contains no \otimes - or \wp -nodes, then it is a single ordinary axiom link with conclusion a, a^{\perp} . In this case our equivariant family is determined by the identity $1 : G(a) \rightarrow G(a)$.
- If one of the root nodes in the net is a \wp , i.e., $A_j = A'_j \wp A''_j$ for some $j \in \{0, \dots, n\}$, then we have by induction hypothesis the equivariant family with representative

$$\otimes \{G(A_i)^{\perp} \mid i \in \{0, \dots, n\} \setminus \{j\}\} \rightarrow G(A'_j) \wp G(A''_j)$$

from which we get immediately

$$\otimes \{G(A_i)^{\perp} \mid i \in \{0, \dots, n\} \setminus \{j\}\} \rightarrow G(A_j)$$

because $G(A'_j) \wp G(A''_j) = G(A_j)$.

- If one of the roots is a splitting \otimes , say $A_j = A'_j \otimes A''_j$, then by removing the \otimes -root we can get two smaller ordinary proof nets π_1 and π_2 , which are both correct. Without loss of generality, π_1 has conclusions $A_0, \dots, A_{j-1}, A'_j$ and π_2 has conclusions $A''_j, A_{j+1}, \dots, A_n$ (i.e., we might have to choose a different ordering of the A_i).

By induction hypothesis, we have two equivariant families $G^\flat(\pi_1)$ and $G^\flat(\pi_2)$, with representatives

$$\begin{aligned} \otimes \{G(A_0)^\perp, \dots, G(A_{j-1})^\perp\} &\rightarrow G(A'_j) \quad \text{and} \\ \otimes \{G(A_{j+1})^\perp, \dots, G(A_n)^\perp\} &\rightarrow G(A''_j) \quad , \end{aligned}$$

respectively, from which we get

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\} \setminus \{j\}\} \rightarrow G(A_j)$$

by applying the functor $-\otimes-$ and the fact that $G(A_j) = G(A'_j) \otimes G(A''_j)$.

In all three cases the construction is uniquely determined by the *-autonomous structure on \mathcal{C} and the choice of atoms.

Remark 2.3.2. Let π_1 and π_2 be as in the last case. We can choose representatives $r: \mathbf{1} \rightarrow G(A_0) \wp \dots \wp G(A_{j-1}) \wp G(A'_j)$ and $s: \mathbf{1} \rightarrow G(A''_j) \wp G(A_{j+1}) \wp \dots \wp G(A_n)$ for $G^\flat(\pi_1)$ and $G^\flat(\pi_2)$. Then because of Proposition 2.1.10 the following

$$\begin{array}{c} \mathbf{1} \\ \downarrow \cong \\ \mathbf{1} \otimes \mathbf{1} \\ \downarrow r \otimes s \\ (G(A_0) \wp \dots \wp G(A_{j-1}) \wp G(A'_j)) \otimes (G(A''_j) \wp G(A_{j+1}) \wp \dots \wp G(A_n)) \\ \downarrow \tau_{G(A_0 \dots A_{j-1}), G(A'_j), G(A''_j), G(A_{j+1} \dots A_n)} \\ G(A_0) \wp \dots \wp G(A_{j-1}) \wp G(A_j) \wp G(A_{j+1}) \wp \dots \wp G(A_n) \end{array}$$

is a representative of $G^\flat(\pi)$. We will also need a more general version of this: let $\{0, \dots, j-1\} = L \cup L'$ and $\{j+1, \dots, n\} = K \cup K'$ be partitions in arbitrary disjoint subsets, and choose bracketings on L , L' , K , and K' . Let

$$r': \otimes_{L'} \{G(A_l)^\perp \mid l \in L'\} \longrightarrow \wp_L \{G(A_l) \mid l \in L\} \wp G(A'_j)$$

and

$$s': \otimes_{K'} \{G(A_k)^\perp \mid k \in K'\} \longrightarrow G(A''_j) \wp \wp_K \{G(A_k) \mid k \in K\}$$

be representatives of $G^\flat(\pi_1)$ and $G^\flat(\pi_2)$ respectively. Then it should be clear, because of Proposition 2.1.11, that

$$\begin{array}{c} (\otimes_{L'} \{G(A_l)^\perp \mid l \in L'\}) \otimes (\otimes_{K'} \{G(A_k)^\perp \mid k \in K'\}) \\ \downarrow r' \otimes s' \\ (\wp_L \{G(A_l) \mid l \in L\} \wp G(A'_j)) \otimes (G(A''_j) \wp \wp_K \{G(A_k) \mid k \in K\}) \\ \downarrow \tau_{\wp_L \{G(A_l) \mid l \in L\}, G(A'_j), G(A''_j), \wp_K \{G(A_k) \mid k \in K\}} \\ \wp_L \{G(A_l) \mid l \in L\} \wp G(A_j) \wp \wp_K \{G(A_k) \mid k \in K\} \end{array}$$

is a representative of π .

It remains to show that the construction is independent from the order in which the \wp - and \otimes -nodes are introduced. We will again proceed by induction on the number of \wp - and \otimes -nodes in the net. In the base case, where there are fewer than two such nodes in the net, we have uniqueness immediately. For the inductive case, consider the last two nodes that have been introduced. If one of them is not a root, then the other is its parent, and there is no possibility in changing the order of the introductions. So assume both of them are roots. There are three cases to consider:

- Both of them are \wp -nodes, say $A_j = A'_j \wp A''_j$ and $A_k = A'_k \wp A''_k$ for some $j, k \in \{0, \dots, n\}$. Then we have by induction hypothesis the unique equivariant family with representative

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\} \setminus \{j, k\}\} \rightarrow \wp \{G(A'_j), G(A''_j), G(A'_k), G(A''_k)\}$$

from which we immediately get

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\} \setminus \{j\}\} \rightarrow G(A_j) \wp G(A_k)$$

because $G(A'_j) \wp G(A''_j) = G(A_j)$ and $G(A'_k) \wp G(A''_k) = G(A_k)$. Uniqueness follows immediately from the associativity of the functor $-\wp-$.

- One is a \wp and the other is a \otimes , say $A_j = A'_j \wp A''_j$ and $A_k = A'_k \otimes A''_k$ for some $j, k \in \{0, \dots, n\}$. Then the \otimes must be splitting, and the formula $A'_j \wp A''_j$ must belong to one of the two parts (if this is not the case, i.e., A'_j is in one part and A''_j in the other, then the \otimes must be introduced before the \wp , and we have uniqueness immediately). Without loss of generality, assume now that $A'_j \wp A''_j$ is in the part of A'_k , that $k = j + 1$, that the formulas A_0, \dots, A_{j-1} are also in the part containing A'_k , and that the formulas A_{k+1}, \dots, A_n are in the part containing A''_k . Then we have by induction hypothesis two unique equivariant families with representatives

$$\otimes \{G(A_0)^\perp, \dots, G(A_{j-1})^\perp\} \rightarrow \wp \{G(A'_j), G(A''_j), G(A'_k)\} \quad \text{and}$$

$$\otimes \{G(A_{k+1})^\perp, \dots, G(A_n)^\perp\} \rightarrow G(A''_k) \quad ,$$

from which we get (by Remark 2.3.2) immediately the unique equivariant family

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\} \setminus \{j, k\}\} \rightarrow G(A_j) \wp G(A_k) \quad .$$

- Both of them are \otimes -nodes, say $A_j = A'_j \otimes A''_j$ and $A_k = A'_k \otimes A''_k$ for some $j, k \in \{0, \dots, n\}$. Then both of them must be splitting (otherwise they cannot have been introduced consecutively.) Without loss of generality, we can decree that $j < k$, and that by removing the two \otimes -roots, we get three smaller nets, where the first contains the formulas $A_0, \dots, A_{j-1}, A'_j$, the second contains $A''_j, A_{j+1}, \dots, A_{k-1}, A'_k$, and the third contains $A''_k, A_{k+1}, \dots, A_n$. By induction hypothesis, we have three uniquely determined equivariant families, with representatives

$$1 \rightarrow \wp \{G(A_0), \dots, G(A_{j-1}), G(A'_j)\} \quad ,$$

$$1 \rightarrow \wp \{G(A''_j), G(A_{j+1}), \dots, G(A_{k-1}), G(A'_k)\} \quad , \quad \text{and}$$

$$1 \rightarrow \wp \{G(A''_k), G(A_{k+1}), \dots, G(A_n)\} \quad .$$

There are now two ways of constructing the representative

$$1 \rightarrow \wp \{G(A_0), \dots, G(A_n)\} \quad .$$

It follows immediately from the two-tensor lemma (Proposition 2.1.14), that both yield the same equivariant family.

We now extend the construction of the equivariant families to ordinary proof nets with cuts, i.e., the case where $m > 0$. This is done by first replacing each cut $B_i \oplus B_i^\perp$ in π by the \otimes -formula $B_i \otimes B_i^\perp$, and then applying the previous construction to the net with conclusions $A_0, A_1, \dots, A_n, B_1 \otimes B_1^\perp, \dots, B_m \otimes B_m^\perp$, which yields in particular the representative

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\}\} \rightarrow \wp \{G(B_j) \otimes G(B_j)^\perp \mid j \in \{1, \dots, m\}\} \quad .$$

Look at the map $\hat{I}_{G(B_j)}^\perp: G(B_j) \otimes G(B_j)^\perp \rightarrow \perp$ (“the co-name of the identity”) which exist for every B_j . By taking the par of the family $(\hat{I}_{G(B_j)}^\perp)$, we construct

$$\zeta: \wp \{G(B_j) \otimes G(B_j)^\perp \mid j \in \{1, \dots, m\}\} \rightarrow \wp \{\perp \mid j \in \{1, \dots, m\}\} \cong \perp$$

and by composition we get

$$\otimes \{G(A_i)^\perp \mid i \in \{0, \dots, n\}\} \rightarrow \perp \quad ,$$

which we define as a representative of the equivariant family $G^b(\pi)$. The uniqueness of this follows from the functoriality and associativity of $- \otimes -$.

Remark 2.3.3. Suppose that we have nets π_1 on Γ, A and π_2 on A^\perp, Δ . Let $f_1: \Gamma^\perp \rightarrow A$ and $f_2: A \rightarrow \Delta$ be representatives of π_1 and π_2 , respectively. It should be clear that if we construct π by cutting π_1, π_2 on A, A^\perp , then $f_2 \circ f_1$ is a representative of $G(\pi)$. This is obtained by looking at the definition of the internal composition γ as well as Proposition 2.1.13 and Remark 2.3.2.

A crucial observation about this construction is that eliminating cuts from an ordinary proof net (see 1.5.13) does not affect the equivariant family it defines:

Lemma 2.3.4. *Let π be an ordinary proof net with conclusions*

$$A_0, \dots, A_n, B_1 \oplus B_1^\perp, \dots, B_m \oplus B_m^\perp$$

(for some $n, m \geq 0$), where A_0, \dots, A_n are the formulas in the sequent that are not cuts, and $B_1 \oplus B_1^\perp, \dots, B_m \oplus B_m^\perp$ are the cuts. Let π' be the ordinary proof net with conclusions

$$A_0, \dots, A_n \quad ,$$

that is obtained from π by applying the cut elimination procedure. Then π and π' determine the same equivariant family of arrows

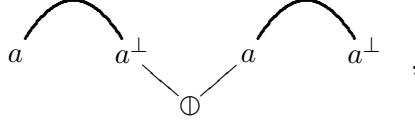
$$\otimes_I \{G(A_i)^\perp \mid i \in I\} \rightarrow \wp_{\mathbb{C}I} \{G(A_i) \mid i \in \mathbb{C}I\}$$

indexed by the bracketings on the subsets $I \subseteq \{0, \dots, n\}$ and their complements.

Remark 2.3.5. This lemma would suffice to prove that “ordinary proof nets with two conclusions form the free *-autonomous category without units”, and is also an immediate consequence of this fact [Blu93]. But it is only recently that precise and fully satisfactory definitions for a notion of *-autonomous category without units have been proposed [LS05a, HHS05, DP05], and we will not pursue this matter any further here.

Proof of Lemma 2.3.4. The proof will be done by induction on the length of the cut reduction. It suffices to show the lemma for the case where π' is obtained from π by a single cut reduction step. We use the following convention: Given a sequent of n formulas $\Gamma = A_1, \dots, A_n$ we write $G(\Gamma)$ for $G(A_1) \wp \dots \wp G(A_n)$. We will now proceed by induction on the size of π . There are four cases to consider:

- (1) If π contains a \wp -root, then this \wp -root is also present in π' . Therefore we can remove it in both nets and apply the induction hypothesis.
- (2) If π is of the following shape:



i.e., it consists of a single \oplus -node and two axiom links. Then π' is a single axiom link:



Identity maps are representatives of axiom links (and in particular of $G^b(\pi')$), and Remark 2.3.3 tells us that $G^b(\pi)$ has representative $1_{G(a)} \circ 1_{G(a)}$.

- (3) The net π' is obtained from π by reducing a cut formula $(A \otimes B) \oplus (B^\perp \wp A^\perp)$, where both, the \oplus -node, as well as its \otimes -child are splitting, i.e., by removing them π falls into three components:

- First, we have the net π_1 with conclusions Γ, A . Let $f_1 : G(\Gamma)^\perp \rightarrow G(A)$ be an arrow that represents $G^b(\pi_1)$.
- Second, we have the net π_2 with conclusions Δ, B . Let $f_2 : G(\Delta)^\perp \rightarrow G(B)$ represent $G^b(\pi_2)$.
- Finally, we have the net π_3 with conclusions $\Theta, B^\perp \wp A^\perp$. Let $f_3 : G(\Theta)^\perp \rightarrow G(B)^\perp \wp G(A)^\perp$ represent $G^b(\pi_3)$. The same arrow also represents $G^b(\pi_4)$, where π_4 is the net with conclusions Θ, B^\perp, A^\perp that is obtained from π_3 by removing the \wp .

Obviously the composite

$$\begin{array}{c}
 (G(\Gamma)^\perp \otimes G(\Delta)^\perp) \otimes G(\Theta)^\perp \\
 \downarrow (f_1 \otimes f_2) \otimes f_3 \\
 (G(A) \otimes G(B)) \otimes (G(B)^\perp \wp G(A)^\perp) \\
 \downarrow \hat{1}_{G(A \otimes B)}^\perp \\
 \perp
 \end{array}$$

represents $G^b(\pi)$. But we can also take our three nets and do two tensor introductions on them, to get a net with conclusions $\Delta, B \otimes B^\perp, A^\perp \otimes A, \Gamma, \Theta$. Let

$$G(\Delta)^\perp \otimes G(\Theta)^\perp \otimes G(\Gamma)^\perp \xrightarrow{h} (G(B) \otimes G(B)^\perp) \wp (G(A)^\perp \otimes G(A))$$

represent that net. It should be obvious that the composite

$$\begin{array}{c}
G(\Delta) \otimes G(\Theta) \otimes G(\Gamma) \\
\downarrow h \\
(G(B) \otimes G(B)^\perp) \wp (G(A)^\perp \otimes G(A)) \\
\downarrow \hat{i}_{G(B)}^\perp \wp \hat{i}_{G(A)}^\perp \\
\perp \wp \perp \\
\downarrow \cong \\
\perp
\end{array}$$

represents the net $G^b(\pi')$. Now look at the following diagram

$$\begin{array}{ccccc}
G(\Gamma)^\perp \otimes G(\Delta)^\perp \otimes G(\Theta)^\perp & \xrightarrow{f_1 \otimes f_2 \otimes f_3} & G(A) \otimes G(B) \otimes (G(B)^\perp \wp G(A)^\perp) & \xrightarrow{\hat{i}_{G(A \otimes B)}^\perp} & \perp \\
\downarrow \cong & & \downarrow \cong & & \downarrow \cong \\
& & G(B) \otimes (G(B)^\perp \wp G(A)^\perp) \otimes G(A) & & \\
& & \downarrow w & & \\
G(\Delta)^\perp \otimes G(\Theta)^\perp \otimes G(\Gamma)^\perp & \xrightarrow{h} & (G(B) \otimes G(B)^\perp) \wp (G(A)^\perp \otimes G(A)) & \xrightarrow{\hat{i}_{G(B)}^\perp \wp \hat{i}_{G(A)}^\perp} & \perp \wp \perp
\end{array}$$

where w is

$$\begin{array}{c}
G(B) \otimes (G(B)^\perp \wp G(A)^\perp) \otimes G(A) \\
\downarrow I_{G(B)} \otimes \tau_{G(B)^\perp, G(A)^\perp, G(A), \emptyset} \\
G(B) \otimes (G(B)^\perp \wp (G(A)^\perp \otimes G(A))) \\
\downarrow \tau_{\emptyset, G(B), G(B)^\perp, G(A)^\perp \otimes G(A)} \\
(G(B) \otimes G(B)^\perp) \wp (G(A)^\perp \otimes G(A))
\end{array}$$

The left rectangle of the big diagram commutes because we can apply the general version of Remark 2.3.2 twice, once for each occurrence of τ in w (i.e., once for each tensor introduction). Then if we take the dual of w , we get (using Equations (2.3))

$$\begin{array}{c}
(G(A)^\perp \wp G(A)) \otimes (G(B) \wp G(B)^\perp) \\
\downarrow \tau_{\emptyset, G(A)^\perp \wp G(A), G(B), G(B)^\perp} \\
((G(A)^\perp \wp G(A)) \otimes G(B)) \wp G(B)^\perp \\
\downarrow \tau_{G(A)^\perp, G(A), G(B), \emptyset} \wp I_{G(B)^\perp} \\
G(A)^\perp \wp (G(A) \otimes G(B)) \wp G(B)^\perp
\end{array}$$

and this shows that $w^\perp = \tau_{G(A)^\perp, G(A), G(B), G(B)^\perp}$ because of Proposition 2.1.12. We can now see that the right half of the big diagram is exactly the dual of Equation (2.4), thus showing that the whole diagram commutes, from which we get that $G^\flat(\pi) = G^\flat(\pi')$.

- (4) If none of the three cases above holds, then π must contain a \otimes -root or a \oplus which is splitting. The same node is also splitting in π' . By removing it, the net π falls into two parts, say π_1 and π_2 . Similarly, π' falls into π'_1 and π'_2 . Without loss of generality, we can assume that π_1 contains the induction hypothesis' cut. Then, we have that π'_1 is the result of reducing it, and also that $\pi'_2 = \pi_2$. We can therefore apply the induction hypothesis. \square

We can now proceed in the proof of Theorem 2.3.1 by showing how the functor G is defined on the arrows. Recall that each proof graph $P \triangleright \Gamma$ can be seen as an ordinary proof net with conclusion P^*, Γ (see Observation 1.3.3), to which we can apply the construction of the equivariant families. This construction gives us in particular for each proof graph $P \triangleright A^\perp, B$ a unique arrow $\psi_{P \triangleright A^\perp, B} : G(P^*)^\perp \rightarrow G(A^\perp) \wp G(B)$.

Furthermore, every linking P uniquely determines an arrow $\phi_P : 1 \textcircled{\otimes} \rightarrow G(P^*)^\perp$ in \mathcal{C} , which is inductively obtained as follows:

$$\begin{aligned} \phi_1 &= 1_1 && : 1 \rightarrow 1 \\ \phi_{a \otimes a^\perp} &= \hat{1}_{G(a)^\perp} && : 1 \rightarrow G(a) \wp G(a)^\perp \\ \phi_{a^\perp \otimes a} &= \hat{1}_{G(a)} && : 1 \rightarrow G(a)^\perp \wp G(a) \\ \phi_{\perp \otimes P'} &= \rho_{G(P'^*)^\perp}^\perp \circ \phi_{P'} && : 1 \rightarrow \perp \wp G(P'^*)^\perp \\ \phi_{P' \otimes \perp} &= \lambda_{G(P'^*)^\perp}^\perp \circ \phi_{P'} && : 1 \rightarrow G(P'^*)^\perp \wp \perp \\ \phi_{P' \wp P''} &= (\phi_{P'} \otimes \phi_{P''}) \circ \lambda_1^{-1} && : 1 \rightarrow G(P'^*)^\perp \otimes G(P''^*)^\perp \end{aligned}$$

The arrow ϕ_P can be composed with $\psi_{P \triangleright A^\perp, B}$ to get $\xi_{[P \triangleright A^\perp, B]} : 1 \rightarrow G(A^\perp) \wp G(B)$. That this is well-defined, is ensured by the following lemma (in which we finally use the fact that the units are units).

Lemma 2.3.6. *If $P \triangleright A^\perp, B \sim Q \triangleright A^\perp, B$, then $\xi_{[P \triangleright A^\perp, B]} = \xi_{[Q \triangleright A^\perp, B]}$.*

Proof. First notice that all the one-step equivalences in 1.2.5 involve two formulas that have exactly the same set of atoms (here, naturally, 1 and \perp are considered to be atoms). Without loss of generality, let us assume that P is the linking tree of the left-hand side of one of these equivalences and Q is the linking tree of the right-hand side. Then there is an ordinary proof net for the sequent $P^*, Q^{\star\perp}$, in which the graph of axiom links forms a bijection between the atoms of the two formulas.¹³ Let us call this proof net π_1 . Thus $G^\flat(\pi_1)$ defines a map $\theta : G(Q^*) \rightarrow G(P^*)$. In addition, θ is always an isomorphism in \mathcal{C} . This is because everything that has to do with the units “just melts away” and the two formulas $G(Q^*)^\perp \cong G(P^*)^\perp$ are thus both isomorphic to $\otimes_{a \otimes a^\perp} \{G(a) \wp G(a)^\perp\}$, where $a \otimes a^\perp$ ranges over the “real” axiom links, which are the same in Q and P . Therefore we

¹³In all cases except the last one, it does not matter whether P^* or Q^* is negated, but in the last one (the one with the side condition) we only get a correct ordinary proof net if we negate Q^* .

immediately have that the left-hand side triangle in the following diagram commutes:

$$\begin{array}{ccc}
 & \xi_{[P \triangleright A^\perp, B]} & \\
 & \curvearrowright & \\
 1 & \begin{array}{c} \xrightarrow{\phi_P} G(P^*)^\perp \\ \xrightarrow{\phi_Q} G(Q^*)^\perp \end{array} & \begin{array}{c} \xrightarrow{\psi_{P \triangleright A^\perp, B}} \\ \xrightarrow{\psi_{Q \triangleright A^\perp, B}} \end{array} G(A^\perp) \wp G(B) \\
 & \begin{array}{c} \downarrow \theta^\perp \\ \downarrow \end{array} & \\
 & \curvearrowleft & \\
 & \xi_{[Q \triangleright A^\perp, B]} &
 \end{array}$$

For showing that the right-hand side triangle commutes we will apply Lemma 2.3.4. For this, consider the following four ordinary proof nets:

- (1) π_1 is as above.
- (2) $\pi_2 = \pi_{P \triangleright A^\perp, B}$ is the ordinary proof net with conclusions P^*, A^\perp, B that corresponds to the proof graph $P \triangleright A^\perp, B$ (see Observation 1.3.3).
- (3) Similarly, $\pi_3 = \pi_{Q \triangleright A^\perp, B}$ is the ordinary proof net with conclusions Q^*, A^\perp, B that corresponds to the proof graph $Q \triangleright A^\perp, B$.
- (4) Finally, the ordinary proof net π_4 is obtained from π_1 and π_3 by connecting Q^* and Q^* with a \oplus -node.

By definition, the arrows $\psi_{P \triangleright A^\perp, B}$ and $\psi_{Q \triangleright A^\perp, B}$ are representatives of the equivariant families obtained from π_2 and π_3 , respectively. Similarly, the isomorphism θ^\perp is a representative of the equivariant family obtained from π_1 . Consequently, the composition $\psi_{Q \triangleright A^\perp, B} \circ \theta^\perp$ is a representative of the equivariant family obtained from π_4 (because of Remark 2.3.3). Furthermore, it is easy to see that eliminating the cut from π_4 (as defined in 1.5.13) yields π_2 . Therefore we can apply Lemma 2.3.4 to get that $\psi_{Q \triangleright A^\perp, B} \circ \theta^\perp = \psi_{P \triangleright A^\perp, B}$. \square

We have shown that for any proof net of the form $f = [P \triangleright A^\perp, B]$, the arrow $G(f) : G(A) \rightarrow G(B)$ determined by $\xi_{[P \triangleright A^\perp, B]}$ via Proposition 2.1.9 is uniquely defined. It remains to prove that $G : \mathbf{PN}(\mathcal{A}) \rightarrow \mathcal{C}$ is indeed a functor (i.e., identities and composition are preserved). That for each formula A , the proof $[I_A \triangleright A^\perp, A]$ is mapped to identity $1 : G(A) \rightarrow G(A)$ is an easy induction on the structure of A and left to the reader. The crucial part is to show that for two given proof nets $f = [P \triangleright A^\perp, B]$ and $g = [Q \triangleright B^\perp, C]$, the composition $G(g) \circ G(f)$ yields the same arrow in \mathcal{C} , as $G(g \circ f)$.

Observe that $g \circ f = [R \triangleright A^\perp, C]$ is the proof net that is obtained by eliminating the cut in $[P \wp Q \triangleright A^\perp, B \oplus B^\perp, C]$, whose equivariant family is determined by

$$\begin{array}{c}
1 \\
\downarrow \cong \\
1 \otimes 1 \\
\downarrow \phi_P \otimes \phi_Q \\
G(P^*)^\perp \otimes G(Q^*)^\perp \\
\downarrow \psi_{P \triangleright A^\perp \wp B} \otimes \psi_{Q \triangleright B^\perp \wp C} \\
(G(A)^\perp \wp G(B)) \otimes (G(B)^\perp \wp G(C)) \\
\downarrow \tau_{G(A)^\perp, G(B), G(B)^\perp, G(C)} \\
G(A)^\perp \wp (G(B) \otimes G(B)^\perp) \wp G(C) \\
\downarrow 1 \otimes \hat{1} \otimes 1 \\
G(A)^\perp \wp G(C)
\end{array}$$

and a by-now standard argument tells us that this is $G(\widehat{g \circ f})$. In order to show that this is also $\widehat{G(g \circ f)}$, it suffices to show the following general result.

Lemma 2.3.7. *Let $T \triangleright \Gamma \rightarrow S \triangleright \Delta$, i.e., the proof graph $S \triangleright \Delta$ is obtained from $T \triangleright \Gamma$ by applying a single cut reduction step. Then $\xi_{[T \triangleright \Gamma]}$ and $\xi_{[S \triangleright \Delta]}$ denote the same morphism $1_{\mathcal{C}} \rightarrow \wp \{G(A_1), \dots, G(A_n)\}$, where A_1, \dots, A_n are the formulas in Γ (resp. Δ) that are not cuts.*

Proof. The proof is very much like that of Lemma 2.3.6: for every case of a cut reduction step we will construct an ordinary proof net π_1 whose conclusions will involve T^* , $S^{*\perp}$, and which will also define a map $\theta: G(S^*) \rightarrow G(T^*)$. Then we will show that the two triangles below commute:

$$\begin{array}{ccccc}
& & G(T^*)^\perp & & \\
& \phi_T \nearrow & \downarrow \theta^\perp & \searrow \psi_{T \triangleright \Gamma} & \\
1 & & & & \wp \{G(A_1), \dots, G(A_n)\} \\
& \phi_S \searrow & & \nearrow \psi_{S \triangleright \Delta} & \\
& & G(S^*)^\perp & &
\end{array}$$

Again we need three ordinary proof nets in addition of π_1 :

- (2) Let $\pi_2 = \pi_{T \triangleright \Gamma}$ with conclusions $T^*, A_1, \dots, A_n, B_1 \oplus B_1^\perp, \dots, B_m \oplus B_m^\perp$, where $B_1 \oplus B_1^\perp, \dots, B_m \oplus B_m^\perp$ are the cuts in Γ . Applying the construction of the equivariant family yields the arrow $\psi_{T \triangleright \Gamma}$.
- (3) Let $\pi_3 = \pi_{S \triangleright \Delta}$ with conclusions $S^*, A_1, \dots, A_n, C_1 \oplus C_1^\perp, \dots, C_l \oplus C_l^\perp$, where $C_1 \oplus C_1^\perp, \dots, C_l \oplus C_l^\perp$ are the cuts in Δ . The arrow $\psi_{S \triangleright \Delta}$ is obtained by applying the construction of the equivariant family to π_3 .
- (4) Finally, π_4 is obtained by composing π_1 and π_3 with a cut on $S^{*\perp}$ and S^* .

There are three cases to consider:

- The reduced cut (see Section 1.5) is on binary connectives. In this case T^* and S^* are identical, and π_1 is the usual identity net, so that θ is the identity map. The commutativity of the left-hand side triangle follows trivially. For the right-hand side triangle, observe that the result of eliminating the S^* -cut on the composite π_4 yields exactly π_2 . We can therefore apply Lemma 2.3.4.
- The reduced cut is on atoms. Then (see Section 1.5) we can assume that $T^* = P\{(a_k^\perp \otimes a_j) \wp (a_i^\perp \otimes a_h)\}$ and $S^* = P\{a_k^\perp \otimes a_h\}$ for some context $P\{ \}$. Furthermore, one of the cuts in Γ is $a_i \oplus a_j^\perp$. It should be clear that there is a correct ordinary proof net π_0 with conclusions

$$(a_k^\perp \otimes a_j) \wp (a_i^\perp \otimes a_h), a_h^\perp \wp a_k, a_i \oplus a_j^\perp \quad ,$$

where the same index on an atom and a negated atom denotes the presence of an axiom link between the two (here there is a little breach in our previous convention of using a different index for every single atom). Out of this we can construct π_1 with conclusions

$$P\{(a_k^\perp \otimes a_j) \wp (a_i^\perp \otimes a_h)\}, P^\perp\{a_h^\perp \wp a_k\}, a_i \oplus a_j^\perp \quad ,$$

where the additional axiom links simply connect every atom of the context $P\{ \}$ to its corresponding negation in $P^\perp\{ \}$. It should be clear that π_1 is correct. Furthermore, the result of eliminating the two cuts $S^{\star\perp} \oplus S^*$ and $a_i \oplus a_j^\perp$ in the composite π_4 yields exactly π_2 . Therefore, by Lemma 2.3.4, the right-hand side triangle commutes. For the commutativity of the left-hand side triangle consider again the net π_0 . A representative for the equivariant family $G^b(\pi_0)$ is given by the map

$$\begin{array}{c} (G(a_h)^\perp \wp G(a_i)) \otimes (G(a_j)^\perp \wp G(a_k)) \\ \downarrow \tau \\ G(a_h)^\perp \wp (G(a_i) \otimes G(a_j)^\perp) \wp G(a_k) \\ \downarrow I \wp \hat{I}^\perp \wp I \\ G(a_h)^\perp \wp G(a_k) \end{array}$$

(here the indices are used only as position markers; there are only two distinct “atomic” objects of \mathcal{C} , namely $G(a)$ and $G(a)^\perp$). By definition, this is the internal composition

$$(G(a)^\perp \wp G(a)) \otimes (G(a)^\perp \wp G(a)) \xrightarrow{\gamma_{G(a), G(a), G(a)}} G(a)^\perp \wp G(a) \quad .$$

If we compose this with the names of the identity:

$$1 \xrightarrow{\cong} 1 \otimes 1 \xrightarrow{\hat{I}_{G(a)} \otimes \hat{I}_{G(a)}} (G(a)^\perp \wp G(a)) \otimes (G(a)^\perp \wp G(a)) \xrightarrow{\gamma_{G(a), G(a), G(a)}} G(a)^\perp \wp G(a)$$

we get (by Proposition 2.1.13) the name of the identity $\hat{I}_{G(a)}: 1 \rightarrow G(a)^\perp \wp G(a)$, which in turn represents the result of eliminating the cut from π_0 . Therefore the left-hand side triangle commutes for the case where $P\{ \} = \{ \}$ is the empty context. The general case follows by a straightforward induction on $P\{ \}$.

- The reduced cut is on units. Then we have that $T^* = P\{\perp_j \wp(1_i \otimes Q)\}$ and $S^* = P\{Q\}$ for some context $P\{ \}$ and some formula Q (and one of the cuts in Γ is $\perp_i \oplus 1_j$). We now construct π_1 so that its conclusions are

$$P\{\perp_j \wp(1_i \otimes Q)\}, P^\perp\{Q^\perp\}, \perp_i \oplus 1_j \quad ,$$

in other words, in such a way that the only non-trivial axiom links are indicated by the indices i and j . It should be obvious that this gives a correct (ordinary) net. We now see that the map θ is an isomorphism and that the two triangles commute, since all the syntactical entities that do not belong to $P\{ \}$ and Q simply “melt away” in the categorical interpretation because they follow the coherence laws for units. \square

We now have completed the proof of Theorem 2.3.1. To see this, let us recall the main features of our construction:

- (1) Functoriality of G together with Lemma 2.3.6 and Proposition 2.1.9 ensure that G does preserve the $*$ -autonomous structure.
- (2) By sequentialisation, every morphism in the category $\mathbf{PN}(\mathcal{A})$ is expressible in the terms of the $*$ -autonomous structure imposed on $\mathbf{PN}(\mathcal{A})$ in Section 2.2. Consequently, the functor G is indeed uniquely defined by the data given in Theorem 2.3.1.

It might be worth mentioning, that the main result of [BC99], namely that two MLL formulas are isomorphic if and only if they can be transformed into each other by applying the standard rewriting rules of associativity, commutativity, and unit (for \wp and \otimes), is an immediate consequence of Theorem 2.3.1. Furthermore, Theorem 2.3.1 provides a decision procedure for the equality of morphisms in the free symmetric $*$ -autonomous category, which is in our opinion simpler than the ones provided in [BCST96] and [KO99].

3. CONCLUSION

We think we made a convincing case for the the cleanest approach yet to proof nets with the multiplicative units. In particular, our main results are stated in such a way as to be easily applicable; in addition our techniques can certainly be used in more general situations than purely multiplicative linear logic.

We began with a discussion on the relationship between proof systems and categories; it turns out that the writing up of this paper gave many occasions to deepen that reflection, and more will said about that relationship in subsequent work. We made use of two unstated assumptions, which certainly belong to “mainstream ideology”:

- that there is a single way to introduce bottom (for instance we also could have a special axiom for it),
- that the standard equations for units in monoidal categories should be used for proofs.

We now think that these standard postulates deserve more scrutiny [LS05b, LS05a], but we make no predictions about the conclusions we will eventually reach.

These new subtleties in no way modify our general belief: that category theory should be used as a general algebraic yardstick for tackling the questions related to identifications of proofs. As we have said at the beginning, this can work well only if we allow a certain ideological flexibility on *both* the proof-theoretical and category-theoretical side.

There are some issues that are left open and that we want to explore in the future:

- The addition of Mix. Ordinary proof nets have a weaker version of the Danos-Regnier correctness criterion (every switching produces an acyclic, but not necessarily connected, graph) which gives a sequentialization theorem for MLL with the (binary) Mix rule added [FR94]. An important property of this setting is that when a net is correct, the number of connected components is invariant with respect to the actual switching. It is not hard to see that if we add binary Mix to our theory of proof nets, this invariant is respected by both bottom introductions and the equations of 1.2.5. This allows us to say that our theory extends to MLL with Mix, although there is some nontrivial work left to be done, namely to prove that we actually have constructed the free *-autonomous category with Mix. Thus we have to manage the additional algebra needed for this (cf. [CS97a, FP04a, LS05a]), which involves the necessary equations that are required to obtain a coherence result when the “mother of all mix maps” $\perp \rightarrow 1$ is added to a *-autonomous category. Another, also standard view of Mix is adding the requirement that this map be an isomorphism. But then the theory of proof nets presented in [FR94] (when we add one constant with two introduction rules to the logic) is sufficient to deal with this case.
- Exploring the noncommutative world, especially the particular logic where the context structure is no longer a multiset of formulas but a cyclic order of formulas [Yet90, LR96]. In the unit-free case, the correctness criterion has to be modified such that the net has to be planar (i.e., no crossings of edges are allowed). It is easy to see that our correctness criterion and the equivalence relation defined in 1.2.5 can be adapted accordingly. However, the question is whether we can obtain a well-behaved cut elimination such that we can construct the free cyclic *-autonomous category [Ros94, Bar95, Sch99]. Here is another interesting question: Could it be that in the noncommutative case we can find normal representatives for proofs instead of having to rely on equivalence classes?
- The relation with the calculus of structures [GS01, BT01] and its use of deep inference. We should mention that the idea behind our approach originates from the new viewpoints that are given by deep inference.
- The addition of additives to our theory. This should not be very hard, given the work done in [HvG03]. The true challenge is to include also the additive units.
- The development of a theory of proof nets for classical logic. The problem is finding the right extension of the axioms of a *-autonomous category, such that on one hand classical proofs are identified in a natural way, but that on the other hand there is no collapse to a boolean algebra. While we were writing this, we became aware of [FP04b, FP04c, FP04a], which tackle this very problem. Some additional research [LS05b, LS05a, Lam06, Str05] allows us to say that the last word on the relationship between classical logic and categories will not be said in the near future.
- The search for meaningful invariants. It is very probable that the equivalence classes of graphs we define have a geometric meaning, and can be related to more abstract invariants like those given by homological algebra. We are convinced that the work in in [Mét94] is only the tip of the iceberg.

REFERENCES

- [Bar79] Michael Barr. **-Autonomous Categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, 1979.
- [Bar95] Michael Barr. Non-symmetric *-autonomous categories. *Theoretical Computer Science*, 139:115–130, 1995.
- [BC99] Vincent Balat and Roberto Di Cosmo. A linear logical view of linear type isomorphisms. In *Computer Science Logic, CSL 1999*, volume 1683 of *LNCS*, pages 250–265. Springer-Verlag, 1999.
- [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.
- [BLR02] Richard Blute, François Lamarche, and Paul Ruet. Entropic hopf algebras and models of non-commutative linear logic. *Theory and Applications of Categories*, 10:0–36, 2002.
- [Blu93] Richard Blute. Linear logic, coherence and dinaturality. *Theoretical Computer Science*, 115:3–41, 1993.
- [BT01] Kai Brännler 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.
- [Chu79] P.-H. Chu. Constructing *-autonomous categories, 1979. Appendix to [Bar79].
- [CS97a] J.R.B. Cockett and R.A.G. Seely. Proof theory for full intuitionistic linear logic, bilinear logic, and mix categories. *Theory and Applications of Categories*, 3(5):85–131, 1997.
- [CS97b] J.R.B. Cockett and R.A.G. Seely. Weakly distributive categories. *Journal of Pure and Applied Algebra*, 114:133–173, 1997.
- [Dan90] Vincent Danos. *La logique linéaire appliquée l'étude de divers processus de normalisation (principalement du λ -calcul)*. Thèse de Doctorat, Université Paris VII, 1990.
- [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.
- [DW03] Gilles Dowek and Benjamin Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.
- [EK66] Samuel Eilenberg and Gregory Maxwell Kelly. Closed categories. In S. Eilenberg, D.K. Harrison, S. MacLane, and H. Roehrl, editors, *Proceedings of the La Jolla Conference in Categorical Algebra*, pages 421–562. Springer, 1966.
- [FP04a] Carsten Führmann and David Pym. On the geometry of interaction for classical logic. preprint, 2004.
- [FP04b] Carsten Führmann and David Pym. On the geometry of interaction for classical logic (extended abstract). In *19th IEEE Symposium on Logic in Computer Science (LICS 2004)*, pages 211–220, 2004.
- [FP04c] 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.
- [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.
- [Gir96a] Jean-Yves Girard. Coherent Banach spaces: a continuous denotational semantics. In Jean-Yves Girard, Mitsuhiro Okada, and Andre Scedrov, editors, *Electronic Notes in Theoretical Computer Science*, volume 3. Elsevier Science Publishers, 1996.
- [Gir96b] 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.
- [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.
- [Hay85] Susumu Hayashi. Adjunctions of semifunctors: Categorical structures in nonextensional lambda calculus. *Theoretical Computer Science*, 41:95–104, 1985.

- [HdP93] J. Martin E. Hyland and Valeria de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64(3):273–291, 1993.
- [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.
- [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.
- [KM71] Gregory Maxwell Kelly and Saunders Mac Lane. Coherence in closed categories. *Journal of Pure and Applied Algebra*, 1:97–140, 1971.
- [KO99] Thong-Wei Koh and Chih-Hao Luke Ong. Internal languages for autonomous and *-autonomous categories. In Martin Hofmann, Giuseppe Rosolini, and Duško Pavlović, editors, *Proceedings of the 8th Conference on Category Theory and Computer Science, 1999*, volume 29 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.
- [Laf88] Yves Lafont. *Logique, Catégories et Machines*. PhD thesis, Université Paris 7, 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.
- [Lam61] Joachim Lambek. On the calculus of syntactic types. In R. Jakobson, editor, *Studies of Language and its Mathematical Aspects*, Proc. of the 12th Symp. Appl. Math., pages 166–178. 1961.
- [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.
- [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.
- [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.
- [LS91] Yves Lafont and Thomas Streicher. Game semantics for linear logic. In *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science (LICS 91)*, pages 43–50, 1991.
- [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 Pawel Urzyczyn, editor, *Typed Lambda Calculi and Applications, TLCA 2005*, volume 3461 of *Lecture Notes in Computer Science*, pages 246–261. Springer-Verlag, 2005.
- [Mac45] G. Mackey. On infinite dimensional vector spaces. *Trans. Amer. Math. Soc.*, 57:155–207, 1945.
- [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.
- [Mét94] François Métayer. Homology of proof nets. *Archive of Mathematical Logic*, 33:169–188, 1994.
- [Pui01] Quintijn Puite. *Sequents and Link Graphs*. PhD thesis, University of Utrecht, 2001.
- [Ret03] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. *Theoretical Computer Science*, 294(3):473–488, 2003.
- [Ros94] Kimmo I. Rosenthal. *-autonomous categories of bimodules. *Journal of Pure and Applied Algebra*, 97(2):189–202, 1994.

- [Sch99] Robert R. Schneck. Natural deduction and coherence for non-symmetric linearly distributive categories. *Theory and Applications of Categories*, 6:105–146, 1999.
- [See89] R.A.G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. *Contemporary Mathematics*, 92, 1989.
- [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.
- [Str05] Lutz Straßburger. On the axiomatisation of Boolean categories with and without medial, 2005. Preprint, available at <http://arxiv.org/abs/cs.LG/0512086>.
- [Str06] Lutz Straßburger. Proof nets and the identity of proofs, 2006. Lecture notes for ESSLLI'06.
- [Sza75] M. E. Szabo. Polycategories. *Comm. Alg.*, 3:663–689, 1975.
- [Thi03] Rüdiger Thiele. Hilbert's twenty-fourth problem. *American Mathematical Monthly*, 110:1–24, 2003.
- [Yet90] David N. Yetter. Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic*, 55(1):41–64, 1990.