The Linear Logic Primer

Vincent Danos

Université de Paris VII
2, Place Jussieu
75251 PARIS Cedex 05
e-mail: danos@logique.jussieu.fr

Roberto Di Cosmo

LIENS-DMI
École normale supérieure
45, Rue d'Ulm
75230 Paris - France
e-mail: dicosmo@dmi.ens.fr

Contents

Preface

Linear Logic was built by Jean-Yves Girard around 1986, and was originally motivated by a deeper investigation of the semantics of λ-calculus. Such investigation led to the surprising discovery that, in *coherence spaces*, the intuitionistic implication, or arrow-type $A \Rightarrow B$ is interpreted, as is now very well known, as $!A \multimap B$, i.e. it is decomposed into two *more primitive* constructs, a *linear* implication $\multimap$ and the modality $!$.

Only afterwards the new connectives were understood from the syntactic point of view.

In these notes, though, we will deliberately present Linear Logic in a very partisan way: we will stick to the syntactical viewpoint, and concentrate all our efforts in familiarizing the reader with the Linear Logic system and the most recent research topics, while referring for the formal semantics to one of the nice presentations that are already available (see for example [Gir87, GLT90]).

These notes were born in Paris, during two weeks of intensive work which allowed the second author to have an exciting tour of Linear Logic, guided by the first one. Some parts are already published and well known, but other ones provide an original presentation of material that can be found mostly in the first author's Phd thesis.

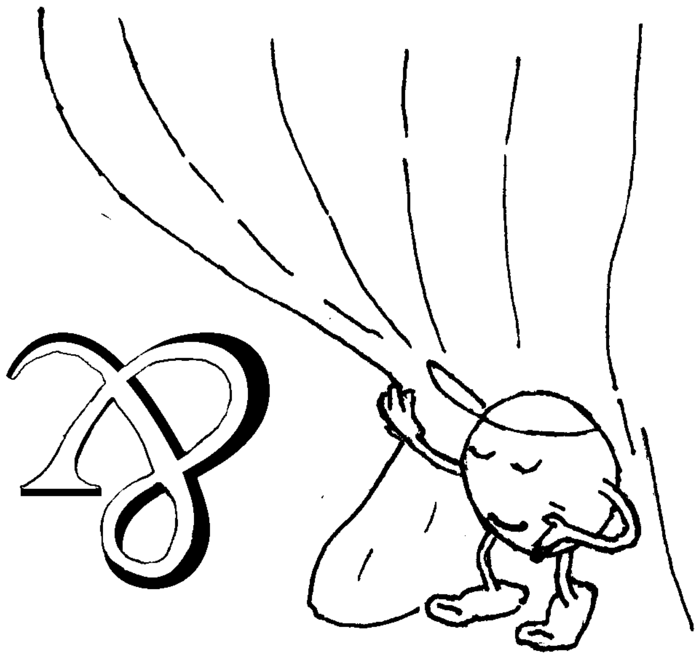
There is a huge amount of exercises: many of them are completely new ones that are especially selected in order to familiarize the reader with some important fine points and are therefore intended as an essential part of the course notes, *not* as a decoration.

Finally, remark that all the presentation given here is inside the world of *propositional* Linear Logic. It is amazing to see how much can be said about propositional calculus, and how much propositional Linear Logic has to say about *computation*, that is here our major concern.

Chapter 1

Introduction

I'm <u>not</u> a linear logician
Jean-Yves Girard
Tokio, April 1996



Linear Logic is a transversal theme: nobody really claims to be purely a linear logician, not even its very inventor, but many people have discovered that Linear Logic provides valuable tools ranging from simply the new language, that conveys better intuitions, to new logical, semantic and algebraic structures. On this tenth anniversary of Linear Logic, even a superficial look at the titles of the over 300 papers in the literature is sufficient to confirm this peculiar aspect: only a few papers are about Linear Logic itself, while the vast majority are examples of how these tools have been fruitfully applied in practically all disciplines of theoretical computer science, and in many cases trickled down to real world applications.

It may well be the case that some problem or application you are currently working at could benefit from the linear logic toolbox, and the goal of this book is precisely to help you find out if this is the case. For this, we think that it is important to provide a lightweight introduction to linear logic, and then give a broad view on the elementary concepts that turned out to be the most fruitful in applications. We will systematically detail the elementary cases to give a thorough understanding of the mechanisms at work, and systematically provide pointers to the relevant literature when dealing with the general case would involve details that obscure the basic structures.

In our survey of linear logic, we will whenever possible take a purely syntactical point of view: this is not because we despise semantics, but because syntax, in particular for this logic, can be more easily accepted and understood by a wide public than its formal semantic underpinnings.

1.1 Linear Logic legitimate ambitions

Let us try to briefly explain why logic took part in the shaping of contemporary computer science, and how linear logic, by giving means to analyze the non-linear treatment of information, offers new tools and ideas in applying logic to computer science.

1.1.1 Logic and Computer Science

Modern logic began when people discussed the possibility of reading 'A implies B' as 'gimme A and I get you B'. They were beginning to suspect something. A proof of such a reading of the implicative statement might be more convincing than a proof of the traditional reading 'B is true whenever A is', and for good reasons: it describes a process by which a proof of B can be produced from a proof of A. What better reason could we think of: it is because my proof of 'A implies B' will get you B, when you give me A, that it is a proof at all !

Logic, or at least proof-theory, sets up formal proof systems: intuitionistic predicate calculus, classical predicate calculus, arithmetics, higher order calculi . . . , i.e., consistent and structured sets of process-building rules. Except that the 'processual' interpretation, if any, has to be made explicit.

Computer science, on the other hand, sets up computational mechanisms: application and substitution, polymorphism, exceptions, methods in object languages, message passing, eval/quote mechanism, variable assignment . . . that is sets of process-building rules. Except that the logic of these processes, if any, has to made explicit.

At a given moment these two sciences met. People realized that the set of implication-only intuitionistic deductions was a core functional language called simply-typed lambda-calculus: the programmming language was a logic, the logic a programming language. This memorable meeting was called the 'Curry-Howard isomorphism'. Since then, 20 or so years ago, logic has ambitions in the field of programming: formal verification, formal semantics, language design, complexity theory . . .

1.1.2 Linear Logic

Linear Logic begins with a further twist in the reading of 'A implies B': read now 'gimme *as many* A as I might need and I get you B'. The notion of *copy* which is so central to the idea of computation is now wired into the logic.

This opens up a new dimension, already intensely exploited, in applications of logic to computer science, and at the same time aims at a mathematical theory within which mathematics are comfortably done. It is both a theory and a tool.

New possibilities show up at two levels, which we might call the language level and the operational level. At the language level the way is open to have:

- *new formulas* expressing refined operational properties: 'gimme A once and I get you B'.

  Applications here range from knowledge representation in IA, refined Logic Programming where the born ability of Linear Logic to represent states is put to use, analysis of Classical Logic and computational interpretations thereof, namely exception mechanisms in programming languages, by means of embeddings in Linear Logic, refined temporal logics, linearity analysis.

- *new rules* expressing constraints on the use of copies resulting in a fragment of Linear Logic for polytime computations to mention only the most spectacular application.

- *new ways* of representing proofs displaying their hidden geometry.

  Here we think of Proof Nets, a parallel syntax for proofs which has been intensively studied, from the graph-theoretic viewpoint to the homological one.

  Proof-nets considerably ease the manipulation of syntax, and gave birth to explicit substitutions systems, themselves useful in the proof of compilers for functional languages, and were also used in linguistics to study parsing in natural languages, again to mention only a few applications.

At the operational level, the way proofs or processes are to be understood operationaly, that is how they are to be computed, there is room to have:

- *new evaluation schemes* of which the most salient aspect is their local and asynchronous treatment of copy.

  Optimal functional evaluation has been understood much better in the light of the 'geometry of interaction' a combinatorial model of proofs for closed computations. Compilations to parallel code are at hand.

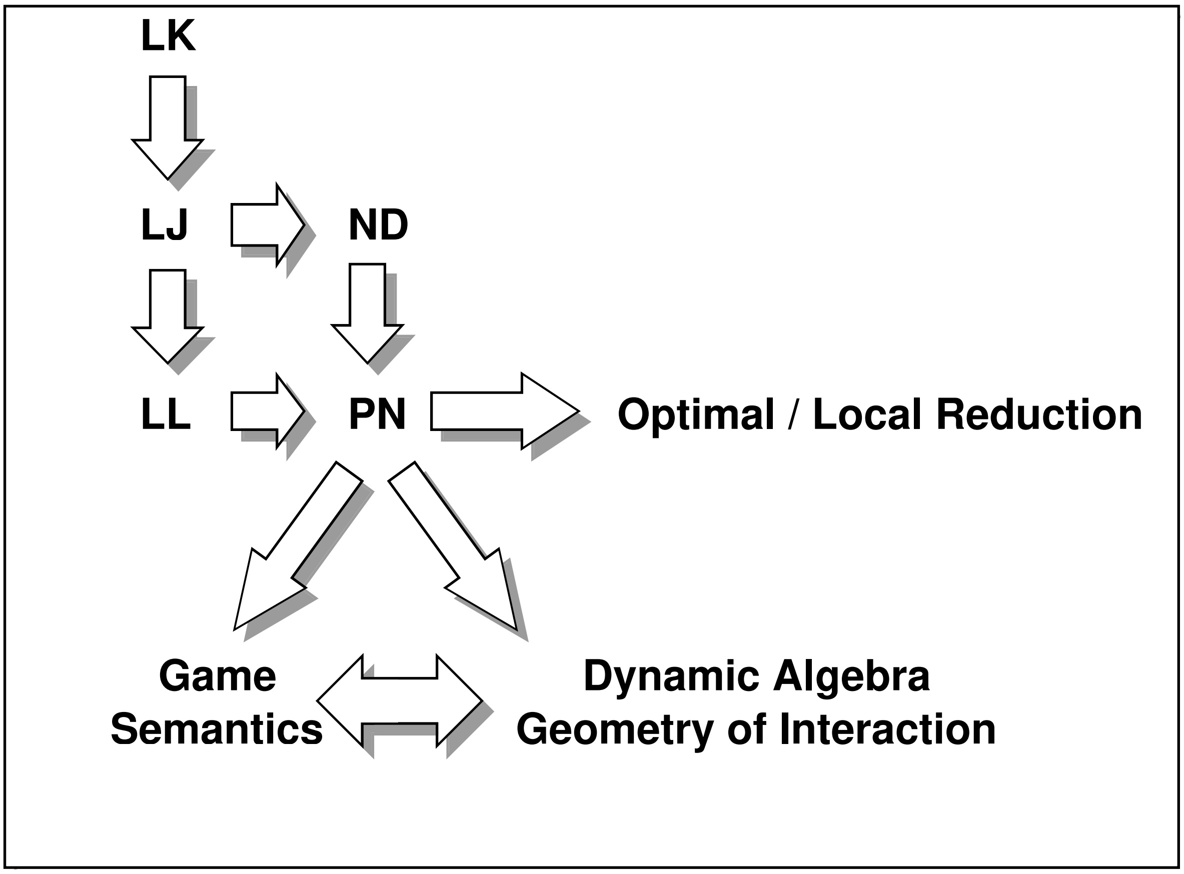
- *new models of processes* capturing in a nice algebraic setting characteristics of evaluation: continuity, stability, and giving tools in proving correction and equivalence of programs.

Linear Logic has lead to the 'best' model of proofs as functions, namely hypercoherences, and has been of crucial help in the new 'games semantics', which solved the long-standing problem of 'full abstraction', that is strategies in those models all come from syntax. Linear Logic also yielded the first model of polymorphism, a logical system where code can be available for objects of different types, anticipating object-oriented programming. Progress has been made also in holding a topological model of programming, bridging the gap with classical mathematical methods.

1.2 Linear Logic and recent advances in theoretical computer science

Linear logic has shown great promise for theoretical computer science ever since its very birth, and over the past years part of these promises (as we will see, mainly, but not exclusively, those concerning a fine analysis of sequential computation) have been fulfilled: optimal reduction, geometry of interaction, fully abstract semantics to cite but a few. In this book, we aim to give a broad overview of what has been accomplished, that should be both accessible for the newcomer and useful for the experienced researcher. Due to the novel nature of the subject, we believe that this goal is best accomplished by exposing in details the results and showing the techniques at work, even if in a simplified framework, rather than flooding the reader with the complex definitions and theorems needed to handle the more complex cases (for those readers that feel confident with the subject after this reading, we give full references to the research papers on the subject).

Here follows a schematic diagram showing the subjects treated in the book, and the chapters where they and the connections between them are treated: there is a clear path from classical sequent calculus (LK) to intuitionistic sequent calculus (LJ) to Linear logic (LL), and from intuitionistic natural deduction (ND) to Proof Nets (PN); then, in the lower part of the diagram we get to the more novel results concerning the algebrization of computation, optimal reduction, game semantics and explicit substitutions. Finally, we will give some taste of interesting fragments of linear logic using ELL.



1.3 Hilbert style or the axiomatic approach

blurb on the axiomatic approach to logical systems and its fitness to model theory.

1.4 Gentzen style or the syntactic approach

blurb on Gentzen's works and the Hauptsatz, plus curry-howard isomorphisms.

1.5 Classical Sequent Calculus

We will give a presentation of Classical Sequent Calculus where we insist upon its symmetries and show the importance of *structural* rules.

Rules of Sequent Calculus are divided into three groups.
Let $\Gamma, \Delta, \ldots$ stand for sequences of formulas.

1.5.1 Identity Rules

The *identity axiom* and the *cut rule*

$$\frac{}{A \vdash A}\,\mathsf{Id} \qquad\qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\mathsf{Cut}$$

simply say that A *is* A.

1.5.2 Structural rules

These rules are concerned with manipulation of contexts. By using them, we can access hypotheses (by *exchange*),

$$\frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta}\,\mathsf{LX} \qquad\qquad \frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'}\,\mathsf{RX}$$

discard hypotheses (by *weakening*)

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\,\mathsf{LW} \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\,\mathsf{RW}$$

or share hypotheses (by *contraction*)

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\,\mathsf{LC} \qquad\qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\,\mathsf{RC}$$

1.5.3 Logical rules

The rules in this group introduce the logical connectors on the left or the right hand side, as in

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\,\mathsf{L}\wedge \qquad\qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\,\mathsf{R}\wedge$$

The rule $\mathsf{L}\wedge$ tells us that the $\wedge$ connective is just an internal notation for the comma on the left. Similarly, we introduce the $\vee$ connective on the left or the right hand side:

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}\,\mathsf{L}\vee \qquad\qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\,\mathsf{R}\vee$$

The rule $\mathsf{R}\vee$ tells us that the $\vee$ connective is just an internal notation for the comma on the right. Notice that we could as well define the introduction rules in another, more economic way:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\,\mathsf{L1}\wedge \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\,\mathsf{L2}\wedge \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\,\mathsf{R}\wedge'$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\,\mathsf{L}\vee' \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}\,\mathsf{R1}\vee \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}\,\mathsf{R2}\vee$$

Notation 1.5.1 (Terminology) *These two distinct styles of managing contexts in the rules are called respectively* multiplicative *and* additive.

Of course, we have now rules for implication and negation, (and for quantifiers if we want to cover predicate calculus, but we will not deal with it here).

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\,\mathsf{L}\neg \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\,\mathsf{R}\neg$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \Rightarrow B \vdash \Delta, \Delta'}\,\mathsf{L}\Rightarrow \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}\,\mathsf{R}\Rightarrow$$

Exercise 1.5.2 (additive cut) *Show that the cut rule* Cut*, in presence of structural rules, can be stated in a more economic, but equivalent way, as*

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,\mathsf{Cut}'$$

Exercise 1.5.3 *Show that in the classical (or intuitionistic*[1]*) framework* $\mathsf{R}\wedge$ *and* $\mathsf{R}\wedge'$ *are equivalent. Similarly, show that rule* $\mathsf{L}\wedge$ *is equivalent to rules* $\mathsf{L1}\wedge$ *and* $\mathsf{L2}\wedge$.

Exercise 1.5.4 *Show that in the classical (or intuitionistic) framework* $\mathsf{L}\vee$ *and* $\mathsf{L}\vee'$ *are equivalent. Similarly, show that rule* $\mathsf{R}\vee$ *is equivalent to rules* $\mathsf{R1}\vee$ *and* $\mathsf{R2}\vee$.

Remark 1.5.5 *Exercises 1.5.3 and 1.5.4 tell us that the choice we make in the presentation of classical (intuitionistic) sequent calculus are just a matter of taste. In Linear Logic, it will no longer be so!*

Exercise 1.5.6 *Show that in the classical framework*[2] *the rules for* $\Rightarrow$ *can be derived from the rules for* $\neg$ *and* $\wedge$ *(or* $\neg$ *and* $\vee$*).*

Notice now that the rules for $\wedge$ are symmetrical with the rules for $\vee$: e.g. $\mathsf{R}\wedge$ is the "mirror image" of $\mathsf{L}\vee$. This is no surprise at all, since a classical sequent is symmetrical and these connectives are just notation for a comma: $\wedge$ on the left and $\vee$ on the right.
Another, deeper symmetry between the introduction and elimination rules for the same connective makes possible to replace a proof looking like

[1]But recall that in intuitionistic sequent calculus you have right to no more than one formula on the right!
[2]In the intuitionistic sequent calculus this fact is not true.

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash A, \Delta} \quad \dfrac{\vdots}{\Gamma' \vdash B, \Delta'}}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}\,\mathsf{R}\wedge \qquad \dfrac{\dfrac{\vdots}{\Gamma'', A \vdash \Delta''}}{\Gamma'', A \wedge B \vdash \Delta''}\,\mathsf{L1}\wedge}{\Gamma, \Gamma', \Gamma'' \vdash \Delta, \Delta', \Delta''}\,\mathsf{Cut}$$

by the following one

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash A, \Delta} \quad \dfrac{\vdots}{\Gamma'', A \vdash \Delta''}}{\Gamma, \Gamma'' \vdash \Delta, \Delta''}\,\mathsf{Cut}}{\Gamma, \Gamma', \Gamma'' \vdash \Delta, \Delta', \Delta''}$$

where the double bar means a sequence of structural rules (weakenings and exchanges).

Exercise 1.5.7 *Formulate such conversions for the three other cases:* $\mathsf{L}\wedge/\mathsf{R}\wedge$*,* $\mathsf{L}\wedge/\mathsf{R}\wedge$*' and* $\mathsf{L}\, i \wedge/\mathsf{R}\wedge$*'.*

Investigating carefully such conversions leads to the central property of viable logics, namely that it is possible to *eliminate the cut rule* from any proof which does not assume extra axioms (Gentzen's Hauptsatz). In other words if a sequent is provable at all then there exists a cut-free proof of it.

Exercise 1.5.8 (Subformula Property) *Show that any cut-free proof of a sequent* $\Gamma \vdash \Delta$ *without proper axioms only mentions subformulas of the formulae in* Γ *and* Δ*.*

Remark 1.5.9 *The Intuitionistic Sequent Calculus can be obtained from the Classical Sequent calculus by constraining the number of formulae on the right of a sequent to at most one. This calculus also enjoys cut-elimination. In the intuitionistic case the process of cut-elimination is very similar to* β*-reduction in* λ*-calculus and promotes logic as a paradigm of computation (see [GLT90] for more details). In the classical case, the computational meaning of this process is still a matter of passionate investigation.*

Exercise 1.5.10 *Show that from any intuitionistic proof of* $\vdash A \vee B$ *one can extract a proof of* A *or a proof of* B*. Notice that this is the case because of the forbidding of the right contraction rule, implicit in the restriction on the right hand side of an intuitionistic sequent. So control over structural rules yields new properties; linear logic will impose the most severe control on these rules.*

Exercise 1.5.11 *State the natural conversion associated to contraction; one can see that this rule is responsible for the combinatorial explosion of cut-elimination.*

Notation 1.5.12 (Terminology) *It is common in the literature to find the following notation:*

LK, CSC the Classical Sequent Calculus

LJ, ISC the Intuitionistic Sequent Calculus

1.5.4 Weakening and contraction: bad bookkeepers

Now, all this is very nice, but before being satisfied with such a formalization of the laws of thought, let's look at the contraction and weakening rules from the point of view of daily life in a capitalistic world.

There are several reasons to get rid of weakening and contraction as structural rules, but we can start with a very simple and intuitive one. If we look at a sequent $\Gamma \vdash \Delta$ from the point of view of classical logic, it reads "from the truth of Γ follows the truth of Δ", but from a capitalistic point of view, we can read it as " if you give me the goods in Γ, I will give you one of the goods in Δ".

Now, classically, the contraction rule says that if you derive the truth of Δ using several occurrences of an assumption A, you can as well derive it with just one assumption: classical logics is interested in knowing whether A exists in the assumptions (i.e. wheter it is true or not), and not in how many times it occurs.

Similarly, weakening says that once you have the truth of Δ it does not harm to add some extraneous assumptions: Δ will stay true, and who cares some additional hypothesis.

Now, let's take the capitalistic point of view:

Contraction reads more or less this way: "if you can buy one of the goods in Δ using several goods A, then you can buy it with just one A".

And weakening becomes: "if you can buy one of the goods in Δ with Γ, then it does not harm to spend some more A's without getting anything more in change".

It is as easy to be convinced that the capitalistic reading would not be accepted by any sane merchant (that is not likely going to make you any present, as contraction would imply), nor by any careful buyer (that would not follow the advice given by weakening to waste carelessly his money for nothing).

Weakening and contraction as used in classical logic are bad bookkeepers: they make us loose control on our expenses, and hence waste our resources.

When the control of the use of some resource A is important, we need to do something to modify the capitalistic meaning of such structural rules: either we keep them (because we are fond of classical logic, and ready to waste our money for it) and we spend a lot of effort to formalize with a series of extra-logical axioms the fact that assuming A has a cost, or we can simply drop these rules, hoping to recover them in a simple way when dealing with resources that cost nothing.

Once we decide to be careful about resources, we start to be suspicious about the traditional $\wedge$ and $\vee$ connectives too, and we discover that each of them is now split in two. In fact, without weakening and contraction, we can no longer show equivalent the two formulations of the left introduction rules for $\wedge$ ($\mathsf{R}\wedge$, $\mathsf{R}\wedge'$) and $\vee$ ($\mathsf{L}\vee$, $\mathsf{L}\vee'$): actually exercises 1.5.3 and 1.5.4 require some weakenings and some contractions to be solved. This means that the different rules are describing *different connectives*, and we have now two conjunctions and two disjunctions.

This leads us to Linear Logic.

1.6 Linear Sequent Calculus

In Linear Logic, we have no longer the right to forget hypotheses or use them more than once, (no *weakening* and no *contraction*). This does not affect *Exchange* and *Identity* (while *cut* requires some care), but the classical $\wedge$ and $\vee$ connectives are now split in two.

We have *two* conjunctions: a *tensor product* (or *cumulative conjunction*: it corresponds to the "non-economic" presentation of the classical $\wedge$)

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\ \mathsf{L}\otimes \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\ \mathsf{R}\otimes$$

and a *direct product* (or *alternative conjunction*: it corresponds to the "economic" presentation of the classical $\wedge$):

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \& B \vdash \Delta}\ \mathsf{L1}\& \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \& B \vdash \Delta}\ \mathsf{L2}\& \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \& B, \Delta}\ \mathsf{R}\&$$

Symmetrically, we get the rules for the *two* disjunctions: a *direct sum*: it corresponds to the "economic" presentation of the classical $\wedge$):

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta}\ \mathsf{L}\oplus \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \oplus B, \Delta}\ \mathsf{R1}\oplus \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \oplus B, \Delta}\ \mathsf{R2}\oplus$$

and a *tensor sum*: it corresponds to the "non-economic" presentation of the classical $\vee$)

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'}\ \mathsf{L}\parr \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \parr B, \Delta}\ \mathsf{R}\parr \quad .$$

The rule for negation stays as in the classical case, and the negation operator (that is now noted $\perp$) is still involutive (i.e. $A^{\perp\perp}$ is the same as A).

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta}\ \mathsf{L}\perp \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}\ \mathsf{R}\perp$$

What about the cut rule then? Here we need some care, as the following remark and exercise show.

Remark 1.6.1 *The additive rule* Cut*' does not allow to internalize the* $\vdash$ *symbol: actually, it is not possible with such a cut rule to prove*

$$\frac{\vdash A \quad A \vdash B}{\vdash B}$$

Exercise 1.6.2 (Schellinx) *Show that adding the additive cut rule yields the same theory as adding the axiom* $A \oplus A^{\perp}$*; infer that this alternative cut cannot be eliminated.*

Hence, for Linear Logic we will keep the multiplicative cut rule Cut.
This presentation of a propositional fragment of Linear Logic based on symmetrical sequents is well suited to show how it can be derived from the classical frameworks by elimination of the *weakening* and *contraction* rules. It also allows us to define the *classical* and *intuitionistic* linear fragment in the same way as for the usual Gentzen calculus.

Notation 1.6.3 (Terminology) *In the literature, one can find the system above, with also rules for the constants and the* exponential *connectives, used to present what is usually called Classical Linear Logic (or* **CLL***) and distinguish it from Intuitionistic Linear Logic (or* **ILL***), where at most one formula is allowed to the right.*

Anyway, we can already see, just with these connectives, that this presentation of the system is very redundant: due to the symmetries (already pointed out in the classical case) present in the rules, there are many ways (using negation) to express some rules in terms of some other ones. This fact can be used to give a more concise presentation of the system, so let's restart from scrath and do the work more properly!

- We give atomic formulae in two forms: A and $A^{\perp}$. Then we say that the negation of A *is* $A^{\perp}$ and the negation $A^{\perp\perp}$ of $A^{\perp}$ *is* A.
- We *define* negation of non-atomic formulae by use of the De Morgan rules:

$$\begin{array}{rclcrcl} (A\otimes B)^{\perp} & = & A^{\perp}⅋B^{\perp} & \qquad & (A\&B)^{\perp} & = & A^{\perp}\oplus B^{\perp} \\ (A⅋B)^{\perp} & = & A^{\perp}\otimes B^{\perp} & & (A\oplus B)^{\perp} & = & A^{\perp}\&B^{\perp} \end{array}$$

- We *define* linear implication A $\multimap$ B as a shorthand for $A^{\perp}⅋$B.
- We convert a symmetric sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ into an asymmetric right-only form by use of linear negation $\vdash A_1^{\perp}, \ldots, A_n^{\perp}, B_1, \ldots, B_m$

Exercise 1.6.4 *Check that the De Morgan's identities above correspond to linear equivalences in the two sided version of the calculus.*

Now, the identity axiom becomes $\vdash A^{\perp}, A$ and also subsumes the same time the rules for linear negation. The cut rule is transformed into:

$$\frac{\vdash A, \Gamma \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\,\mathsf{Cut}$$

There is only one *exchange* rule that subsumes the left and right one of the symmetric presentation.

$$\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}\,\mathsf{X}$$

and the logical rules are now expressed by:

$$\frac{\vdash A, B, \Gamma}{\vdash A⅋B, \Gamma}\,⅋ \qquad\qquad \frac{\vdash A, \Gamma \quad \vdash B, \Delta}{\vdash A\otimes B, \Gamma, \Delta}\,\otimes$$

$$\frac{\vdash A, \Gamma}{\vdash A\oplus B, \Gamma}\,1\oplus \qquad \frac{\vdash B, \Gamma}{\vdash A\oplus B, \Gamma}\,2\oplus \qquad \frac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A\&B, \Gamma}\,\&$$

This is much better then the previous presentation! We can now continue with the full system of Propositional Linear Logic.
We introduce units:

$$\mathbf{1}^\perp = \perp \qquad \perp^\perp = \mathbf{1} \qquad \top^\perp = \mathbf{0} \qquad \mathbf{0}^\perp = \top$$

$$\frac{\vdash \Gamma}{\vdash \perp, \Gamma}\perp \qquad \frac{}{\vdash \mathbf{1}}\mathbf{1} \qquad \text{(no rule for } \mathbf{0}\text{)} \qquad \frac{}{\vdash \top, \Gamma}\top$$

Exercise 1.6.5 *Check that actually* $\mathbf{1}$ *is an identity for* $\otimes$*,* $\perp$ *for* $⅋$*,* $\top$ *for* $\&$ *and* $\mathbf{0}$ *for* $\oplus$*.*

And finally, we allow two modalities $!\,A$ (*of course* A) and $?\,A$ (*why not* A) to deal with non-capitalistic resources:

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

$$\frac{\vdash \Gamma}{\vdash ?\,A, \Gamma}\mathsf{W?} \qquad \frac{\vdash ?\,A,\ ?\,A, \Gamma}{\vdash ?\,A, \Gamma}\mathsf{C?} \qquad \frac{\vdash A, \Gamma}{\vdash ?\,A, \Gamma}\mathsf{D?} \qquad \frac{\vdash A,\ ?\,\Gamma}{\vdash !\,A,\ ?\,\Gamma}\,!$$

Exercise 1.6.6 (Terminology) *Prove the following linear equivalences:*

$$!(A\&B) \equiv (!A)\otimes(!B) \quad ?(A\oplus B) \equiv (?A)⅋(?B).$$

So that is the reason why $\otimes$ *and* $⅋$ *are called* multiplicatives*, while the* $\&$ *and* $\oplus$ *are called* additives *and ! and ?* exponentials.

Exercise 1.6.7 *Show that the aforementioned fundamental isomorphisms are indeed isomorphisms of vector spaces if read with the following dictionary:* $E^\perp$ *is the dual of* E*,* $\otimes$ *and* $⅋$ *are both the tensor product,* $\&$ *and* $\oplus$ *the direct sum and both modalities the symmetric algebra.*

Remark 1.6.8 *The rule for ! imposes constraints on the context, as the rule for* $\&$*. The rules for exponentials carry some resemblance to the rules for the modal connectives for* necessity *and* possibility *(*$\Box$ *and* $\Diamond$*).*

Notation 1.6.9 (Terminology) *It is common in the literature to find the following names for different fragments of Linear Logic.*

LL The full propositional fragment of Linear Logic

MLL The multiplicative fragment of Linear Logic (no exponentials)

MALL The multiplicative and additive fragment of Linear Logic (no exponentials)

MELL The multiplicative fragment of Linear Logic with exponentials.

Linear logic also enjoys cut-elimination and subformula property: state the conversions (reduction moves) for each pair of dual connectives : $\otimes$ and $\parr$, $\&$ and $\oplus$, ? and !.
Cut-elimination is now (as opposed to the case of classical logic) a deterministic process; and indeed full LL can be seen as a computationaly sound completion of *MALL*.

Exercise 1.6.10 *Notice that the commas are multiplicative: show that* $\Gamma \vdash \Delta$ *if and only if* $\otimes\Gamma \vdash \parr\Delta$*. Does the same hold for* $\&\Gamma \vdash \oplus\Delta$*?*

Exercise 1.6.11 (invertible connectives) *Prove that the* $\parr$ *is invertible, i.e., that* $\Gamma, A\parr B$ *is provable iff* Γ, A, B *is provable.*
State and prove (resp. disprove) a similar property for the $\&$ *(resp.* $\otimes$ *and* $PLUS$*).*
Prove also that the $!$ *is invertible if all formulae of the context are prefixed by a* $?$ *modality.*

Exercise 1.6.12 (Unprovable formulae)

1. *Show that the following sequents are unprovable in LL, although they are classically valid substituting* $\wedge$ *to* $\otimes$ *(check).*

$$\vdash A\otimes A, A^\perp\otimes A^\perp \text{ and } \vdash A\otimes B, A^\perp\otimes B, A\otimes B^\perp, A^\perp\otimes B^\perp$$

$$\vdash (A\otimes A)\multimap A \text{ and } \vdash A\multimap(A\otimes A)$$

2. *Show also that* $\vdash A^\perp\oplus A$ *is not provable. So it is of no interest to define an additive implication: compare with 1.6.1.*

Exercise 1.6.13 (Girard) *Suppose one adds a new pair of modalities, !', ?', with the same rules as for the standard ones. Show you cannot prove the equivalence of this new pair with the old one.*

Now let's see that the rules for exponentials capture exactly *weakening* and *contraction*.

Exercise 1.6.14 (Schellinx)

1. *Show that if* $A\multimap(\mathbf{1}\&(A\otimes A))$ *then left* weakening *and* contraction *rules are* admissible *for* A*.*
2. *Show that if* $A\multimap\,!\,A$ *then* $A\multimap(\mathbf{1}\&(A\otimes A))$*.*
3. *Open problem: does the converse hold? Try to state the problem.*

Exercise 1.6.15 (Joinet) $*$

1. *Prove that* $!!A$ *is linearly equivalent to* $!A$*.*
2. *Prove there are only seven modalities (combinations of the two modalities ! and ?) up to linear equivalence; draw the lattice.*

Exercise 1.6.16 (Danos-Joinet-Schellinx) $*$
The purpose of this long exercise is to state an admissible extension of the contraction-rule in LL. We shall use $\diamond A$ to denote one of A and $?A$, and $\bullet$ to denote any of the connectives $\otimes, \parr, \&, \oplus$. Let the notions of positive, negative subformulas of a linear formula F be given as usual. Say that a linear formula G is a positive (resp. negative) decoration of F iff G has been obtained from F by questioning some (maybe none) positive (resp. negative) subformulas of F .
a) Let F be any formula, F^+ be any positive and F^- any negative decoration of F. By induction on F show: $F \vdash F^+$ and $F^- \vdash F$.
b) Show that one turns the set $\mathsf{D} + (F)$ of all positive decorations of F into a lattice with the ordering $A \leq B$ iff the decoration of B is equal to or extends that of A; the join $A \sqcup B$ of two elements A and B being the surperposition of both decorations.
c) So by a) $A \leq B$ entails $B \vdash A$, is the converse true ?
d) show with the help of a) that the "extended" dereliction:

$$\frac{\Gamma \vdash F', F'', \Delta}{\Gamma \vdash \diamond(F' \sqcup F'') \vdash \Delta}$$

where F', F'' are in $\mathsf{D}^+(F)$ and $\diamond =?$ iff neither F' nor F'' is questioned, is admissible.

Exercise 1.6.17 (Girard) *Assign an integer $v(p_i)$ in $\boldsymbol{Z}$ to every atomic symbol p_i, and then extend this valuation to non atomic formulae as follows:*

$$v(p_i^\perp) = 1 - v(p_i) \qquad v(\mathbf{1}) = 1 \qquad v(\perp) = 0$$

$$v(A \otimes B) = v(A) + v(B) - 1 \qquad v(A \parr B) = v(A) + v(B).$$

1. *find a relation between $v(A)$ and $v(A^\perp)$*
2. *show that if A is provable then v(A) = 1*
3. *find a formula A (without $\mathbf{1}$ or $\perp$) that is not provable and such that $v(A) = 1$ for every possible valuation v.*

Exercise 1.6.18 (Girard) *Let's focus now on those formulae of MALL that are closed, i.e. the ones built out of $\mathbf{1}$ and $\perp$ only. Consider now the classical valuation $c(\mathbf{1}) = \top, v(\perp) = \perp$ extended to non atomic formulae as follows:*

$$v(A \otimes B) = c(A) \wedge c(B) \qquad c(A \parr B) = c(A) \vee v(B).$$

1. *show that $\mathbf{1} \parr \mathbf{1}$, $\perp$ are not provable, and deduce that $\perp \otimes (\mathbf{1} \parr \mathbf{1})$ is not provable*
2. *show a closed formula A that is not provable, classically false (i.e. $c(A) = \perp$) and such that $v(A) = 1$; show also a classically true formula B such that $v(B) \neq 1$.*
3. *find for every $n \in \boldsymbol{Z}$ a closed formula A_n such that $v(A_n) = n$ and $c(A_n) = \top$*
4. *show a non provable closed formula A s.t. $v(A) = 1$ and $c(A) = \top$*

Exercise 1.6.19 (Girard) *We still stay in the world of closed formulae.*

1. *show a formula A s.t.* $(A \otimes A^{\perp}) \parr \mathbf{1}$ *is not provable*
2. *given any set* E *and a valuation* $\phi(A) \in E$ *for all closed formulae* A *that enjoys the following properties:*
   - *if* $\phi(A) = \phi(B)$ *then* $\phi(A^{\perp}) = \phi(B^{\perp})$
   - *if* $\phi(A) = \phi(A')$ *and* $\phi(B) = \phi(B')$ *then* $\phi(A \otimes B) = \phi(A' \otimes B')$

   Show that for every $e \in E$ *the set of closed formulae* A *s.t.* $\phi(A) = e$ *is different from the set of provable formulae.*
3. *in particular, recover the results of 1.6.18, i.e. the fact that* $v(A) = 1$ *and* $c(A) = \top$ *are not sufficient conditions for provability*

The last point of the exercise above tells us that there is no hope to find a "simple" valuation catching multiplicative truth. This is a hint that multiplicative provability is a difficult problem (in terms of complexity). Indeed Kahnovich (ITLI Preprint, Amsterdam) proved recently that *MLL* provability is NP-complete, Lincoln-Mitchell-Shankar (in FOCS'90) proved that *MALL* is PSPACE-complete, they proved furthermore in a joint work with Scedrov that full LL is undecidable (Symp. Found. of C. Sc. 90).

1.7 Informal Semantics

Before starting our study of Linear Logic, that involves a series of relevant syntactic results, we will now try to give an informal semantics of Linear Logic, and try to understand what these linear connectives are by using that capitalistic point of view that already helped us to be suspicious about the *weakening* and *contraction* rules, otherwise so innocuous-looking.
This will help us to familiarize with Linear Logic, and will hopefully convey enough intuitive meaning to make these new entities $\otimes$, $\parr$, $\&$, $\oplus$ etc. as natural as (or *more natural than*) the usual classical connectives.

1.7.1 Linear connectives: useful devices to read your menu

Linear implication and the four basic connectives that linear logic supllies in change of the traditional $\wedge$ and $\vee$ are really nothing new to the people that is used to go frequently to restaurants that offer fixed-price menu's. Let's have a look at a typical one (as in [Gir90]).
We can easily formulate this menu in linear logic as

$$75\mathrm{FF}^{\perp} \parr (Q \& S) \otimes (P \& F) \otimes ((B \oplus R \oplus O \oplus A) \& (M \& G \& T))$$

The $\parr$ connective tells us that if we give in 75 Frs (that sadly disappear immediately afterwards: the fact that we *give* in the francs is expressed by the little $\perp$) for one "entree" *and* one "plat" *and* either a "dessert" or some "fruit". So a $\parr$ connective somewhere in a formula tells us that we are faced with a trading situation: $A^{\perp} \parr$B means we can get A *or* B in the very particular sense that if we got an A somewhere (we are not too poor) either we keep this A *or* we get a B by exchanging it with our A. We cannot get both, because we live in a capitalistic world. On the other hand, we

Menu a 75 Frs
Entree
quiche lorraine
ou
saumon fume
et
Plat
pot-au-feu
ou
filet de canard
et
Fruit
selon saison (banane ou raisin ou oranges ou ananas)
ou
Dessert au choix (mistere, glace, tarte aux pommes)

Figure 1.1: The linear Menu

are pretty sure that we will get both one "entree" and a "plat", that is formalised by the $\otimes$ linear connective.
Now, let's keep looking at the menu. As an "entree" we can have a "quiche lorraine" or a "saumon fume", but not both. Here, anyway, the connective we use is not a $\parr$, but a $\&$: we are in a different situation that allows us to get a "quiche lorraine" *or* a "saumon fume", at our will, but to get the "quiche lorraine" we are not obliged to give in a "saumon fume". After all, we went in the restaurant to get some food, not to give food away! Similarly for the "plat". Then we can choose to have one (but only one) between a "Dessert" or a "fruit" (again a $\&$), but here some other peculiar phenomenon happens: we know that we can get only one of the fruits listed in the Menu, but now we cannot decide which one, not even by trading something in exchange. Hence we cannot use a $\otimes$ to formalize it (we get only one fruit, not all), nor a $\&$ or a $\parr$ (we cannot choose the one we want, not even by trading), so here comes to our rescue the last connective of linear logic: the $\oplus$. This connective tells us that the object we will get from A$\oplus$B is either an A or a B, but nothing more.