Handbook of Linear Logic

November 2, 2023

Contents

Chapter 1

Instructions for authors

The source for this chapter is in `chapter-instructions/chapter.tex`. To compile this chapter as a standalone document, run LaTeX on file `standalone.tex` in the same folder.

1.1 Starting to work

The files for each chapter are collected in a specific folder `chapter-`*`something`*: this includes the LaTeX source (`chapter.tex`), but also images, included `.tex` files, local macro definitions, *etc.*

Compiling the file `standalone.tex` in the same folder should produce a document containing only what is included in the `chapter.tex` file.

The authors of a given chapter should feel free to tinker with the source files in the dedicated folder.

1.2 Macros

The common macros for the entire book are collected in various files in the `lib` folder of the repository.

If you need more macros, you can add them to the `localmacros.tex` file of your chapter folder (create it if necessary). These may be reviewed and discussed for inclusion in the unified set of macros.

For instance: *this macro is defined locally*

1.3 Linear logic symbols

We use the `cmll` package from Manu Beffara for linear logic symbols. Additional symbols may be added to `lib/llsymbols.sty` in the future.

Linear negation	`\lneg`	$A^\perp$
Binary tensor	`\tensor`	$A \otimes B$
n-ary tensor	`\tensop`	$\otimes(A, B, C)$

1.4 Proof trees

We use the `ebproof` package from Manu Beffara for typesetting proof trees. It is similar in spirit to the venerable `bussproof` package, but it is somehow more flexible, developed using modern LaTeX programming discipline... and we have the developer at hand. Beware that the latest (May 2017) version of `ebproof` relies on quite recent features from the LaTeX3 project, so be sure to upgrade your LaTeX distribution to the latest version.

In order to have consistent notations for rule names, we use an overlay to `ebproof`, that is defined in `lib/llproofs.sty`. You can find (and alter) the list of available rules at the bottom of the file: each line has the shape

```
\DeclareRule name arity symbol
```

where *`name`* is the identifier of the rule, *`arity`* is the number of premises of the rule, and *`symbol`* is used to label the rules in proof trees as well as to refer to the rule in running text. After such a declaration, a call to

```
\apply name conclusion
```

is equivalent to

```
\infer arity [symbol] conclusion
```

and a call to

```
\rulename name
```

renders the label of the rule (*i.e.* *`symbol`* between parentheses) in running text. The general syntax for `\apply` and `\rulename` is in fact a bit more complicated to tackle some corner cases (using another arity than the default, passing optional arguments to `\infer`, tweaking the symbol, *etc.*): refer to the source if necessary.

To get the flavour of this system, consider the multiplicative fragment. The rules are declared as follows:

```
\DeclareRule{axiom} 0 {\mathit{ax}}
\DeclareRule{cut} 2 {\mathit{cut}}
\DeclareRule{tensor} 2 {\tensor }
\DeclareRule{parr} 1 {\parr }
```

so that

```
\begin{prooftree*}
 \apply{axiom} {A,\lneg A}
 \apply{parr}  {A\parr\lneg A}
```

```
 \apply{axiom} {A,\lneg A}
 \apply{axiom} {A,\lneg A}
 \apply{tensor}{A,\lneg A\otimes A,\lneg A}
 \apply{cut}   {A,\lneg A}
\end{prooftree*}
```

renders as

$$\dfrac{\dfrac{\dfrac{}{A, A^\perp}\,(ax)}{A \parr A^\perp}\,(\parr) \qquad \dfrac{\dfrac{}{A, A^\perp}\,(ax) \qquad \dfrac{}{A, A^\perp}\,(ax)}{A, A^\perp \otimes A, A^\perp}\,(\otimes)}{A, A^\perp}\,(cut)$$

and we can refer to rules (ax), (cut), $(\otimes)$ and $(\parr)$ using `\rulename{axiom}`, `\rulename{cut}`, `\rulename{tensor}` and `\rulename{parr}` respectively.

1.5 Other notations and conventions

1.5.1 Labels and refs

When introducing a label via the `\label` macro use the following convention for the label:

```
\label{<chapter name>:<environment type>:<label id>}
```

where:

- `<chapter name>` is the short name of the current chapter, which is used to identify the file containing the chapter; for example `coh` for the chapter contained in `chapter-coh/chapter.tex`;
- `<environment type>` defines the type of LaTeX object being labelled and is:
  - `chapter` for a `\chapter`;
  - `sec` for a `\section`;
  - `ssec` for a `\subsection`;
  - `sssec` for a `\subsubsection`;
  - `eq` for an `equation`;
  - `fig` for a `figure` environment's caption;
  - `enum` for an `item` in an `enumerate` environment;
  - `def` for a `definition` environment;
  - `rem` for a `remark` environment;
  - `exo` for an `exercise` environment;
  - `exa` for an `example` environment;

  - `lem` for a `lemma` environment;
  - `prop` for a `proposition` environment;
  - `theo` for a `theorem` environment.
  - `coro` for a `corollary` environment.

- `label id` is the name of the label, making it unique in the whole document. It may contain white spaces and semicolons.

1.5.2 Defining new terms

When introducing a notion for the first time, use the `\definitive` macro from `lib/llnotations.sty`: `\definitive{my new notion}` typesets its argument in italics and adds it to the index.

1.5.3 Logical systems

Logical systems are rendered using the `\LogicalSystem` macro from `lib/llnotations.sty`. Some systems are predefined there: MLL, MELL, *etc.* The corresponding macros admit an optional argument that is rendered as a subscript: `\MALL[0]` renders as MALL_0.

1.6 Bibliography

TBD.

Part I

Proofs

Chapter 2

MLL

Chapter 3

Sequents

This chapter presents the language and sequent calculus of second-order Linear Logic and the basic properties of this sequent calculus. The core of the chapter uses the two-sided system with negation as a proper connective; the one-sided system, often used as the definition of Linear Logic, is presented later and used for describing the cut elimination procedure.

rewrite this later.— Alexis

	Positive		*Negative*	*Class*
α	atom	$A^\perp$	negation	
$A \otimes B$	tensor	$A \parr B$	par	multiplicatives
$\mathbf{1}$	one	$\perp$	bottom	multiplicative units
$A \oplus B$	plus	$A \,\&\, B$	with	additives
$\mathbf{0}$	zero	$\top$	top	additive units
$!A$	of course	$?A$	why not	exponentials
$\exists \xi . A$	there exists	$\forall \xi . A$	for all	quantifiers

Table 3.1: Formulas of Linear Logic.

3.1 Formulas

We must fix notations for references within the text:

- chapters should be numbered with roman numerals;
- all references should include section numbers;
- chapter numbers should appear only in references outside of the current chapter.

For instance:

- This section must be named **1 Formulas** (and not **III.1 Formulas**).
- The first table in this section should be named Table 1.1 (and not Table 1, nor Table III.1.1).
- A reference to this table within the present chapter should be given as Table 1.1.
- A reference to this table within another chapter (or in the table of tables) should be given as Table III.1.1.

We should moreover encourage a systematic form for references (e.g, to avoid mixing Fig. and Figure, or having a reference to Lemma 2.2, when our 2.2 statement is a Proposition) by providing suitable macros. When this is done, proper instructions should be added to `chapter-instructions`. — Lionel

The **formulas** of Linear Logic are defined by Table 3.1. Capital Latin letters A, B, C will range over the set of formulas. **Atomic formulas**, written α, β, γ, are predicates of the form $p(t_1, \ldots, t_n)$, where the t_i are terms from some first-order language. The predicate symbol p may be either a predicate constant or a second-order variable, we call n the **arity** of p. By convention we will write first-order variables as x, y, z, second-order variables as X, Y, Z, and ξ for a variable of arbitrary order.

Each line of Table 3.1 (except the first one) corresponds to a particular class

of connectives, and each class consists in a pair of connectives (which are said to be **dual** of each other). Those in the left column are called positive and those in the right column are called negative (see Section 3.6 for practical impacts of the notion of polarity). The atoms have no predefined polarity and negation changes the polarity of the negated formula. The *tensor* and *with* connectives have conjunctive flavour while *par* and *plus* have disjunctive flavour. Indeed mapping $\otimes$ and & to $\wedge$ and $\parr$ and $\oplus$ to $\vee$ turns linear provable formulas into valid classical formulas (see Section 3.8). The exponential connectives are also called **modalities**, and traditionally read *of course* A (or *bang* A) for $!A$ and *why not* A for $?A$. Quantifiers may apply to first- or second-order variables.

It might be useful to fix one — Olivier

conjunction/ disjunction: does not mean much with no more details: refer to multiplicative/additive LK inferences? Semantics? provable formulas? — Alexis

The *linear implication* and the *linear equivalence* are presented as defined multiplicative connectives, by $A \multimap B := A^\perp \parr B$ and $A \circ\!\!-\!\!\circ B := (A^\perp \parr B) \otimes (A \parr B^\perp)$, respectively. In order to underline the symmetries acting on Linear Logic formulas, we consider the implication and equivalence as defined connectives, similarly to the decomposition $A \to B = \neg A \vee B$ in classical logic. Notice that $A \multimap B$ and $A \circ\!\!-\!\!\circ B$ are defined by the multiplicative connectives, in fact their additive versions are not suitable, for example the disjunction $A^\perp \oplus A$ is not provable for all formula A.

Free and bound variables and first-order substitution $A[t/x]$ are defined in the standard way. Formulas are always considered up to renaming of bound names. If A is a formula, X is a second-order variable of arity n and $B[x_1, \dots, x_n]$ is a formula with variables among x_i, then the formula $A[B/X]$ is A where every atom $X(t_1, \dots, t_n)$ is replaced by $B[t_1/x_1, \dots, t_n/x_n]$. For example, $(\forall y.X(y))[\forall z.p(x,z)/X] = \forall y.\forall z.p(y,z)$.

For a book, I think we should define this, don't you think? — Alexis

3.2 Sequents and proofs

A **sequent** is an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite sequences of formulas. For a sequence $\Gamma = A_1, \dots, A_n$, the notation $\$\Gamma$, for $\$ \in \{?, !\}$, represents the sequence $\$A_1, \dots, \$A_n$, and similarly $\Gamma^\perp$ represents the sequence $A_1^\perp, \dots, A_n^\perp$.

Table 3.2 gives a picture of the LL inference rules together with the labelling of the inference name. These latter are oriented top-down: the sequents at the top of an inference rule are called **premises** and the one at the bottom is called **conclusion**. The **arity** of a rule is the number of its premises, for example the axiom has arity 0, the cut has arity 2 and the two rules introducing the negation have arity 1. The occurrences of a formula that are explicit in the picture of an inference rule in Table 3.2 are called **active**, the other occurrences being **passive**. The passive occurrences provide the **context** of the rule. For the rules introducing a new occurrence of a connective, the occurrence of formula containing this connective is called the **principal** occurrence of the rule. A principal occurrence is unique in a rule and always occurs in the concusion. For example in the right introduction rule of the tensor connective, the explicit occurrences of A and B are active in the premises and the explicit occurrence of $A \otimes B$ is active in the conclusion (it is moreover the principal occurrence),

explain — Olivier

while all occurrences of the formulas in the sequences Γ, Γ', Δ and Δ' are passive (they constitute the context).

Observe that the rules $(\exists^1_L)$ and $(\exists^2_L)$ (resp. $(\forall^1_R)$ and $(\forall^2_R)$) differ only by the order of the quantified variable: we may write $(\exists_L)$ (resp. $(\forall_R)$) for either rule. Similarly the rules $(\exists^1_R)$ and $(\exists^2_R)$ (resp. $(\forall^1_L)$ and $(\forall^2_L)$) differ only by the order of the quantified variable and the substitution performed in the premise: again, we may write $(\exists_R)$ (resp. $(\forall_L)$) for either rule, when the order is either irrelevant or clear from the context.

The left (resp. right) introduction rule for the ! (resp. ?) modality is called *dereliction*, and the right (resp. left) introduction rule for the ! (resp. ?) modality is called *promotion*. Notice that the promotion rules require that all passive formulas have a suitable exponential modality. Applying the right (resp. left) introduction rule for the ! or $\forall$ (resp. ? or $\exists$) connective thus requires constraints on the context. These rules are called *contextual*.

an inductive definition wouldn't be simpler? — Olivier

improve this — Alexis

define it — Alexis

Proofs are labelled trees with nodes labelled with inference rules and edges labelled by sequents[1].

Alternatively, an LL proof of a sequent s can be defined inductively as the data of

1. an instance of an n-ary inference rule of Table 3.2, $\dfrac{s_1 \quad \dots \quad s_n}{s}\ (r)$, and
2. a family of n proofs $(\pi_i)_{1\leq i\leq n}$, such that π_i is a proof of s_i for $1 \leq i \leq n$,

that we write $\dfrac{\pi_1 \quad \dots \quad \pi_n}{s}\ (r)$ (or $\dfrac{\begin{matrix}\pi_1 & & \pi_n\\ s_1 & \dots & s_n\end{matrix}}{s}\ (r)$ when one wants to emphasize the premises of the last rule).

Notation 3.2.1. We shall write $\pi : \Gamma \vdash \Delta$ to signify that π is a proof with conclusion $\Gamma \vdash \Delta$.

In a proof considered as a tree, the leaves are given by applications of the rules (ax), $(\mathbf{1}_R)$, $(\bot_L)$ and $(\top_R)$. By allowing arbitrary sequents as (non-justified) leaves, one gets the notion of **open proof** (or partial proof). These special leaves are called **holes** (with the idea that plugging a proof with appropriate concluson in each of the holes of an open proof gives you a proof). An open proof with holes $\Gamma_1 \vdash \Delta_1, \dots, \Gamma_n \vdash \Delta_n$ and conclusion $\Gamma \vdash \Delta$ is also called a **derivation** of $\Gamma \vdash \Delta$ from $\Gamma_1 \vdash \Delta_1, \dots, \Gamma_n \vdash \Delta_n$.

Definition 3.2.2 (Provability, admissibility, derivability)**.** A sequent is **provable** if there exists a proof with this sequent as a conclusion. A formula is **provable** if the singleton sequent $(\vdash A)$ of this formula is provable. An inference rule:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta}$$

is **admissible** (drawn with a dashed line) from a set of rules $\mathcal{S}$ if the provability of all its premises (using $\mathcal{S}$) implies the provability of its conclusion (using $\mathcal{S}$).

[1] In a formal definition, the root of the tree is a node with no label (or a with a special conclusion label). The associated edge is called the conclusion of the proof.

Identity and negation group

$$\frac{}{A \vdash A}\ (ax) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ (cut) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta}\ (n_L) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^{\perp}, \Delta}\ (n_R)$$

Multiplicative group

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\ (\otimes_L) \quad \frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}\ (\mathbf{1}_L) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\ (\otimes_R) \quad \frac{}{\vdash \mathbf{1}}\ (\mathbf{1}_R)$$

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'}\ (\parr_L) \quad \frac{}{\perp \vdash}\ (\perp_L) \quad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \parr B, \Delta}\ (\parr_R) \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta}\ (\perp_R)$$

Additive group

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta}\ (\oplus_L) \quad \frac{}{\Gamma, \mathbf{0} \vdash \Delta}\ (\mathbf{0}_L) \quad \frac{\Gamma \vdash A_i, \Delta}{\Gamma \vdash A_1 \oplus A_2, \Delta}\ (\oplus_{Ri})$$

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \,\&\, A_2 \vdash \Delta}\ (\&_{Li}) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \,\&\, B, \Delta}\ (\&_R) \quad \frac{}{\Gamma \vdash \top, \Delta}\ (\top_R)$$

Quantifier group

In the rules $(\exists^1_L)$ (resp. $(\exists^2_L)$) and $(\forall^1_R)$ (resp. $(\forall^2_R)$), the variable x (resp. X) must not occur free in Γ nor in Δ.

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists x.A \vdash \Delta}\ (\exists^1_L) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, \exists X.A \vdash \Delta}\ (\exists^2_L) \quad \frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x.A, \Delta}\ (\exists^1_R) \quad \frac{\Gamma \vdash A[B/X], \Delta}{\Gamma \vdash \exists X.A, \Delta}\ (\exists^2_R)$$

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x.A \vdash \Delta}\ (\forall^1_L) \quad \frac{\Gamma, A[B/X] \vdash \Delta}{\Gamma, \forall X.A \vdash \Delta}\ (\forall^2_L) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x.A, \Delta}\ (\forall^1_R) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall X.A, \Delta}\ (\forall^2_R)$$

Exponential group

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta}\ (!_L) \quad \frac{!\Gamma \vdash A, ?\Delta}{!\Gamma \vdash !A, ?\Delta}\ (!_R) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash ?A, \Delta}\ (?_R) \quad \frac{!\Gamma, A \vdash ?\Delta}{!\Gamma, ?A \vdash ?\Delta}\ (?_L)$$

Structural group

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}\ (ex_L) \quad \frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2}\ (ex_R)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta}\ (w_L) \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?A, \Delta}\ (w_R) \quad \frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta}\ (c_L) \quad \frac{\Gamma \vdash ?A, ?A, \Delta}{\Gamma \vdash ?A, \Delta}\ (c_R)$$

Table 3.2: Inference rules for two-sided Linear Logic sequent calculus

The inference rule:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta}$$

is said **derivable** (drawn with a double line) from a set of rules $\mathcal{S}$ if there exists a derivation of $\Gamma \vdash \Delta$ from the premises $\Gamma_1 \vdash \Delta_1, \dots, \Gamma_n \vdash \Delta_n$ using only the rules in $\mathcal{S}$.

Notice that the derivability of an inference rule implies its admissibility, but the converse does not hold. For example the cut rule is admissible from the other rules of Figure 3.2, as we will prove in the next sections, but it is not derivable.

Example 3.2.3. The inference rule $\dfrac{\vdash A \otimes B}{\vdash A}$ is admissible but not derivable.

Note the fundamental fact that the exchange structural rule is free on every formula, while the left (resp. right) contraction and weakening require the active formulas to be an of course (resp. why not) modality. This is a specificity of Linear Logic with respect to classical logic: if weakening and contraction were allowed for arbitrary formulas, then the multiplicatives and additives would collapse, in the sense that one group would become derivable from the other group and the free structural rules (Exercise 3.2.4).

Exercise 3.2.4. Prove that each rule in the additive (resp. multiplicative) group in Table 3.2 is derivable from the multiplicative (resp. additive) group and the structural rules free on every formulas, *i.e.*:

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ (w_L^{\text{free}}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ (w_R^{\text{free}}) \qquad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ (c_L^{\text{free}}) \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ (c_R^{\text{free}})$$

Remark 3.2.5. Since Linear Logic formulas do not have free structural rules, the multiplicative and additive rules are not interderivable from each other. On the other hand, since the exponential formulas have structural rules, one can restore de derivability *up to* some additional exponential connectives:

Provide the two sided version — Alexis

SHOW DERIVATIONS FOR EXPONENTIALS ISO — Alexis

3.3 Basic properties

rename — Alexis

First: admissible rules: free exchange + axiom for all formulas (so only atomic axiom as primitive). => DONE
Second: exercices on equivalences
Third: one-sided sequent calculus as a consequence of involution of the negation — Michele

ame section — Alexis

3.3.1 Multiset-based sequent rules

Notice that all the active occurrences in the rules of Table 3.2 are formulas next to the sequent symbol $\vdash$, except for the exchange rules. Indeed, these latter ones allow for relieving this constraint, making admissible rules firing on formulas wherever in a sequent, as the following exercise proves.

Use a uniforme notation for the permutation group — Alexis

Exercise 3.3.1. Given a natural number n, let us denote by $\mathcal{P}_n$ the set of all permutations over n. Given a list of formulas $\Gamma = A_1, \dots, A_n$ and a permutation $\sigma \in \mathcal{P}_n$, we write $\Gamma \cdot \sigma$ for the action of σ over Γ, *i.e.* the list $A_{\sigma(1)}, \dots, A_{\sigma(n)}$.

1. Prove the derivability of the following generalized exchange rule for every permutations σ and ρ:
$$\frac{\Gamma \vdash \Delta}{\Gamma \cdot \sigma \vdash \Delta \cdot \rho}$$

2. Prove that the derivability of every rule in Table 3.2 is invariant under the action of any permutation over the sequents appearing in the rule. For example, prove that for every permutations $\sigma, \sigma', \sigma'', \rho, \rho', \rho''$ of suitable domain, the rule:
$$\frac{(\Gamma, A) \cdot \sigma \vdash \Delta \cdot \rho \qquad (\Gamma', B) \cdot \sigma' \vdash \Delta' \cdot \rho'}{(\Gamma, \Gamma', A \parr B) \cdot \sigma'' \vdash (\Delta, \Delta') \cdot \rho''}$$
is derivable from the rule $(\parr_L)$ and the generalized exchange rule.

This exercise shows that one can consider sequences of formulas up to the permutations of their elements as soon as only provability matters. This means that considering sequents as made of finite multisets instead of finite sequences of formulas is a correct abstraction as long as one is only concerned with provability. More precisely, one can straightforwardly associate to any sequent $\Gamma \vdash \Delta$ the associated multiset-based sequent $(\Gamma)^{\mathsf{ms}} \vdash (\Delta)^{\mathsf{ms}}$ and define a multiset-based sequent LL calculus by viewing each sequents in the rules of Table 3.1 as a pair of multisets and considering the associated notion of proof. Let us call $\mathsf{LL}_{\mathsf{ms}}$ the associated sequent calculus.

Proposition 3.3.2. *The following hold:*

- *Every LL proof $\pi : \Gamma \vdash_{\mathsf{LL}} \Delta$ can be mapped to a $\mathsf{LL}_{\mathsf{ms}}$ proof $(\pi)^{\mathsf{ms}}$ of $(\Gamma)^{\mathsf{ms}} \vdash_{\mathsf{LL}_{\mathsf{ms}}} (\Delta)^{\mathsf{ms}}$*
- *For any LL sequent $\Gamma \vdash \Delta$, if $\Pi : (\Gamma)^{\mathsf{ms}} \vdash_{\mathsf{LL}_{\mathsf{ms}}} (\Delta)^{\mathsf{ms}}$, there is an LL proof π such that $\pi : \Gamma \vdash_{\mathsf{LL}} \Delta$ and $(\pi)^{\mathsf{ms}} = \Pi$.*

We will adopt this convention henceforth, keeping the use of the exchange rule implicit, whenever we focus *on provability*.

When interested in proofs and not only provability, especially with a focus on cut-elimination, the multiset abstraction is unsound. The previous example still shows that one can consider a proof build only with the derived rules and therefore abstract and therefore neglect the exchange rules in the proof.

3.3.2 Expansion of identities

The axiom rule in Table 3.2 is defined for any formula. However regarding the expressiveness of the system, it is enough to restrict it to atomic formulas. Indeed, Table 3.3 defines a cut-free proof $\eta(A) : A \vdash A$ for every formula A in which all occurrences of the (ax) rule have atomic formulas in conclusion. This proof requires almost no structural rule (just one exchange rule in the case of negation). We call $\eta(A)$ the **extensional expansion of** A, or the *η-expansion of* A, as it corresponds with the η-expansion rule in λ-calculus. It is also called the **identity expansion of** A: the definition of $\eta(A)$ reflects the decomposition of the identity morphism in categorical models, to be described in Section 6.1. The definition of $\eta(A)$ is also crucial in the notion of syntactic isomorphism.

refer to the section on λ-calculus when we have one; note that it is a typed η-expansion — Lionel

We should add a section on isomorphisms. There is stuff on the LLWiki that could be reused. — Lionel

3.3.3 Linear equivalences

Two formulas A and B are (linearly) **equivalent**, written $A \dashv\vdash B$, if both implications $A \multimap B$ and $B \multimap A$ are provable (*i.e.* if $A \multimap\!\!\circ\, B$ is provable). Equivalently, $A \dashv\vdash B$ if both $A \vdash B$ and $B \vdash A$ are provable. Thanks to the cut rule, this is also equivalent to asking that for all Γ and Δ: $\Gamma \vdash A, \Delta$ is provable if and only if $\Gamma \vdash B, \Delta$ is provable.

Remark 3.3.3. By definition, we have $A \dashv\vdash B$ if and only if $A^\perp \dashv\vdash B^\perp$.

reference to appropriate section? — Olivier

Two related notions are isomorphism (stronger than equivalence) and equiprovability (weaker than equivalence): $\vdash A \iff \vdash B$.

this latter can be already defined, while for isomorphism we should wait for cut-elimination (right ?). Most of the equicalences below seems to me isomorphisms... — Michele

Example 3.3.4. For any formulas A and B, $A \otimes B$ and $A \,\&\, B$ are equiprovable. However neither $\bot \otimes \bot \vdash \bot \,\&\, \bot$ nor $\bot \,\&\, \bot \vdash \bot \otimes \bot$ are provable.

Example 3.3.5 (Linear distributivity). Given three LL formulas A, B and C, it is in general not the case that $A \otimes (B \parr C) \dashv\vdash (A \otimes B) \parr C$. Indeed, while $A \otimes (B \parr C) \vdash (A \otimes B) \parr C)$ is provable, $(A \otimes B) \parr C) \vdash A \otimes (B \parr C)$ is not. (The proof of this latter fact immediately follows from cut-elimination.)

Exercise 3.3.6 (Beffara's formula). Prove that $A \otimes (A^\perp \parr A)$ is linearly equivalent to A.

3.3.3.1 De Morgan laws

A key property of Linear Logic coming from the left-right symmetry of sequents is that negation is involutive:

$$A \dashv\vdash (A^\perp)^\perp$$

$$\eta(\alpha) = \frac{}{\alpha \vdash \alpha}\ (ax) \quad \eta(\mathbf{0}) = \frac{}{\mathbf{0} \vdash \mathbf{0}}\ (\mathbf{0}_L) \quad \eta(\top) = \frac{}{\top \vdash \top}\ (\top_R)$$

$$\eta(A^\perp) = \frac{\dfrac{\eta(A) : A \vdash A}{A^\perp, A \vdash}\ (n_L)}{A^\perp \vdash A^\perp}\ (n_R) \quad \eta(\mathbf{1}) = \frac{\dfrac{}{\vdash \mathbf{1}}\ (\mathbf{1}_R)}{\mathbf{1} \vdash \mathbf{1}}\ (\mathbf{1}_L) \quad \eta(\bot) = \frac{\dfrac{}{\bot \vdash}\ (\bot_L)}{\bot \vdash \bot}\ (\bot_R)$$

$$\eta(A \otimes B) = \frac{\dfrac{\eta(A) : A \vdash A \qquad \eta(B) : B \vdash B}{A, B \vdash A \otimes B}\ (\otimes_R)}{A \otimes B \vdash A \otimes B}\ (\otimes_L)$$

$$\eta(A \parr B) = \frac{\dfrac{\eta(A) : A \vdash A \qquad \eta(B) : B \vdash B}{A \parr B \vdash A, B}\ (\parr_L)}{A \parr B \vdash A \parr B}\ (\parr_R)$$

$$\eta(A \oplus B) = \frac{\dfrac{\eta(A) : A \vdash A}{A \vdash A \oplus B}\ (\oplus_{R1}) \qquad \dfrac{\eta(B) : B \vdash B}{B \vdash A \oplus B}\ (\oplus_{R2})}{A \oplus B \vdash A \oplus B}\ (\oplus_L)$$

$$\eta(A\&B) = \frac{\dfrac{\eta(A) : A \vdash A}{A\&B \vdash A}\ (\&_{L1}) \qquad \dfrac{\eta(B) : B \vdash B}{A\&B \vdash B}\ (\&_{L2})}{A\&B \vdash A\&B}\ (\&_R)$$

$$\eta(!A) = \frac{\dfrac{\eta(A) : A \vdash A}{!A \vdash A}\ (!_L)}{!A \vdash !A}\ (!_R) \qquad \eta(?A) = \frac{\dfrac{\eta(A) : A \vdash A}{A \vdash ?A}\ (?_R)}{?A \vdash ?A}\ (?_L)$$

$$\eta(\exists \xi.A) = \frac{\dfrac{\eta(A) : A \vdash A}{A \vdash \exists \xi.A}\ (\exists_R)}{\exists \xi.A \vdash \exists \xi.A}\ (\exists_L) \qquad \eta(\forall \xi.A) = \frac{\dfrac{\eta(A) : A \vdash A}{\forall \xi.A \vdash A}\ (\forall_L)}{\forall \xi.A \vdash \forall \xi.A}\ (\forall_R)$$

Table 3.3: η-expansion

This induces a De Morgan duality between connectives:

$$\begin{array}{cc}
(A \otimes B)^\perp \dashv\vdash A^\perp \parr B^\perp & (A \parr B)^\perp \dashv\vdash A^\perp \otimes B^\perp \\
\mathbf{1}^\perp \dashv\vdash \bot & \bot^\perp \dashv\vdash \mathbf{1} \\
(A \oplus B)^\perp \dashv\vdash A^\perp \,\&\, B^\perp & (A \,\&\, B)^\perp \dashv\vdash A^\perp \oplus B^\perp \\
\mathbf{0}^\perp \dashv\vdash \top & \top^\perp \dashv\vdash \mathbf{0} \\
(!A)^\perp \dashv\vdash ?(A^\perp) & (?A)^\perp \dashv\vdash !(A^\perp) \\
(\exists \xi.A)^\perp \dashv\vdash \forall \xi.(A^\perp) & (\forall \xi.A)^\perp \dashv\vdash \exists \xi.(A^\perp)
\end{array}$$

Notice that one can define a rewrite system $\to_{\mathsf{dM}}$ on LL formulas by orienting from left to right each of the above twelve linear equivalences and adding the reduction $a^{\perp\perp} \to_{\mathsf{dM}} a$ for a an atomic formula:

$$\begin{array}{cc}
(a^\perp)^\perp \to_{\mathsf{dM}} a & (a \text{ an atom}) \\
\mathbf{1}^\perp \to_{\mathsf{dM}} \bot & \bot^\perp \to_{\mathsf{dM}} \mathbf{1} \\
(A \otimes B)^\perp \to_{\mathsf{dM}} A^\perp \parr B^\perp & (A \parr B)^\perp \to_{\mathsf{dM}} A^\perp \otimes B^\perp \\
\top^\perp \to_{\mathsf{dM}} \mathbf{0} & \mathbf{0}^\perp \to_{\mathsf{dM}} \top \\
(A \oplus B)^\perp \to_{\mathsf{dM}} A^\perp \,\&\, B^\perp & (A \,\&\, B)^\perp \to_{\mathsf{dM}} A^\perp \oplus B^\perp \\
(!A)^\perp \to_{\mathsf{dM}} ?A^\perp & (?A)^\perp \to_{\mathsf{dM}} !A^\perp \\
(\forall \xi.A)^\perp \to_{\mathsf{dM}} \exists \xi.A^\perp & (\exists \xi.A)^\perp \to_{\mathsf{dM}} \forall \xi.A^\perp \quad .
\end{array}$$

The resulting rewriting system is convergent: it has the diamond property and it is strongly normalizing (defining the weight of a negation $A^\perp$ as being the size of the formula tree of A, neglecting the negations, one remarks that each reduction step decreases the weight of the negation involved in the redex without modifying the weights of the others negation, therefore every reduction step strictly decreases the sum of the weights of all negations). The unique $\to_{\mathsf{dM}}$-normal form of a formula A, written A^{dM}, is called the *de Morgan normal form* of A.

3.3.3.2 Fundamental equivalences

We now give a list of equivalences which constitute key properties of Linear Logic provability.

Associativity, commutativity, neutrality:

$$\begin{array}{ccc}
A \otimes (B \otimes C) \dashv\vdash (A \otimes B) \otimes C & A \otimes B \dashv\vdash B \otimes A & A \otimes \mathbf{1} \dashv\vdash A \\
A \parr (B \parr C) \dashv\vdash (A \parr B) \parr C & A \parr B \dashv\vdash B \parr A & A \parr \bot \dashv\vdash A \\
A \oplus (B \oplus C) \dashv\vdash (A \oplus B) \oplus C & A \oplus B \dashv\vdash B \oplus A & A \oplus \mathbf{0} \dashv\vdash A \\
A \,\&\, (B \,\&\, C) \dashv\vdash (A \,\&\, B) \,\&\, C & A \,\&\, B \dashv\vdash B \,\&\, A & A \,\&\, \top \dashv\vdash A
\end{array}$$

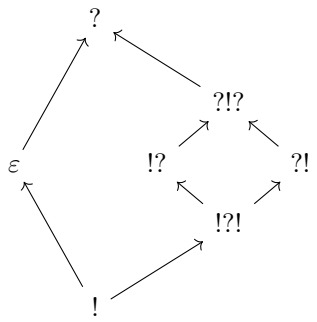


Figure 3.1: The lattice of iterated exponential modalities

Idempotence of additives:

$$A \oplus A \dashv\vdash A \qquad\qquad A \,\&\, A \dashv\vdash A$$

Distributivity of multiplicatives over additives:

$$A \otimes (B \oplus C) \dashv\vdash (A \otimes B) \oplus (A \otimes C) \qquad A \otimes \mathbf{0} \dashv\vdash \mathbf{0}$$
$$A \parr (B \,\&\, C) \dashv\vdash (A \parr B) \,\&\, (A \parr C) \qquad A \parr \top \dashv\vdash \top$$

Defining property of exponentials:

$$!(A \,\&\, B) \dashv\vdash !A \otimes !B \qquad !\top \dashv\vdash \mathbf{1}$$
$$?(A \oplus B) \dashv\vdash ?A \parr ?B \qquad ?\mathbf{0} \dashv\vdash \bot$$

Idempotence of exponentials:

$$!!A \dashv\vdash !A \qquad\qquad ??A \dashv\vdash ?A$$

Alternated idempotence of exponentials:

$$!?!?A \dashv\vdash !?A \qquad !?\mathbf{1} \dashv\vdash \mathbf{1}$$
$$?!?!A \dashv\vdash ?!A \qquad ?!\bot \dashv\vdash \bot$$

These properties of exponentials lead to the lattice of iterated exponential modalities (see Figure 3.1 where an arrow $\mu \to \nu$ means that for any formula A, we have $\mu A \vdash \nu A$, and see Exercise 3.3.7 for corresponding proofs).

Exercise 3.3.7. An *iterated exponential modality* is a (possibly empty) sequence of exponential modalities (for example the empty one ε, !!! or ?!!?). Given two iterated exponential modalities μ and ν, we say that $\mu \leq \nu$ if for any A, $\mu A \vdash \nu A$ is provable. If μ is an iterated exponential modality, its dual $\mu^\perp$ is obtained by turning each ! into a ? and each ? into a ! (*e.g.* $(!!?)^\perp = ??!$).

1. Prove that $\leq$ defines a preorder on iterated exponential modalities.

2. Prove that $\leq$ is not a total preorder.
3. Given three iterated exponential modalities μ, ν_1 and ν_2 prove that $\nu_1 \leq \nu_2$ implies $\mu\nu_1 \leq \mu\nu_2$.
4. Given two iterated exponential modalities μ and ν prove that $\mu \leq \nu$ implies $\nu^\perp \leq \mu^\perp$.
5. Using the properties about exponentials given before this exercise, prove that the equivalence relation induced by $\leq$ has at most 7 equivalence classes.
6. Prove that the order relations pictured on Figure 3.1 hold.
7. Prove that $? \not\leq \varepsilon$, $? \not\leq ?!?$, $?! \not\leq !?$ and $!? \not\leq ?!$.
8. Conclude that Figure 3.1 exactly describes the preorder relation on iterated exponential modalities.

Commutation of quantifiers (we assume that ζ does not occur in A):

$$\exists\xi.\exists\psi.A \dashv\vdash \exists\psi.\exists\xi.A \quad \exists\xi.(A \oplus B) \dashv\vdash \exists\xi.A \oplus \exists\xi.B \quad \exists\zeta.(A \otimes B) \dashv\vdash A \otimes \exists\zeta.B \quad \exists\zeta.A \dashv\vdash A$$
$$\forall\xi.\forall\psi.A \dashv\vdash \forall\psi.\forall\xi.A \quad \forall\xi.(A \,\&\, B) \dashv\vdash \forall\xi.A \,\&\, \forall\xi.B \quad \forall\zeta.(A \parr B) \dashv\vdash A \parr \forall\zeta.B \quad \forall\zeta.A \dashv\vdash A$$

3.3.3.3 Second-order definability

The units and the additive connectives can be defined using second-order quantification and exponentials, indeed the following equivalences hold:

$$\begin{aligned}
\mathbf{0} &\dashv\vdash \forall X.X \\
\top &\dashv\vdash \exists X.X \\
\mathbf{1} &\dashv\vdash \forall X.(X \multimap X) \\
\perp &\dashv\vdash \exists X.(X \otimes X^\perp) \\
A \oplus B &\dashv\vdash \forall X.(!(A \multimap X) \multimap !(B \multimap X) \multimap X) \\
A \,\&\, B &\dashv\vdash \exists X.(!(A \multimap X) \otimes !(B \multimap X) \multimap X^\perp)
\end{aligned}$$

The constants $\top$ and $\perp$ and the connective & can be defined by duality (see Remark 3.3.3).

Why not giving it fully? — Alexis

3.3.4 Deduction lemma

As already mentioned in Section 3.3.3, there are many ways of relating the provability of two formulas in Linear Logic. The goal of a deduction lemma is to relate the possibility of deriving $\vdash B$ from $\vdash A$ and the provability of $A \vdash B$, but things are a bit subtle in Linear Logic.

Lemma 3.3.8 (Weakening)**.** *If* $\Gamma \vdash \Delta$ *is derivable from* $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$*, then* $!A, \Gamma \vdash \Delta$ *is derivable from* $!A, \Gamma_1 \vdash \Delta_1, \ldots, !A, \Gamma_n \vdash \Delta_n$ *as soon as* A *is a closed formula.*

Proof. By induction on the derivation of $\Gamma \vdash \Delta$. Typical key cases are:

- (ax) rule:

$$\frac{}{B \vdash B}\,(ax) \qquad \mapsto \qquad \frac{\dfrac{}{B \vdash B}\,(ax)}{!A, B \vdash B}\,(w_L)$$

- $(\otimes_R)$ rule:

$$\frac{\Gamma \vdash B, \Delta \qquad \Gamma' \vdash C, \Delta'}{\Gamma, \Gamma' \vdash B \otimes C, \Delta, \Delta'}\,(\otimes_R) \qquad \mapsto \qquad \frac{\dfrac{!A, \Gamma \vdash B, \Delta \qquad !A, \Gamma' \vdash C, \Delta'}{!A, !A, \Gamma, \Gamma' \vdash B \otimes C, \Delta, \Delta'}\,(\otimes_R)}{!A, \Gamma, \Gamma' \vdash B \otimes C, \Delta, \Delta'}\,(c_L)$$

- $(!_R)$ rule:

$$\frac{!\Gamma \vdash B, ?\Delta}{!\Gamma \vdash !B, ?\Delta}\,(!_R) \qquad \mapsto \qquad \frac{!A, !\Gamma \vdash B, ?\Delta}{!A, !\Gamma \vdash !B, ?\Delta}\,(!_R)$$

- $(\forall_R)$ rule:

$$\frac{\Gamma \vdash B, \Delta}{\Gamma \vdash \forall \xi.B, \Delta}\,(\forall_R) \qquad \mapsto \qquad \frac{!A, \Gamma \vdash B, \Delta}{!A, \Gamma \vdash \forall \xi.B, \Delta}\,(\forall_R)$$

  This is correct because A is closed and thus ξ is not free in $!A$.

□

In the previous lemma, it is crucial that we use a prefixing ! to cross $(!_R)$ and $(?_L)$ rules. Similarly the closure assumption on A allows to cross $(\forall_R)$ and $(\exists_L)$ rules.

Exercise 3.3.9. Prove that the rule of example 3.2.3, $\dfrac{\vdash A \otimes B}{\vdash A}$, is not derivable.

Lemma 3.3.10 (Deduction)**.** *Assuming* A *is a closed formula, there is a derivation of* $\Gamma \vdash \Delta$ *from possibly many assumptions* $\vdash A$ *if and only if* $!A, \Gamma \vdash \Delta$ *is provable.*

Proof. In the first direction, we apply Lemma 3.3.8 to get a derivation of $!A, \Gamma \vdash \Delta$ from assumptions $!A \vdash A$. We then turn it into a proof of $!A, \Gamma \vdash \Delta$ by replacing these assumptions with:

$$\frac{\dfrac{}{A \vdash A}\,(ax)}{!A \vdash A}\,(!_L)$$

Conversely, we can build the derivation:

$$\dfrac{\dfrac{\vdash A}{\vdash !A}\,(!_R) \qquad !A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\,(cut)$$

□

As already mentioned above for the weakening lemma, the introduction of the ! connective in the deduction lemma is crucial since $A \vdash !A$ is not provable in general while we have:

$$\dfrac{\vdash A}{\vdash !A}\,(!_R)$$

3.3.5 One-sided sequent calculus

is dual connective defined? mention this in Section 3.1. — Alexis

Notice that Table 3.2 is symmetric, similarly to the sequent calculus LK for classical logic: for every left introduction rule, there is a right introduction rule for the dual connective that has the exact same structure. Moreover, because of the involutivity of negation proved in Section 3.3.3, the hypothesis and the thesis in a sequent can be exchanged by negation.

Exercise 3.3.11. A sequent $\Gamma \vdash \Delta$ is provable iff $\vdash \Gamma^\perp, \Delta$ is provable iff $\Delta^\perp \vdash \Gamma^\perp$ is provable iff $\Gamma, \Delta^\perp \vdash$ is provable.

Similarly to what happens in LK, these remarks allow to define a one-sided sequent calculus, proving the same formulas as the calculus in Table 3.2, while enjoying the following features:

- formulas are built inductively from literals (atoms and negated atoms) using all LL connectives except linear negation:

$$\begin{array}{rcccccccc} F, G & ::= & a & \mid F \otimes G & \mid \mathbf{1} & \mid F \oplus G & \mid \mathbf{0} & \mid\, !F \\ & & \mid a^\perp & \mid F \parr G & \mid \perp & \mid F \,\&\, G & \mid \top & \mid\, ?F; \end{array}$$

- negation is not a connective anymore, but a syntactically defined operation on formulas given by the De Morgan laws: the negation of A is defined to be the formula $(A^\perp)^{\mathsf{dM}}$ (see Section 3.3.3), that we shall simply write $(A)^\perp$, or simply $A^\perp$, in the context one-sided sequent calculus. The full definition of the negation operator is:

$$\begin{array}{rlrl} (\alpha)^\perp &:= \alpha^\perp & (\alpha^\perp)^\perp &:= \alpha \\ (\mathbf{1})^\perp &:= \perp & (\perp)^\perp &:= \mathbf{1} \\ (A \otimes B)^\perp &:= (A)^\perp \parr (B)^\perp & (A \parr B)^\perp &:= (A)^\perp \otimes (B)^\perp \\ (\top)^\perp &:= \mathbf{0} & (\mathbf{0})^\perp &:= \top \\ (A \oplus B)^\perp &:= (A)^\perp \,\&\, (B)^\perp & (A \,\&\, B)^\perp &:= (A)^\perp \oplus (B)^\perp \\ (!A)^\perp &:= ?(A)^\perp & (?A)^\perp &:= !(A)^\perp \\ (\forall \xi.A)^\perp &:= \exists \xi.(A)^\perp & (\exists \xi.A)^\perp &:= \forall \xi.(A)^\perp \quad . \end{array}$$

Identity group

$$\frac{}{\vdash F^\perp, F}\ (ax) \qquad \frac{\vdash F, \Gamma \quad \vdash F^\perp, \Delta}{\vdash \Gamma, \Delta}\ (cut)$$

Multiplicative group

$$\frac{\vdash F, \Gamma \quad \vdash G, \Delta}{\vdash F \otimes G, \Gamma, \Delta}\ (\otimes) \qquad \frac{}{\vdash \mathbf{1}}\ (\mathbf{1}) \qquad \frac{\vdash F, G, \Gamma}{\vdash F \parr G, \Gamma}\ (\parr) \qquad \frac{\vdash \Gamma}{\vdash \perp, \Gamma}\ (\perp)$$

Additive group

$$\frac{\vdash F_i, \Gamma}{\vdash F_1 \oplus F_2, \Gamma}\ (\oplus_i) \qquad \frac{\vdash F, \Gamma \quad \vdash G, \Gamma}{\vdash F \,\&\, G, \Gamma}\ (\&) \qquad \frac{}{\vdash \top, \Gamma}\ (\top)$$

Quantifier group

In the rule $(\forall^1)$ (resp. $(\forall^2)$), the variable x (resp. X) must not occur free in Γ.

$$\frac{\vdash F[t/x], \Gamma}{\vdash \exists x.F, \Gamma}\ (\exists^1) \qquad \frac{\vdash F[B/X], \Gamma}{\vdash \exists X.F, \Gamma}\ (\exists^2) \qquad \frac{\vdash F, \Gamma}{\vdash \forall x.F, \Gamma}\ (\forall^1) \qquad \frac{\vdash F, \Gamma}{\vdash \forall X.F, \Gamma}\ (\forall^2)$$

Exponential group

$$\frac{\vdash F, ?\Gamma}{\vdash !F, ?\Gamma}\ (!) \qquad \frac{\vdash F, \Gamma}{\vdash ?F, \Gamma}\ (?)$$

Structural group

$$\frac{\vdash \Gamma, F, G, \Delta}{\vdash \Gamma, G, F, \Delta}\ (ex) \qquad \frac{\vdash \Gamma}{\vdash ?F, \Gamma}\ (w) \qquad \frac{\vdash ?F, ?F, \Gamma}{\vdash ?F, \Gamma}\ (c)$$

Table 3.4: Inference rules for the one-sided Linear Logic sequent calculus

- sequents now have the form $\vdash \Gamma$;
- the rules are essentially the same as those of the two-sided version, except that the left hand side of sequents is kept empty and that the axiom and cut rules shall be adapted.

The rules of the one-sided sequent calculus are presented in Table 3.4, where every formula occurring in each sequent is assumed to be DM-normal.[2] Note that there is no rule for negation in Table 3.4, as this is now an involutive operator on formulas rather than a connective.

Theorem 3.3.12. *A two-sided sequent $\Gamma \vdash \Delta$ is provable (resp. provable without (cut)) by rules of Table 3.2 if and only if the sequent $\vdash \Gamma^{\perp\mathsf{dM}}, \Delta^{\mathsf{dM}}$ is provable*

[2] We assume the same conventions as above regarding the derivable rules using the exchange rule.

(resp. provable without (cut)) by the rules of Table 3.4.

Proof (Sketch). The *if*-direction is a consequence of Exercise 3.3.11 and the fact that the rules of Table 3.4 are specific instances of the right rules of Table 3.2, but for the axiom and the cut rules, which can be easily proved admissible from Table 3.2. The *only-if*-direction can be proven by structural induction on a proof of $\Gamma \vdash \Delta$. □

The one-sided calculus is often used when studying proofs because it is much lighter (less than half the number of rules) than the two-sided form while keeping the same expressiveness. In the next sections, we will establish the key properties of this sequent calculus — including the admissibility of the (*cut*) rule, the subformula property, *etc.* — for the one-sided version: their generalization to the two-sided version is straightforward. Moreover, proof nets, to be introduced in Chapter 4, can be seen as a quotient of one-sided sequent calculus proofs under some commutations of rules.

add a reference to the commutation theorem here — Michele

Beyond that point, unless we explicitly consider a two-sided calculus, we will generally identify any formula A with its DM-normal form A^{dM}.

3.4 Some fragments of interest

In general, a **fragment** of a logical system S is a logical system obtained by restricting the language of S, and by restricting the rules of S accordingly.

The most well known fragments are obtained by combining/removing in different ways the classes of formula constructors present in the language of Linear Logic formulas (see Table 3.1):

- atoms;
- multiplicative connectives and their units;
- additive connectives and their units;
- exponential modalities;
- quantifiers.

The fragments of LL obtained in this way are denoted by prefixing LL with letters corresponding to the classes of connectives being considered: M for multiplicative connectives, A for additive connectives, and E for exponential connectives. Additional subscripts specify what atoms and/or quantifiers are included: 0 when we include units and propositional variables; 1 when we include general predicates and first-order quantification; 2 when we include second-order quantification on propositional variables; and these can be combined, so that the subscript 02 indicates that we consider units, propositional variables, and quantification on the latter. We moreover consider two further restrictions of the propositional case, denoted by specific subscripts: u when units are the only

atoms; v when propositional variables are the only atoms; and in both cases we exclude any form of quantification.

For instance, in the multiplicative case, we obtain the following fragments of the language of formulas:

- MLL_u with constructors: $\mathbf{1}$, $\bot$, $\otimes$, $\parr$;
- MLL_v with constructors: X, $X^\bot$, $\otimes$, $\parr$;
- MLL_0 with constructors: $\mathbf{1}$, $\bot$, X, $X^\bot$, $\otimes$, $\parr$;
- MLL_1 with constructors: $p(t_1,\dots,t_n)$, $p(t_1,\dots,t_n)^\bot$, $\otimes$, $\parr$, $\forall x$, $\exists x$ — where p ranges over predicate symbols;
- MLL_{01} with constructors: $\mathbf{1}$, $\bot$, $p(t_1,\dots,t_n)$, $p(t_1,\dots,t_n)^\bot$, $\otimes$, $\parr$, $\forall x$, $\exists x$ — where p ranges over predicate symbols;
- MLL_2 with constructors: X, $X^\bot$, $\otimes$, $\parr$, $\forall X$, $\exists X$;
- MLL_{02} with constructors: $\mathbf{1}$, $\bot$, X, $X^\bot$, $\otimes$, $\parr$, $\forall X$, $\exists X$;
- MLL_{12} with constructors: $p(t_1,\dots,t_n)$, $p(t_1,\dots,t_n)^\bot$, $\otimes$, $\parr$, $\forall x$, $\exists x$, $\forall X$, $\exists X$ — where p ranges over predicate symbols and second order variables;
- MLL_{012} with constructors: $\mathbf{1}$, $\bot$, $p(t_1,\dots,t_n)$, $p(t_1,\dots,t_n)^\bot$, $\otimes$, $\parr$, $\forall x$, $\exists x$, $\forall X$, $\exists X$ — where p ranges over predicate symbols and second order variables.

Having fixed the classes of connectives, atoms and quantifiers to be considered, the induced fragment consists of the rules of Table 3.4 (or Table 3.2 in the two-sided version, in which case one must also allow linear negation as a connective), minus those that mention missing constructors.

Fragments of interest include:

- MLL_v: formulas are built from propositional variables (and their duals) using only $\otimes$ and $\parr$ connectives; and the only rules are (ax), (cut), $(\otimes)$, $(\parr)$ and (ex). This forms the minimal core of Linear Logic, and generally serves as a playground where everything works perfectly. For instance, we will first present proof nets in that setting: see Section 4.2.

Also mention MLL_u, with complexity results? — Lionel

- LL_0: formulas are built from propositional variables (and their duals) using multiplicative and additive connectives and units, as well as exponential modalities; and the rules are those of Table 3.4 minus the quantifier group. This is the fragment for which we will provide a full proof of the admissibility of (cut), in Section 3.5.
- MELL_v: formulas are built from propositional variables (and their duals) using multiplicative connectives as well as exponential modalities; and the rules are those of Table 3.4 minus the additive and quantifier groups. This is the fragment for which proof nets are more generally introduced and studied; it is moreover the target of translations of the λ-calculus.

- LL_{02}: all constructors are allowed except for predicates of non-zero arity and first-order quantification. This is often considered as the full version of Linear Logic, as first-order terms and quantification are generally left aside.
- MELL_2: formulas are built from propositional variables (and their duals) using multiplicative connectives, as well as exponential modalities and second-order quantifiers; and the rules are those of Table 3.4 minus the additive group and the rules for first-order quantifiers and multiplicative and additive units. This is in fact as expressive as LL_{02}: as we have seen in Section 3.3.3.3, the additive connectives and the multiplicative and additive units can be encoded using second-order quantification.

Other fragments are built by keeping connectives from all classes while contraining the way they can be combined.

- Intuitionistic formulas are *output formulas* (noted o) and *input formulas* (noted ι):

$$\begin{array}{rcl} o & ::= & \alpha \mid o \otimes o \mid \iota \parr o \mid \mathbf{1} \mid o \oplus o \mid o \,\&\, o \mid \mathbf{0} \mid \top \mid !o \mid \forall \xi.o \mid \exists \xi.o \\ \iota & ::= & \alpha^\perp \mid \iota \parr \iota \mid o \otimes \iota \mid \bot \mid \iota \,\&\, \iota \mid \iota \oplus \iota \mid \top \mid \mathbf{0} \mid ?\iota \mid \exists \xi.\iota \mid \forall \xi.\iota \end{array}$$

  Note that $\bot \otimes ??\bot$ is not an intuitionistic formula, while $\top$ is both input and output. If o (resp. ι) is an output (resp. input) formula then $o^\perp$ (resp. $\iota^\perp$) is an input (resp. output) formula. See Section 3.7.2.1 for more details about this fragment and its link with Intuitionistic Linear Logic (ILL).

- Polarized formulas are *positive formulas* (noted P) and *negative formulas* (noted N):

$$\begin{array}{rcl} P & ::= & \alpha \mid P \otimes P \mid \mathbf{1} \mid P \oplus P \mid \mathbf{0} \mid !N \mid \exists \xi.P \\ N & ::= & \alpha^\perp \mid N \parr N \mid \bot \mid N \,\&\, N \mid \top \mid ?P \mid \forall \xi.N \end{array}$$

  Note that positive and negative formulas are disjoints classes of formulas and that $?\bot \otimes \top$ is not a polarized formula. If P (resp. N) is a positive (resp. negative) formula then $P^\perp$ (resp. $N^\perp$) is a negative (resp. positive) formula.

3.5 Cut elimination and consequences

The admissibility of the cut rule is a corner property of Linear Logic (as for many other sequent calculi). It leads in particular to the sub-formula property and then to consistency.

Theorem 3.5.1 (Cut admissibility)**.** *For every sequent $\Gamma \vdash \Delta$, there is a proof of $\Gamma \vdash \Delta$ if and only if there is a proof of $\Gamma \vdash \Delta$ that does not use the cut rule.*

In order to prove this admissibility property, we are going to provide an explicit *cut elimination* procedure which progressively reduces cuts in a proof (which may contain many) until the proof becomes cut-free.

3.5.1 A proof for propositional Linear Logic

This section presents a proof of the cut elimination property for the sequent calculus of propositional Linear Logic, that is Linear Logic without the second-order nor first-order quantifiers. The method used here consists in defining an appropriate reduction relation over proofs and proving its weak normalization, to cut-free proofs, by a simple induction over proofs with an appropriate termination measure. While the technique can be easily extended to first-order (and its extension does not bear any specificities due to Linear Logic itself), it does not extend to second-order logic: although the induction steps are the same, the termination argument requires more powerful tools.

give reference — Olivier

In order to motivate the main ingredient of the proof, we shall first consider few examples of proofs with cuts and how to simplify them:

$$\dfrac{\dfrac{\vdash A, \Gamma_1 \qquad \vdash B, \Gamma_2}{\vdash A \otimes B, \Gamma_1, \Gamma_2}\,(\otimes) \qquad \dfrac{\vdash A^\perp, B^\perp, \Delta}{\vdash A^\perp \parr B^\perp, \Delta}\,(\parr)}{\vdash \Gamma_1, \Gamma_2, \Delta}\,(cut)$$

$$\xrightarrow{\otimes/\parr} \qquad \dfrac{\vdash A, \Gamma_1 \qquad \dfrac{\vdash B, \Gamma_2 \qquad \vdash A^\perp, B^\perp, \Delta}{\vdash A^\perp, \Gamma_2, \Delta}\,(cut)}{\vdash \Gamma_1, \Gamma_2, \Delta}\,(cut)$$

here one cut generates two cuts but they act on strictly smaller formulas.

$$\dfrac{\dfrac{\vdash A, B, C, \Gamma}{\vdash A \parr B, C, \Gamma}\,(\parr) \qquad \vdash C^\perp, \Delta}{\vdash A \parr B, \Delta, \Gamma}\,(cut)$$

$$\xrightarrow{\mathsf{comm}(\parr)} \qquad \dfrac{\vdash A, B, C, \Gamma \qquad \vdash C^\perp, \Delta}{\dfrac{\vdash A, B, \Delta, \Gamma}{\vdash A \parr B, \Delta, \Gamma}\,(\parr)}\,(cut)$$

here the cut still acts on the same formula but its left premise comes from a strictly smaller proof.

$$\dfrac{\dfrac{\vdash ?A, ?A, \Gamma}{\vdash ?A, \Gamma}\,(c) \qquad \vdash !A^\perp, ?\Delta}{\vdash ?\Delta, \Gamma}\,(cut)$$

$$\xrightarrow{\mathsf{c}/!} \qquad \dfrac{\dfrac{\vdash ?A, ?A, \Gamma \qquad \vdash !A^\perp, ?\Delta}{\vdash ?A, ?\Delta, \Gamma}\,(cut) \qquad \vdash !A^\perp, ?\Delta}{\dfrac{\vdash ?\Delta, ?\Delta, \Gamma}{\vdash ?\Delta, \Gamma}\,(c)}\,(cut)$$

here one cut generates two cuts acting on the same formula, the top-most one acts on smaller proofs but there is no such guarantee for the bottom one (this is the main source of difficulty in the proof to come).

Definition 3.5.2 (Cut Rank)**.** Let $\pi : \vdash \Gamma$ be a proof and r an occurrence of a cut inference of π.

The *cut rank* of r, $\mathsf{rk}(r)$ is the complexity of its cut-formula, that is the number of connectives of the cut formula. The *proof rank* of π, $\mathsf{rk}(\pi)$ is the supremum of its cut ranks.

Definition 3.5.3 (Level of a cut)**.** Let $\pi : \vdash \Gamma$ be a proof and r an occurrence of a cut inference of π. The *level* of r, $\mathsf{lvl}(r)$, is the size of the proof tree rooted in r.

As rewriting steps on proof to eliminate cuts, we consider the transformations described in Fig. 3.2 to Fig. 3.8. They have the property that if no step can be applied in a proof, then this proof is cut-free. Using the notions of cut rank and level of a cut, it is possible to prove the weak normalization of this rewriting system. We will follow this approach with proof-nets for the multiplicative exponential fragment. We adopt here a slightly different approach in the sequent calculus by generalizing the notion of cut to avoid the problem we have seen with the reduction of cuts on contractions. This is similar to the approach of Gentzen for the sequent calculus of classical logic.

ref — Olivier

ref — Olivier

The following definition introduces a generalized cut which is a derivable rule in LL thanks to the cut rule and the structural rules of weakening and contraction. In a sequent, $\vdash \Gamma, A^{(n)}, \Delta$ means $\vdash \Gamma, A, \ldots, A, \Delta$ with n occurrences of A.

Definition 3.5.4 (Structural cut)**.** The following inference is called *structural cut*: $\dfrac{\vdash C^{(k)}, \Gamma \quad \vdash C^{\perp\,(l)}, \Delta}{\vdash \Gamma, \Delta}\ (scut)$ where an index among k, l differs from 1 only if the formula it labels is a ?-formula.

As a consequence of the definition of structural cut, it is not possible that both k and l differ from 1. Moreover if $\vdash C^{(k)}, \Gamma$ (and the same for $\vdash C^{\perp\,(l)}, \Delta$) is the premise of a structural cut then: $\dfrac{\vdash C^{(k)}, \Gamma}{\vdash C, \Gamma}$ is derivable (if $k = 1$ it is immediate, otherwise C starts with a ? and we can use a (w) rule for $k = 0$ and $k - 1$ (c) rules for $k > 1$).

In the following, we consider LL sequent calculus extended with the structural cut inference, the proof of which being called structural proofs. We will prove LL cut-elimination by defining a weakly-normalizing reduction, $\longmapsto_{\mathsf{c}}$, on those structural derivation trees, such that normal forms are (structural) cut-free proofs.

3.5.1.1 Rank-decreasing reductions

Before actually defining the cut reduction, let us first consider a sufficient condition for (structural) cut-elimination.

In the following, all relations we shall consider will be assumed to have the property that if two proofs are in relation, they have the same conclusion.

$$\dfrac{\pi_1 : \vdash C, \Gamma \qquad \dfrac{}{\vdash C^\perp, C}\,(ax)}{\vdash C, \Gamma}\,(cut) \xrightarrow{\mathsf{ax}} \pi_1 : \vdash C, \Gamma$$

Figure 3.2: Axiom case.

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma_1, B \qquad \pi_2 : \vdash \Gamma_2, C}{\vdash \Gamma_1, \Gamma_2, B \otimes C}\,(\otimes) \qquad \dfrac{\pi_3 : \vdash \Delta, B^\perp, C^\perp}{\vdash \Delta, B^\perp \parr C^\perp}\,(\parr)}{\vdash \Gamma_1, \Gamma_2, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\otimes/\parr} \qquad \dfrac{\pi_1 : \vdash \Gamma_1, B \qquad \dfrac{\pi_2 : \vdash \Gamma_2, C \qquad \pi_3 : \vdash \Delta, B^\perp, C^\perp}{\vdash \Gamma_2, \Delta, B^\perp}\,(cut^\gamma)}{\vdash \Gamma_1, \Gamma_2, \Delta}\,(cut^\beta)$$

$$\dfrac{\dfrac{}{\vdash \mathbf{1}}\,(\mathbf{1}) \qquad \dfrac{\pi_1 : \vdash \Gamma}{\vdash \Gamma, \bot}\,(\bot)}{\vdash \Gamma}\,(cut) \xrightarrow{\mathbf{1}/\bot} \pi_1 : \vdash \Gamma$$

Figure 3.3: Multiplicative key cases.

$$\dfrac{\dfrac{\pi_0 : \vdash \Gamma, C_i}{\vdash \Gamma, C_1 \oplus C_2}\,(\oplus_i) \qquad \dfrac{\pi_1 : \vdash \Delta, C_1^\perp \qquad \pi_2 : \vdash \Delta, C_2^\perp}{\vdash, \Delta, C_1^\perp \,\&\, C_2^\perp}\,(\&)}{\vdash \Gamma, \Delta}\,(cut^\alpha) \xrightarrow{\oplus/\&} \dfrac{\pi_0 : \vdash \Gamma, C_i \qquad \pi_i : \vdash \Delta, C_i^\perp}{\vdash \Gamma, \Delta}\,(cut^\beta) \qquad i \in \{1, 2\}$$

Figure 3.4: Additive key case.

$$\dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \dfrac{\pi_2 : \vdash \Delta, B^\perp}{\vdash \Delta, ?B^\perp}\,(?)}{\vdash ?\Gamma, \Delta}\,(cut^\alpha) \xrightarrow{!/?} \dfrac{\pi_1 : \vdash ?\Gamma, B \qquad \pi_2 : \vdash \Delta, B^\perp}{\vdash ?\Gamma, \Delta}\,(cut^\beta)$$

$$\dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \dfrac{\pi_2 : \vdash \Delta}{\vdash \Delta, ?B^\perp}\,(w)}{\vdash ?\Gamma, \Delta}\,(cut) \xrightarrow{!/?} \dfrac{\pi_2 : \vdash \Delta}{\vdash ?\Gamma, \Delta}\,(w)$$

$$\dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \dfrac{\pi_2 : \vdash \Delta, ?B^\perp, ?B^\perp}{\vdash \Delta, ?B^\perp}\,(c)}{\vdash ?\Gamma, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{!/?} \dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \pi_2 : \vdash \Delta, ?B^\perp, ?B^\perp}{\vdash ?\Gamma, \Delta, ?B^\perp}\,(cut^\gamma)}{\vdash ?\Gamma, ?\Gamma, \Delta}\,(cut^\beta)}{\vdash ?\Gamma, \Delta}\,(c)$$

Figure 3.5: Exponential key cases.

Definition 3.5.5 (Contextual reduction)**.** A binary relation R on proof trees is *contextual* if for every proofs π_0, π_1, π_0' such that $\pi_0 \, R \, \pi_1$, the proof π_1' obtained by replacing a subtree of π_0' equal to π_0 with π_1 is such that $\pi_0' \, R \, \pi_1'$.

Definition 3.5.6 (Rank-decreasing reduction)**.** Let $\rightsquigarrow$ be a contextual reduction on structural proofs. $\rightsquigarrow$ is said to be *rank-decreasing* if for any proof π of the form $\dfrac{\pi_1 : \vdash C^{(k)}, \Gamma \qquad \pi_2 : \vdash C^{\perp\,(l)}, \Delta}{\vdash \Gamma, \Delta}\,(scut)$ such that $\mathsf{rk}(\pi_1), \mathsf{rk}(\pi_2) < \mathsf{rk}(\pi)$, there exists π' such that $\pi \rightsquigarrow^\star \pi'$ and $\mathsf{rk}(\pi') < \mathsf{rk}(\pi)$.

Rank-decreasing reductions satisfy the following:

Theorem 3.5.7. *If $\rightsquigarrow$ is rank-decreasing, then for any sequent Γ and for any proof $\pi : \vdash \Gamma$, there exists a cut-free proof π' of $\vdash \Gamma$ such that $\pi \rightsquigarrow^\star \pi'$.*

Proof. We prove the theorem by induction on the following measure (ordered lexicographically):

$$\mathsf{w}(\pi) = \begin{cases} (0,0) & \text{if } \pi \text{ is scut-free} \\ (r,n) & \text{otherwise, with } r = \mathsf{rk}(\pi) \text{ and } n \\ & \text{the number of cuts of rank } r \text{ in } \pi. \end{cases}$$

Indeed, if $\mathsf{w}(\pi) = (0,0)$, π is cut-free by definition. Otherwise, let $(r,n) = \mathsf{w}(\pi)$. π contains n structural cuts of rank r. Consider an uppermost occurrence

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, A, B, C}{\vdash \Gamma, A \parr B, C}\,(\parr) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash \Gamma, A \parr B, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\parr)} \qquad \dfrac{\pi_1 : \vdash \Gamma, A, B, C \qquad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \parr B, \Delta}\,(\parr)}\,(cut^\beta)$$

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, A, C \qquad \pi_2 : \vdash \Gamma', B}{\vdash \Gamma, \Gamma', A \otimes B, C}\,(\otimes) \qquad \pi_3 : \vdash \Delta, C^\perp}{\vdash \Gamma, \Gamma', A \otimes B, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\otimes\mathsf{l})} \qquad \dfrac{\dfrac{\pi_1 : \vdash \Gamma, A, C \qquad \pi_3 : \vdash \Delta, C^\perp}{\vdash \Gamma, A, \Delta}\,(cut^\beta) \qquad \pi_2 : \vdash \Gamma', B}{\vdash \Gamma, \Gamma', A \otimes B, \Delta}\,(\otimes)$$

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, A \qquad \pi_2 : \vdash \Gamma', B, C}{\vdash \Gamma, \Gamma', A \otimes B, C}\,(\otimes) \qquad \pi_3 : \vdash \Delta, C^\perp}{\vdash \Gamma, \Gamma', A \otimes B, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\otimes\mathsf{r})} \qquad \dfrac{\pi_1 : \vdash \Gamma, A \qquad \dfrac{\pi_2 : \vdash \Gamma', B, C \qquad \pi_3 : \vdash \Delta, C^\perp}{\vdash \Gamma', B, \Delta}\,(cut^\beta)}{\vdash \Gamma, \Gamma', A \otimes B, \Delta}\,(\otimes)$$

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, C}{\vdash \Gamma, \bot, C}\,(\bot) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash \Gamma, \bot, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\bot)} \qquad \dfrac{\pi_1 : \vdash \Gamma, C \qquad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash \Gamma, \Delta}{\vdash \Gamma, \bot, \Delta}\,(\bot)}\,(cut^\beta)$$

Figure 3.6: Multiplicative commutation cases.

$$\dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C \quad \pi_2 : \vdash B, \Gamma, C}{\vdash A \,\&\, B, \Gamma, C}\,(\&) \qquad \pi_3 : \vdash \Delta, C^\perp}{\vdash A \,\&\, B, \Gamma, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\&)} \quad \dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C \quad \pi_3 : \vdash \Delta, C^\perp}{\vdash A, \Gamma, \Delta}\,(cut^\beta) \qquad \dfrac{\pi_2 : \vdash B, \Gamma, C \quad \pi_3 : \vdash \Delta, C^\perp}{\vdash B, \Gamma, \Delta}\,(cut^\gamma)}{\vdash A \,\&\, B, \Gamma, \Delta}\,(\&)$$

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, A_i, C}{\vdash \Gamma, A_1 \oplus A_2, C}\,(\oplus_i) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash \Gamma, A_1 \oplus A_2, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\oplus_i)} \quad \dfrac{\pi_1 : \vdash \Gamma, A_i, C \quad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash \Gamma, A_i, \Delta}{\vdash \Gamma, A_1 \oplus A_2, \Delta}\,(\oplus_i)}\,(cut^\beta)$$

$$\dfrac{\dfrac{}{\vdash \top, \Gamma, C}\,(\top) \qquad \pi : \vdash \Delta, C^\perp}{\vdash \top, \Gamma, \Delta}\,(cut) \qquad \xrightarrow{\mathsf{comm}(\top)} \qquad \dfrac{}{\vdash \top, \Gamma, \Delta}\,(\top)$$

Figure 3.7: Additive commutation cases.

of such a cut of maximal rank and call π' the subproof rooted in this cut. By the rank-decreasing property, there exists a proof π'' such that $\pi' \rightsquigarrow^\star \pi''$ and $\mathsf{rk}(\pi'') < \mathsf{rk}(\pi')$. Indeed, the premises of π' have strictly smaller ranks by maximallity of the cut inference concluding π'. Therefore, by contextual closure of $\rightsquigarrow$, there is $\overline{\pi}$ such that $\pi \rightsquigarrow \overline{\pi}$.

Either $n > 1$ and therefore $\mathsf{w}(\overline{\pi}) = (r, n-1)$ or $n = 1$ and $\mathsf{w}(\overline{\pi}) = (\overline{r}, \overline{n})$ with $\overline{r} < r$. In both cases $\mathsf{w}(\overline{\pi}) <_{\mathsf{lex}} \mathsf{w}(\pi)$ and we can apply the induction hypothesis to $\overline{\pi}$: there exists a scut-free proof $\pi^\star$ such that $\overline{\pi} \rightsquigarrow^\star \pi^\star$ and we can conclude:

$$\pi \rightsquigarrow^\star \overline{\pi} \rightsquigarrow^\star \pi^\star.$$

□

3.5.1.2 Definition of $\longmapsto_{\mathsf{c}}$

It is therefore sufficient to exhibit a rank-decreasing reduction to deduce cut-elimination: we shall now construct such a reduction.

Two approaches: small-step vs big-step cut-reduction: add a remark on this and comments with natural deduction / λ-calculus on the one hand and explicit substitution on the other. — Alexis

In order to obtain $\longmapsto_{\mathsf{c}}$, we shall consider some cases of scut inference and collect them to the reduction relation, analyzing as we proceed their impact on the rank and level of the scuts involved in this transformations. Once this is

$$\dfrac{\dfrac{\pi_1 : \vdash A, ?\Gamma, ?C}{\vdash\, !A, ?\Gamma, ?C}\,(!) \qquad \dfrac{\pi_2 : \vdash ?\Delta, C^\perp}{\vdash ?\Delta, !C^\perp}\,(!)}{\vdash\, !A, ?\Gamma, ?\Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(!)} \qquad \dfrac{\dfrac{\pi_1 : \vdash A, ?\Gamma, ?C \qquad \dfrac{\pi_2 : \vdash ?\Delta, C^\perp}{\vdash ?\Delta, !C^\perp}\,(!)}{\vdash A, ?\Gamma, ?\Delta}\,(cut^\beta)}{\vdash\, !A, ?\Gamma, ?\Delta}\,(!)$$

$$\dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C}{\vdash ?A, \Gamma, C}\,(?) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash ?A, \Gamma, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(?)} \qquad \dfrac{\pi_1 : \vdash A, \Gamma, C \qquad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash A, \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(?)}\,(cut^\beta)$$

$$\dfrac{\dfrac{\pi_1 : \vdash ?A, ?A, \Gamma, C}{\vdash ?A, \Gamma, C}\,(c) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash ?A, \Gamma, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\mathsf{c})} \qquad \dfrac{\pi_1 : \vdash ?A, ?A, \Gamma, C \qquad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash ?A, ?A, \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(c)}\,(cut^\beta)$$

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, C}{\vdash ?A, \Gamma, C}\,(w) \qquad \pi_2 : \vdash \Delta, C^\perp}{\vdash ?A, \Gamma, \Delta}\,(cut^\alpha)$$

$$\xrightarrow{\mathsf{comm}(\mathsf{w})} \qquad \dfrac{\pi_1 : \vdash \Gamma, C \qquad \pi_2 : \vdash \Delta, C^\perp}{\dfrac{\vdash \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(w)}\,(cut^\beta)$$

Figure 3.8: Exponential commutation cases.

- $k = 1$

$$\dfrac{\pi_1 : \vdash C, \Gamma \qquad \dfrac{}{\vdash C^{\perp}, C}\,(ax)}{\vdash \Gamma, C}\,(scut) \xrightarrow{\mathsf{ax}} \pi_1 : \vdash C, \Gamma$$

- $k = 0$

$$\dfrac{\pi_1 : \vdash \Gamma \qquad \dfrac{}{\vdash !B^{\perp}, ?B}\,(ax)}{\vdash \Gamma, ?B}\,(scut) \xrightarrow{\mathsf{ax}} \dfrac{\pi_1 : \vdash \Gamma}{\vdash ?B, \Gamma}\,(w)$$

- $k > 1$

$$\dfrac{\pi_1 : \vdash ?B^{(k)}, \Gamma \qquad \dfrac{}{\vdash !B^{\perp}, ?B}\,(ax)}{\vdash \Gamma, ?B}\,(scut) \xrightarrow{\mathsf{ax}} \dfrac{\pi_1 : \vdash ?B^{(k)}, \Gamma}{\vdash ?B, \Gamma}\,(c^{k-1})$$

Figure 3.9: Axiom key cases.

done, $\longmapsto_{\mathsf{c}}$ will be defined as the contextual/compatible closure of the previous notion of reduction.

The cases we analyze will be essentially of four types: (i) one of the premises is an axiom inference and one of its conclusions is cut, this is called an *axiom key case*, (ii) the cut formula is principal in a logical rule in both premises of the cut, this is called a *key logical case*, (iii) the cut formula is an exponential formula with a ?-occurrence immediately introduced by a structural rule, this is called a *structural case*, (iv) or there is at least one premise in which the cut formula is not active, this is called a *commutative case*.

Axiom key cases When a proof has the following shape:

$$\dfrac{\pi_1 : \vdash C^{(k)}, \Gamma \qquad \dfrac{}{\vdash C^{\perp}, C}\,(ax)}{\vdash \Gamma, C}\,(scut)$$

When one of the premises of the cut is an axiom, say π_2 (the other case is treated symmetrically and will also be added in $\longrightarrow$), we distinguish two main cases:

- if $k = 1$, then we reduce π and π_1 have the same conclusion and one reduces π to π_1.
- if $k \neq 1$, then necessarily, C is a ?-formula, $?B$, for which structural rules of weakening and contraction are available. Using the structural rules, one reduces π to π_1 extended with a weakening if $k = 0$ and with the adequate number of contractions if $k > 1$.

The corresponding relation, $\xrightarrow{\mathsf{ax}}$ is defined in Figure 3.9.

Logical key cases:

Multiplicative key case: In the case of mutiplicative cut formulas, $\otimes$ vs. $\parr$ inferences, proof π has the following shape (in particular we have $k = l = 1$ in the structural cut):

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma_1, B \quad \pi_2 : \vdash \Gamma_2, C}{\vdash \Gamma_1, \Gamma_2, B \otimes C}\,(\otimes) \quad \dfrac{\pi_3 : \vdash \Delta, B^\perp, C^\perp}{\vdash \Delta, B^\perp \parr C^\perp}\,(\parr)}{\vdash \Gamma_1, \Gamma_2, \Delta}\,(scut)$$

We add the reduction depicted in Figure 3.3, denoted as $\xrightarrow{\otimes/\parr}$, where each of the bottommost cut occurrences has been labelled and we notice that $\mathsf{rk}(\beta), \mathsf{rk}(\gamma) < \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\gamma) < \mathsf{lvl}(\alpha) = \mathsf{lvl}(\beta) + 1$.

Similarly, the nullary case of the multiplicative constants is:

$$\dfrac{\dfrac{}{\vdash \mathbf{1}}\,(\mathbf{1}) \quad \dfrac{\pi_1 : \vdash \Gamma}{\vdash \Gamma, \bot}\,(\bot)}{\vdash \Gamma}\,(scut) \xrightarrow{\mathbf{1}/\bot} \pi_1 : \vdash \Gamma$$

Additive key case: In the case of additive cut formulas, $\oplus$ vs. & inferences, proof π has the following shape (in particular we have $k = l = 1$ in the structural cut):

$$\dfrac{\dfrac{\pi_0 : \vdash \Gamma, C_i}{\vdash \Gamma, C_1 \oplus C_2}\,(\oplus_i) \quad \dfrac{\pi_1 : \vdash \Delta, C_1^\perp \quad \pi_2 : \vdash \Delta, C_2^\perp}{\vdash, \Delta, C_1^\perp \,\&\, C_2^\perp}\,(\&)}{\vdash \Gamma, \Delta}\,(scut^\alpha)$$

We consider the reduction depicted in Figure 3.4, denoted as $\xrightarrow{\oplus/\&}$, where each of the bottommost cut occurrences has been labelled and we notice that $\mathsf{rk}(\beta) < \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) < \mathsf{lvl}(\alpha)$. The symmetrical $\xrightarrow{\&/\oplus}$ is naturally considered too.

Remark that there is no key case for the additive constant as 0 has no introduction rule.

Exponential key case: In the case of exponential cut formulas, ? vs. ! inferences, proof π has the following shape (with $l \geq 0$):

$$\dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \quad \dfrac{\pi_2 : \vdash \Delta, B^\perp, ?B^{\perp\,(l)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(?)}{\vdash ?\Gamma, \Delta}\,(scut)$$

We add the reduction depicted in Figure 3.10, denoted $\xrightarrow{!/?}$, where the bottommost cut occurrence has been labelled and we notice $\mathsf{rk}(\beta) < \mathsf{rk}(\alpha)$, $\mathsf{rk}(\gamma) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\gamma) = \mathsf{lvl}(\alpha) - 1$ (but $\mathsf{lvl}(\beta)$ may be larger than the $\mathsf{lvl}(\alpha)$).

$$\dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \dfrac{\pi_2 : \vdash \Delta, B^{\perp}, ?B^{\perp\,(l)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(?)}{\vdash ?\Gamma, \Delta}\,(scut^{\alpha}) \xrightarrow{!/?} \dfrac{\pi_1 : \vdash ?\Gamma, B \qquad \dfrac{\dfrac{\pi_1 : \vdash ?\Gamma, B}{\vdash ?\Gamma, !B}\,(!) \qquad \pi_2 : \vdash \Delta, B^{\perp}, ?B^{\perp}}{\vdash ?\Gamma, \Delta, B^{\perp}}\,(scut^{\beta})}{\vdash ?\Gamma, ?\Gamma, \Delta}\,(c^{\star})}{\vdash ?\Gamma, \Delta}$$

Figure 3.10: Exponential key case.

- Weakening:

$$\dfrac{\pi_1 : \vdash \Gamma, !B \qquad \dfrac{\pi_2 : \vdash \Delta, ?B^{\perp\,(l)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(w)}{\vdash \Gamma, \Delta}\,(scut^{\alpha}) \xrightarrow{\mathrm{w}} \dfrac{\pi_1 : \vdash \Gamma, !B \qquad \pi_2 : \vdash \Delta, ?B^{\perp\,(l)}}{\vdash \Gamma, \Delta}\,(scut^{\beta})$$

- Contraction:

$$\dfrac{\pi_1 : \vdash \Gamma, !B \qquad \dfrac{\pi_2 : \vdash \Delta, ?B^{\perp\,(l+2)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(c)}{\vdash ?\Gamma, \Delta}\,(scut^{\alpha}) \xrightarrow{\mathrm{c}} \dfrac{\pi_1 : \vdash \Gamma, !B \qquad \pi_2 : \vdash \Delta, ?B^{\perp\,(l+2)}}{\vdash \Gamma, \Delta}\,(scut^{\beta})$$

Figure 3.11: Structural cases.

Structural cases These are cases which are specific to the use of structural cuts. Such situations are handled differently with usual cuts.

- Weakening:

$$\dfrac{\pi_1 : \vdash \Gamma, !B \qquad \dfrac{\pi_2 : \vdash \Delta, ?B^{\perp\,(l)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(w)}{\vdash \Gamma, \Delta}\,(scut)$$

- Contraction:

$$\dfrac{\pi_1 : \vdash \Gamma, !B \qquad \dfrac{\pi_2 : \vdash \Delta, ?B^{\perp\,(l+2)}}{\vdash \Delta, ?B^{\perp\,(l+1)}}\,(c)}{\vdash \Gamma, \Delta}\,(scut)$$

We add the reductions depicted in Figure 3.11, denoted $\xrightarrow{\mathrm{w}}$ and $\xrightarrow{\mathrm{c}}$ where each of the bottommost cut occurrences has been labelled and we notice the following relations about the ranks and levels of cuts:

- (w): $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\beta) = \mathsf{lvl}(\alpha) - 1$.
- (c): $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\beta) = \mathsf{lvl}(\alpha) - 1$.

Commutative cases When none of the previous cases applies, one considers commutation steps which do not modify the rank of the cut but decrease the level of the cut. We depict some of thoses cases:

ax **commutation step**

$$\dfrac{\dfrac{}{\vdash A^{\perp}, A, C^{(k)}}\,(ax) \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\vdash A^{\perp}, A, \Delta}\,(scut)$$

We notice that necessarily $k = 0$ and thus C is a ?-formula $?B$. This makes this situation non symmetric, and allows us to assume (after applying commutative steps) that $!B^{\perp}$ is principal:

$$\dfrac{\dfrac{}{\vdash A^{\perp}, A}\,(ax) \qquad \pi_2 : \vdash ?\Delta, !B^{\perp}}{\vdash A^{\perp}, A, ?\Delta}\,(scut)$$

$$\xrightarrow{\mathsf{comm}(ax)} \qquad \dfrac{\dfrac{}{\vdash A^{\perp}, A}\,(ax)}{\vdash A^{\perp}, A, ?\Delta}\,(w)$$

The result is cut-free.

$\parr$ commutation step

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, A, B, C^{(k)}}{\vdash \Gamma, A \parr B, C^{(k)}}\,(\parr) \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\vdash \Gamma, A \parr B, \Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(\parr)} \qquad \dfrac{\pi_1 : \vdash \Gamma, A, B, C^{(k)} \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\dfrac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, A \parr B, \Delta}\,(\parr)}\,(scut^{\beta})$$

We notice that $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) < \mathsf{lvl}(\alpha)$.

$\&$ commutation step

$$\dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C^{(k)} \qquad \pi_2 : \vdash B, \Gamma, C^{(k)}}{\vdash A \,\&\, B, \Gamma, C^{(k)}}\,(\&) \qquad \pi_3 : \vdash \Delta, C^{\perp\,(l)}}{\vdash A \,\&\, B, \Gamma, \Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(\&)} \qquad \dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C^{(k)} \qquad \pi_3 : \vdash \Delta, C^{\perp\,(l)}}{\vdash A, \Gamma, \Delta}\,(scut^{\beta}) \qquad \dfrac{\pi_2 : \vdash B, \Gamma, C^{(k)} \qquad \pi_3 : \vdash \Delta, C^{\perp\,(}}{\vdash B, \Gamma, \Delta}\,(\&)}{\vdash A \,\&\, B, \Gamma, \Delta}$$

We notice that $\mathsf{rk}(\beta) = \mathsf{rk}(\gamma) = \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta), \mathsf{lvl}(\gamma) < \mathsf{lvl}(\alpha)$.

$\top$ commutation step

$$\dfrac{\dfrac{}{\vdash \top, \Gamma, C^{(k)}}\,(\top) \qquad \pi : \vdash \Delta, C^{\perp\,(l)}}{\vdash \top, \Gamma, \Delta}\,(scut^{\alpha}) \qquad \xrightarrow{\mathsf{comm}(\top)} \qquad \dfrac{}{\vdash \top, \Gamma, \Delta}\,(\top)$$

We notice that the resulting proof is cut-free.

Promotion commutation step

$$\dfrac{\dfrac{\pi_1 : \vdash A, ?\Gamma, ?C^{(k)}}{\vdash\, !A, ?\Gamma, ?C^{(k)}}\,(!) \qquad \pi_2 : \vdash ?\Delta, !C^{\perp}}{\vdash\, !A, ?\Gamma, ?\Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(!)} \qquad \dfrac{\pi_1 : \vdash A, ?\Gamma, ?C^{(k)} \qquad \pi_2 : \vdash ?\Delta, !C^{\perp}}{\dfrac{\vdash A, ?\Gamma, ?\Delta}{\vdash\, !A, ?\Gamma, ?\Delta}\,(!)}\,(scut^{\beta})$$

We notice that $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) < \mathsf{lvl}(\alpha)$. Note also that in the case where the context of $!C^{\perp}$ is not made of ?-formulas, we can apply a commutation step on π_2 since $!C^{\perp}$ is not principal.

Dereliction commutation step

$$\dfrac{\dfrac{\pi_1 : \vdash A, \Gamma, C^{(k)}}{\vdash ?A, \Gamma, C^{(k)}}\,(?) \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\vdash ?A, \Gamma, \Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(?)} \qquad \dfrac{\pi_1 : \vdash A, \Gamma, C^{(k)} \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\dfrac{\vdash A, \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(?)}\,(scut^{\beta})$$

We notice that $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) < \mathsf{lvl}(\alpha)$.

Structural commutation step For contraction:

$$\dfrac{\dfrac{\pi_1 : \vdash ?A, ?A, \Gamma, C^{(k)}}{\vdash ?A, \Gamma, C^{(k)}}\,(c) \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\vdash ?A, \Gamma, \Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(\mathsf{c})} \qquad \dfrac{\pi_1 : \vdash ?A, ?A, \Gamma, C^{(k)} \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\dfrac{\vdash ?A, ?A, \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(c)}\,(scut^{\beta})$$

For weakening:

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, C^{(k)}}{\vdash ?A, \Gamma, C^{(k)}}\,(w) \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\vdash ?A, \Gamma, \Delta}\,(scut^{\alpha})$$

$$\xrightarrow{\mathsf{comm}(\mathsf{w})} \qquad \dfrac{\pi_1 : \vdash \Gamma, C^{(k)} \qquad \pi_2 : \vdash \Delta, C^{\perp\,(l)}}{\dfrac{\vdash \Gamma, \Delta}{\vdash ?A, \Gamma, \Delta}\,(w)}\,(scut^{\beta})$$

For both cases, we notice that $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) < \mathsf{lvl}(\alpha)$.

3.5.1.3 $\longmapsto_{\mathsf{c}}$ is rank-decreasing

We now prove that $\longrightarrow$ is rank-decreasing, therefore concluding that cut-elimination holds.

Lemma 3.5.8. *Let π be an LL structural proof of the form:*

$$\dfrac{\pi_1 : \vdash C^{(k)}, \Gamma \qquad \pi_2 : \vdash C^{\perp\,(l)}, \Delta}{\vdash \Gamma, \Delta}\,(scut)$$

such that $\mathsf{rk}(\pi_1), \mathsf{rk}(\pi_2) < \mathsf{rk}(\pi)$, then there exists π' such that $\pi \longrightarrow^{\star} \pi'$ and $\mathsf{rk}(\pi') < \mathsf{rk}(\pi)$.

Proof. We prove the lemma by induction on the size of π.

Base case: Assume that at both premises of π are or size 1: they are either an axiom, a $\mathbf{1}$ or a $\top$. Then π reduces to some scut-free π', therefore of rank 0.

Inductive case: We reason by case analysis on the last inference of π_1 and π_2.

ADDREF — Alexis

1. Linear key-cases: we saw in the previous section that the rank decrease by one step of $\longrightarrow$.
2. Axiom/scut case: assume, wlog, that π_2 is an axiom: context Δ is the singleton context C and $\pi \vdash \Gamma, C$. Then either $k = 1$ and we set π' to π_1 or $l \neq 1$ and C is an ?-formula and there is some π' obtained by adding structural rules on C to the conclusion of π_1, such that $\pi \longrightarrow \pi'$. In both cases $\mathsf{rk}(pi') = \mathsf{rk}(\pi_1) < \mathsf{rk}(\pi)$.

ADDREF — Alexis

3. Exponential key-case: we saw in the previous section that $\pi \longrightarrow \pi'$ by reducing the root cut of π into two scuts: one of the same rank and lower level and one of lower rank. We can apply the induction hypothesis to the subproof of π' rooted in the cut of maximal rank (equal to $\mathsf{rk}(\pi)$) as it is of a smaller size than π and we obtain a proof π'', such that $\pi \longrightarrow \pi' \longrightarrow^\star \pi''$ and $\mathsf{rk}(\pi'') < \mathsf{rk}(\pi)$.

ADDREF — Alexis

4. Commutative cases: In all cases, $\pi \longrightarrow \pi_0$ where the root cut of π is transformed into one or more cuts on $C, C^\perp$, *ie* of the same rank as π, which are all incomparable (*ie* they are on different branches) and of a strictly smaller level.

   As a consequence, the induction hypothesis can be applied independently to each of the proofs $\pi_{01}, \dots, \pi_{0k}$ rooted in those cuts of rank equating $\mathsf{rk}(\pi)$ leading proofs $\pi'_{01}, \dots, \pi'_{0k}$ such that $\pi_{0i} \longrightarrow^\star \pi'_{0i}, 1 \leq i \leq k$ of rank strictly smaller that $\mathsf{rk}(\pi)$ and we conclude by contextuality of $\longrightarrow$ that $\pi_0 \longrightarrow^\star \pi'$ with $\mathsf{rk}(\pi') < \mathsf{rk}(\pi)$.
5. Structural cases: we have $\pi \longrightarrow \pi'$ by reducing the root cut of π into at most one cut of the same rank and lower level. We can apply the induction hypothesis to the subproof of π' rooted in the cut (if any) and we obtain a proof π'', such that $\pi \longrightarrow \pi' \longrightarrow^\star \pi''$ and $\mathsf{rk}(\pi'') < \mathsf{rk}(\pi)$.

□

3.5.2 Consequences

Cut elimination has several important consequences:

Definition 3.5.9 (Subformula)**.** The subformulas of a formula A are A and, inductively, the subformulas of its immediate subformulas:

- the immediate subformulas of $A \otimes B$, $A \parr B$, $A \oplus B$, $A \,\&\, B$ are A and B,
- the only immediate subformula of $!A$ and $?A$ is A,

- $\mathbf{1}$, $\bot$, $\mathbf{0}$, $\top$ and atomic formulas have no immediate subformula,
- the immediate subformulas of $\exists x.A$ and $\forall x.A$ are all the $A[t/x]$ for all first-order terms t,
- the immediate subformulas of $\exists X.A$ and $\forall X.A$ are all the $A[B/X]$ for all formulas B (with the appropriate number of parameters).

why not removing $A[t/x]$ from immediate subformulas of $\forall x.A$? — Olivier

Theorem 3.5.10 (Subformula property)**.** *A sequent $\Gamma \vdash \Delta$ is provable if and only if it is the conclusion of a proof in which each intermediate conclusion is made of subformulas of the formulas of Γ and Δ.*

Proof. By the cut elimination theorem, if a sequent is provable, then it is provable by a cut-free proof. In each rule except the cut rule, all formulas of the premises are either formulas of the conclusion, or immediate subformulas of it, therefore cut-free proofs have the subformula property. □

The subformula property means essentially nothing in the second-order system, since any formula is a subformula of a quantified formula where the quantified variable occurs. However, the property is very meaningful if the sequent Γ does not use second-order quantification, as it puts a strong restriction on the set of potential proofs of a given sequent.

In particular, it implies that the first-order fragment without quantifiers is decidable.

This has to be a proposition. — Alexis

Theorem 3.5.11 (Consistency)**.** *The empty sequent $\vdash$ is not provable. Subsequently, it is impossible to prove both a formula A and its negation $A^\bot$; it is impossible to prove $\mathbf{0}$ or $\bot$.*

Proof. If a sequent is provable, then it is the conclusion of a cut-free proof. In each rule except the cut rule, there is at least one formula in conclusion. Therefore $\vdash$ cannot be the conclusion of a proof. The other properties are immediate consequences: if $\vdash A^\bot$ and $\vdash A$ are provable, then by the cut rule one gets empty conclusion, which is not possible. As particular cases, since $\mathbf{1}$ and $\top$ are provable, $\bot$ and $\mathbf{0}$ are not, since they are equivalent to $\mathbf{1}^\bot$ and $\top^\bot$ respectively. □

3.6 Reversibility and focusing

As already seen in Section 3.5.2, cut-free proofs play a central role when studying provability. In particular when trying to determine whether a sequent $\vdash \Gamma$ is provable or not, it is equivalent to wonder whether it has a cut-free proof or not. This is a very important property for *proof search* since it induces a huge restriction on the set of proofs one has to explore when trying to find a proof of a given sequent. One can wonder whether it is possible to restrict even more the set of proofs without loosing provability. That is to find constraints on proofs such that the sequents provable with and without these constraints are the same.

3.6.1 Reversibility

Definition 3.6.1 (Reversibility)**.** A connective c is called *reversible* if for every proof $\pi : \vdash \Gamma, c(A_1, \dots, A_n)$, there is a proof π' with the same conclusion in which $c(A_1, \dots, A_n)$ is introduced by the last rule (*i.e.* principal).

Remark 3.6.2. Now that we have Theorem 3.5.1, we can refine a bit Definition 3.2.2 with the following implications:

$$\begin{array}{ccccc} \text{derivable without cuts} & \Rightarrow & \text{derivable with cuts} & \Rightarrow & \text{admissible} \\ \dfrac{\vdash A}{\vdash A \parr \bot} & & \dfrac{\vdash A \,\&\, B}{\vdash A} & & \dfrac{\vdash ?A}{\vdash A, ?A} \end{array}$$

Example rules satisfy the given property but not the stronger ones, thus proving that the reverse implications do not hold in general.

Lemma 3.6.3. *The following* reversed rules *are derivable with cuts:*

$$\frac{\vdash A \parr B, \Gamma}{\vdash A, B, \Gamma}\,(\parr^{rev}) \qquad \frac{\vdash \bot, \Gamma}{\vdash \Gamma}\,(\bot^{rev}) \qquad \frac{\vdash A_1 \,\&\, A_2, \Gamma}{\vdash A_i, \Gamma}\,(\&_i^{rev}) \quad i \in \{1,2\}$$

$$\frac{\vdash\, !A, \Gamma}{\vdash A, \Gamma}\,(!^{rev}) \qquad \frac{\vdash \forall \xi.A, \Gamma}{\vdash A, \Gamma}\,(\forall^{rev})$$

Proof. Derivability results from the following proof schema (which relies on considering proofs from Table 3.3 and removing their last rule):

- $$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax) \quad \dfrac{}{\vdash B, B^\perp}\,(ax)}{\vdash A, B, A^\perp \otimes B^\perp}\,(\otimes) \qquad \vdash A \parr B, \Gamma}{\vdash A, B, \Gamma}\,(cut)$$
- $$\dfrac{\dfrac{}{\vdash 1}\,(\mathbf{1}) \qquad \vdash \bot, \Gamma}{\vdash \Gamma}\,(cut)$$
- $$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax)}{\vdash A, A^\perp \oplus B^\perp}\,(\oplus_1) \qquad \vdash A \,\&\, B, \Gamma}{\vdash A, \Gamma}\,(cut)$$
- $$\dfrac{\dfrac{\dfrac{}{\vdash B, B^\perp}\,(ax)}{\vdash B, A^\perp \oplus B^\perp}\,(\oplus_2) \qquad \vdash A \,\&\, B, \Gamma}{\vdash B, \Gamma}\,(cut)$$
- $$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax)}{\vdash A, ?A^\perp}\,(?) \qquad \vdash\, !A, \Gamma}{\vdash A, \Gamma}\,(cut)$$

- $$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax)}{\vdash A, \exists \xi.A^\perp}\,(\exists) \qquad \vdash \forall \xi.A, \Gamma}{\vdash A, \Gamma}\,(cut)$$

□

Theorem 3.6.4. *The negative connectives $\parr$, $\bot$, $\&$ and $\forall$ are reversible.*

Proof. For each connective $c \in \{\parr, \bot, \&, \forall\}$, we can apply the reversed rule from Lemma 3.6.3 followed by the introduction rule of c to the proof of $\vdash \Gamma, c(A_1, \dots, A_n)$:

$$\dfrac{\dfrac{\vdash A \parr B, \Gamma}{\vdash A, B, \Gamma}\,(\parr^{\mathrm{rev}})}{\vdash A \parr B, \Gamma}\,(\parr) \qquad \dfrac{\dfrac{\vdash \bot, \Gamma}{\vdash \Gamma}\,(\bot^{\mathrm{rev}})}{\vdash \bot, \Gamma}\,(\bot) \qquad \dfrac{\dfrac{\vdash A \,\&\, B, \Gamma}{\vdash A, \Gamma}\,(\&_1^{\mathrm{rev}}) \qquad \dfrac{\vdash A \,\&\, B, \Gamma}{\vdash B, \Gamma}\,(\&_2^{\mathrm{rev}})}{\vdash A \,\&\, B, \Gamma}\,(\&)$$

$$\dfrac{\dfrac{\vdash \forall \xi.A, \Gamma}{\vdash A, \Gamma}\,(\forall^{\mathrm{rev}})}{\vdash \forall \xi.A, \Gamma}\,(\forall)$$

In the $(\forall)$ case, this requires to choose a ξ which is not free in Γ (this is always possible up to renaming). But a similar dependency over the context with !, makes ! non reversible since it is not possible to apply (!) to $\vdash A, \Gamma$ in general since Γ may not start with a ?. □

Should make a section on proof search? — Alexis

A consequence of this fact is that, when searching for a proof of some sequent $\vdash \Gamma$, one can always start by decomposing negative connectives in Γ without losing provability. Applying this result to successive connectives, we can get generalized formulations for more complex formulas. For instance:

$\vdash \Gamma, (A \parr B) \parr (B \,\&\, C)$ is provable

iff $\vdash \Gamma, A \parr B, B \,\&\, C$ is provable

iff $\vdash \Gamma, A \parr B, B$ and $\vdash \Gamma, A \parr B, C$ are provable

iff $\vdash \Gamma, A, B, B$ and $\vdash \Gamma, A, B, C$ are provable

So without loss of provability, we can assume that any proof of $\vdash \Gamma, (A \parr B) \parr (B \,\&\, C)$ ends like:

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma, A, B, B}{\vdash \Gamma, A \parr B, B}\,(\parr) \qquad \dfrac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A \parr B, C}\,(\parr)}{\vdash \Gamma, A \parr B, B \,\&\, C}\,(\&)}{\vdash \Gamma, (A \parr B) \parr (B \,\&\, C)}\,(\parr)$$

In order to define a general statement for compound formulas, as well as an analogous result for positive connectives, we need to give a proper status to clusters of connectives of the same polarity.

3.6.2 Generalized connectives and rules

Definition 3.6.5. A *positive generalized connective* is a parametrized formula $P[X_1, \dots, X_n]$ made from the variables X_i using the connectives $\otimes$, $\oplus$, $\mathbf{1}$, $\mathbf{0}$.

A *negative generalized connective* is a parametrized formula $N[X_1, \dots, X_n]$ made from the variables X_i using the connectives $⅋$, $\&$, $\bot$, $\top$.

If $C[X_1, \dots, X_n]$ is a generalized connectives (of any polarity), the *dual* of C is the connective C^* such that $C^*[X_1^\bot, \dots, X_n^\bot] = C[X_1, \dots, X_n]^\bot$.

It is clear that dualization of generalized connectives is involutive and exchanges polarities. We do not include quantifiers in this definition, mainly for simplicity. Extending the notion to quantifiers would only require taking proper care of the scope of variables.

Sequent calculus provides introduction rules for each connective. Negative connectives have one rule, positive ones may have any number of rules, namely 2 for $\oplus$ and 0 for $\mathbf{0}$. We can derive introduction rules for the generalized connectives by combining the different possible introduction rules for each of their components.

Considering the previous example $N[X_1, X_2, X_3] = (X_1 ⅋ X_2) ⅋ (X_2 \,\&\, X_3)$, we can derive an introduction rule for N as

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma, X_1, X_2, X_2}{\vdash \Gamma, X_1 ⅋ X_2, X_2}\,(⅋) \quad \dfrac{\vdash \Gamma, X_1, X_2, X_3}{\vdash \Gamma, X_1 ⅋ X_2, X_3}\,(⅋)}{\vdash \Gamma, X_1 ⅋ X_2, X_2 \,\&\, X_3}\,(\&)}{\vdash \Gamma, (X_1 ⅋ X_2) ⅋ (X_2 \,\&\, X_3)}\,(⅋) \quad \text{or} \quad \dfrac{\dfrac{\dfrac{\vdash \Gamma, X_1, X_2, X_2 \quad \vdash \Gamma, X_1, X_2, X_3}{\vdash \Gamma, X_1, X_2, X_2 \,\&\, X_3}\,(\&)}{\vdash \Gamma, X_1 ⅋ X_2, X_2 \,\&\, X_3}\,(⅋)}{\vdash \Gamma, (X_1 ⅋ X_2) ⅋ (X_2 \,\&\, X_3)}\,(⅋)$$

but these rules only differ by the commutation of independent rules. In particular, their premises are the same. The dual of N is $P[X_1, X_2, X_3] = (X_1 \otimes X_2) \otimes (X_2 \oplus X_3)$, for which we have two possible derivations:

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma_1, X_1 \quad \vdash \Gamma_2, X_2}{\vdash \Gamma_1, \Gamma_2, X_1 \otimes X_2}\,(\otimes) \quad \dfrac{\vdash \Gamma_3, X_2}{\vdash \Gamma_3, X_2 \oplus X_3}\,(\oplus_1)}{\vdash \Gamma_1, \Gamma_2, \Gamma_3, (X_1 \otimes X_2) \otimes (X_2 \oplus X_3)}\,(\otimes) \quad \dfrac{\dfrac{\dfrac{\vdash \Gamma_1, X_1 \quad \vdash \Gamma_2, X_2}{\vdash \Gamma_1, \Gamma_2, X_1 \otimes X_2}\,(\otimes) \quad \dfrac{\vdash \Gamma_3, X_3}{\vdash \Gamma_3, X_2 \oplus X_3}\,(\oplus_2)}{\vdash \Gamma_1, \Gamma_2, \Gamma_3, (X_1 \otimes X_2) \otimes (X_2 \oplus X_3)}\,(\otimes)$$

These are actually different, in particular their premises differ. Each possible derivation corresponds to the choice of one side of the $\oplus$ connective.

We can remark that the branches of the negative inference precisely correspond to the possible positive inferences:

- the first branch of the negative inference has a premise X_1, X_2, X_2 and the first positive inference has three premises, holding X_1, X_2 and X_2 respectively.
- the second branch of the negative inference has a premise X_1, X_2, X_3 and the second positive inference has three premises, holding X_1, X_2 and X_3 respectively.

This phenomenon extends to all generalized connectives.

Definition 3.6.6. The *branching* of a generalized connective $P[X_1, \ldots, X_n]$ is the multiset $\mathcal{I}_P$ of multisets over $\{1, \ldots, n\}$ defined inductively as

$$\begin{aligned}
\mathcal{I}_{P \otimes Q} &:= [I + J \mid I \in \mathcal{I}_P, J \in \mathcal{I}_Q], \\
\mathcal{I}_{P \oplus Q} &:= \mathcal{I}_P + \mathcal{I}_Q, \\
\mathcal{I}_{\mathbf{1}} &:= [[]], \\
\mathcal{I}_{\mathbf{0}} &:= [], \\
\mathcal{I}_{X_i} &:= [[i]].
\end{aligned}$$

The branching of a negative generalized connective is the branching of its dual. Elements of a branching are called branches.

In the example above, the branching will be $[[1, 2, 2], [1, 2, 3]]$, which corresponds to the branches of the negative inference and to the cases of positive inference.

Definition 3.6.7. Let $\mathcal{I}$ be a branching. Write $\mathcal{I}$ as $[I_1, \ldots, I_k]$ and write each I_j as $[i_{j,1}, \ldots, i_{j,\ell_j}]$. The derived rule for a negative generalized connective N with branching $\mathcal{I}$ is

$$\frac{\vdash \Gamma, A_{i_{1,1}}, \ldots, A_{i_{1,\ell_1}} \quad \cdots \quad \vdash \Gamma, A_{i_{k,1}}, \ldots, A_{i_{k,\ell_k}}}{\vdash \Gamma, N[A_1, \ldots, A_n]}\ (N)$$

For each branch $I = [i_1, \ldots, i_\ell]$ of a positive generalized connective P, the derived rule for branch I of P is

$$\frac{\vdash \Gamma_1, A_{i_1} \quad \cdots \quad \vdash \Gamma_\ell, A_{i_\ell}}{\vdash \Gamma_1, \ldots, \Gamma_\ell, P[A_1, \ldots, A_n]}\ (P_I)$$

The reversibility property of negative connectives can be rephrased in a generalized way as follows:

Theorem 3.6.8. *Let N be a negative generalized connective. A sequent $\vdash \Gamma, N[A_1, \ldots, A_n]$ is provable if and only if, for each $[i_1, \ldots, i_k] \in \mathcal{I}_N$, the sequent $\vdash \Gamma, A_{i_1}, \ldots, A_{i_k}$ is provable.*

The corresponding property for positive connectives is the focusing property, defined in the next section.

3.6.3 Focusing

Definition 3.6.9. A formula is *positive* if it has a principal connective among $\otimes$, $\oplus$, $\mathbf{1}$, $\mathbf{0}$. It is called *negative* if it has a principal connective among $\parr$, $\&$, $\bot$, $\top$. It is called *neutral* if it is neither positive nor negative.

If we extended the theory to include quantifiers in generalized connectives, then the definition of positive and negative formulas would be extended to include them too.

Definition 3.6.10. A proof $\pi : \vdash \Gamma, A$ is said to be *positively focused on* A if it has the shape

$$\frac{\pi_1 : \vdash \Gamma_1, A_{i_1} \quad \cdots \quad \pi_\ell : \vdash \Gamma_\ell, A_{i_\ell}}{\vdash \Gamma_1, \ldots, \Gamma_\ell, P[A_1, \ldots, A_n]}\ (P_{[i_1,\ldots,i_\ell]})$$

where P is a positive generalized connective, the A_i are non-positive and $A = P[A_1, \ldots, A_n]$. The formula A is called the *focus* of the proof π.

Commentaire de PLC après son cours ECI 2023: This section should be improved by insisting on hereditary focused proof (not only at the bottom of the proof like in [the previous definition]). — Lionel

maybe we need to make more explicit what we mean by cluster. — Alexis

In other words, a proof is positively focused on a conclusion A if its last rules build A from some of its non-positive subformulas in one cluster of inferences. Note that this notion only makes sense for a sequent made only of positive formulas, since by this definition a proof is obviously positively focused on any of its non-positive conclusions, using the degenerate generalized connective $P[X] = X$.

Theorem 3.6.11. *A sequent* $\vdash \Gamma$ *is cut-free provable if and only if it is provable by a cut-free proof that is positively focused.*

Proof. We reason by induction on a (cut-free) proof π of Γ. As noted above, the result trivially holds if Γ has a non-positive formula. We can thus assume that Γ contains only positive formulas and reason on the nature of the last rule, which is necessarily the introduction of a positive connective (it cannot be an axiom rule because an axiom always has at least one non-positive conclusion).

Suppose that the last rule of π introduces a tensor, so that π is

$$\frac{\rho : \vdash \Gamma, A \qquad \theta : \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ (\otimes)$$

By induction hypothesis, there are positively focused proofs $\rho' : \vdash \Gamma, A$ and $\theta' \vdash \Delta, B$. If A is the focus of ρ' and B is the focus of θ', then the proof

$$\frac{\rho' : \vdash \Gamma, A \qquad \theta' : \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ (\otimes)$$

is positively focused on $A \otimes B$, so we can conclude. Otherwise, one of the two proofs is positively focused on another conclusion. Without loss of generality, suppose that ρ' is not positively focused on A. Then it decomposes as

$$\frac{\rho_1 : \vdash \Gamma_1, C_{i_1} \quad \cdots \quad \rho_\ell : \vdash \Gamma_\ell, C_{i_\ell}}{\vdash \Gamma_1, \ldots, \Gamma_\ell, P[C_1, \ldots, C_n]}$$

where the C_i are not positive and A belongs to some context Γ_j that we will write Γ'_j, A. Then we can conclude with the proof

$$\frac{\rho_1 : \vdash \Gamma_1, C_{i_1} \quad \cdots \quad \dfrac{\rho_j : \vdash \Gamma_j, A, C_{i_j} \qquad \theta : \vdash \Delta, B}{\vdash \Gamma_j, \Delta, A \otimes B, C_{i_j}}\ (\otimes) \quad \cdots \quad \rho_\ell : \vdash \Gamma_\ell, C_{i_\ell}}{\vdash \Gamma_1, \ldots, \Gamma_\ell, \Delta, A \otimes B, P[C_1, \ldots, C_n]}$$

which is positively focused on $P[C_1, \ldots, C_n]$.

If the last rule of π introduces a $\oplus$, we proceed the same way except that there is only one premise. If the last rule of π introduces a $\mathbf{1}$, then it is the only rule of π, which is thus positively focused on this $\mathbf{1}$. □

As in the reversibility theorem, this proof only makes use of commutation of independent rules.

These results say nothing about exponential modalities, because they respect neither reversibility nor focusing. However, if we consider the fragment of LL which consists only of multiplicative and additive connectives, we can restrict the proof rules to enforce focusing without loss of expressiveness.

3.7 Variations

In order to discuss some varations on the sequent calculus rules, we go back to two-sided sequents in this section.

3.7.1 Exponential rules

The promotion rule, on the right-hand side for example:

$$\frac{!A_1, \ldots, !A_n \vdash B, ?B_1, \ldots, ?B_m}{!A_1, \ldots, !A_n \vdash !B, ?B_1, \ldots, ?B_m}\ (!_R)$$

can be replaced by a *multi-functorial* promotion rule:

$$\frac{A_1, \ldots, A_n \vdash B, B_1, \ldots, B_m}{!A_1, \ldots, !A_n \vdash !B, ?B_1, \ldots, ?B_m}\ (!^{\mathrm{mf}}_R)$$

and *digging* rules:

$$\frac{\Gamma \vdash ??A, \Delta}{\Gamma \vdash ?A, \Delta}\ (digR) \qquad \frac{\Gamma, !!A \vdash \Delta}{\Gamma, !A \vdash ?A, \Delta}\ (digL)$$

without modifying the provability:

$$\cfrac{\cfrac{\cfrac{!A_1, \ldots, !A_n \vdash B, ?B_1, \ldots, ?B_m}{!!A_1, \ldots, !!A_n \vdash !B, ??B_1, \ldots, ??B_m}\ (!^{\mathrm{mf}}_R)}{!!A_1, \ldots, !!A_n \vdash !B, ?B_1, \ldots, ?B_m}\ (digR)}{!A_1, \ldots, !A_n \vdash !B, ?B_1, \ldots, ?B_m}\ (digL)$$

$$\cfrac{\cfrac{\cfrac{A_1, \ldots, A_n \vdash B, B_1, \ldots, B_m}{!A_1, \ldots, !A_n \vdash B, B_1, \ldots, B_m}\ (!_L)}{!A_1, \ldots, !A_n \vdash B, ?B_1, \ldots, ?B_m}\ (?_R)}{!A_1, \ldots, !A_n \vdash !B, ?B_1, \ldots, ?B_m}\ (!_R) \qquad \cfrac{\Gamma \vdash ??A, \Delta \quad \cfrac{\cfrac{}{?A \vdash ?A}\ (ax)}{??A \vdash ?A}\ (?_L)}{\Gamma \vdash ?A, \Delta}\ (cut)$$

Note that digging violates the subformula property.

In presence of the digging rule, the multiplexing rule $\dfrac{\Gamma \vdash A^{(n)}, \Delta}{\Gamma \vdash ?A, \Delta}\ (mplex)$ (remember that $A^{(n)}$ stands for n occurrences of formula A) is equivalent (for provability) to the triple of rules: contraction, weakening, dereliction.

Another possible exponential rule is the *selection* rule:

$$\frac{\Gamma \vdash A, ?A, \Delta}{\Gamma \vdash ?A, \Delta}\ (sel)$$

One can derive selection from dereliction and contraction and one can derive dereliction from selection and weakening. Thus in presence of contraction and weakening, selection and dereliction generate the same provable sequents.

3.7.2 Non-symmetric sequents

The same remarks that lead to the definition of the one-sided calculus can lead to the definition of other simplified systems:

- A one-sided variant with sequents of the form $\Gamma \vdash$ could be defined.
- When considering formulas up to De Morgan duality, an equivalent system is obtained by considering only the left and right rules for positive connectives (or the ones for negative connectives only, obviously).
- Intuitionistic Linear Logic is the two-sided system where the right-hand side is constrained to always contain exactly one formula (with a few associated restrictions), see below.
- Similar restrictions are used in various semantics and proof search formalisms.

3.7.2.1 Intuitionistic Linear Logic

The connectives of Intuitionistic Linear Logic (ILL) are not exactly the same as in LL since not only some connectives are rejected ($\_^{\perp}$, $\parr$, $\perp$ and ?), but also $\multimap$ is now a primitive connective. The ILL formulas are then obtained as:

$$I ::= \alpha \mid I \otimes I \mid I \multimap I \mid \mathbf{1} \mid I \oplus I \mid I \,\&\, I \mid \mathbf{0} \mid \top \mid !I \mid \forall \xi.I \mid \exists \xi.I$$

Sequents are two sided but their right-hand side contains exactly one formula: $\Gamma \vdash I$.

The rules are described in Table 3.5. For each connective of ILL, they are obtained from the corresponding rules of Table 3.2 by restricting to exactly one formula on the right-hand side of the sequents. The case of linear implication is slightly different since it is not a primitive connective of LL. However the

Identity group

$$\frac{}{I \vdash I}\ (ax) \quad \frac{\Gamma \vdash I \qquad \Delta, I \vdash K}{\Gamma, \Delta \vdash K}\ (cut)$$

Multiplicative group

$$\frac{\Gamma, I, J \vdash K}{\Gamma, I \otimes J \vdash K}\ (\otimes_L) \quad \frac{\Gamma \vdash K}{\Gamma, \mathbf{1} \vdash K}\ (\mathbf{1}_L) \quad \frac{\Gamma \vdash I \qquad \Delta \vdash J}{\Gamma, \Delta \vdash I \otimes J}\ (\otimes_R) \quad \frac{}{\vdash \mathbf{1}}\ (\mathbf{1}_R)$$

$$\frac{\Gamma \vdash I \qquad \Delta, J \vdash K}{\Gamma, \Delta, I \multimap J \vdash K}\ (\multimap_L) \quad \frac{\Gamma, I \vdash J}{\Gamma \vdash I \multimap J}\ (\multimap_R)$$

Additive group

$$\frac{\Gamma, I \vdash K \qquad \Gamma, J \vdash K}{\Gamma, I \oplus J \vdash K}\ (\oplus_L) \quad \frac{}{\Gamma, \mathbf{0} \vdash K}\ (\mathbf{0}_L) \quad \frac{\Gamma \vdash I_i}{\Gamma \vdash I_1 \oplus I_2}\ (\oplus_{Ri})$$

$$\frac{\Gamma, I_i \vdash K}{\Gamma, I_1 \,\&\, I_2 \vdash K}\ (\&_{Li}) \quad \frac{\Gamma \vdash I \qquad \Gamma \vdash J}{\Gamma \vdash I \,\&\, J}\ (\&_R) \quad \frac{}{\Gamma \vdash \top}\ (\top_R)$$

Quantifier group

In the rules $(\exists^1_L)$ (resp. $(\exists^2_L)$) and $(\forall^1_R)$ (resp. $(\forall^2_R)$),
the variable x (resp. X) must not occur free in Γ nor in K.

$$\frac{\Gamma, I \vdash K}{\Gamma, \exists x.I \vdash K}\ (\exists^1_L) \qquad \frac{\Gamma, I \vdash K}{\Gamma, \exists X.I \vdash K}\ (\exists^2_L) \qquad \frac{\Gamma \vdash I[t/x]}{\Gamma \vdash \exists x.I}\ (\exists^1_R) \qquad \frac{\Gamma \vdash I[J/X]}{\Gamma \vdash \exists X.I}\ (\exists^2_R)$$

$$\frac{\Gamma, I[t/x] \vdash K}{\Gamma, \forall x.I \vdash K}\ (\forall^1_L) \qquad \frac{\Gamma, I[J/X] \vdash K}{\Gamma, \forall X.I \vdash K}\ (\forall^2_L) \qquad \frac{\Gamma \vdash I}{\Gamma \vdash \forall x.I}\ (\forall^1_R) \qquad \frac{\Gamma \vdash I}{\Gamma \vdash \forall X.I}\ (\forall^2_R)$$

Exponential group

$$\frac{\Gamma, I \vdash K}{\Gamma, !I \vdash K}\ (!_L) \quad \frac{!\Gamma \vdash I}{!\Gamma \vdash !I}\ (!_R)$$

Structural group

$$\frac{\Gamma_1, I, J, \Gamma_2 \vdash K}{\Gamma_1, J, I, \Gamma_2 \vdash K}\ (ex_L) \quad \frac{\Gamma \vdash K}{\Gamma, !I \vdash K}\ (w_L) \quad \frac{\Gamma, !I, !I \vdash K}{\Gamma, !I \vdash K}\ (c_L)$$

Table 3.5: Inference rules for Intuitionistic Linear Logic sequent calculus

ILL rules can nevertheless be obtained by restricting the following derivations to intuitionistic sequents:

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta}{\Gamma, A^\perp \vdash \Delta}\,(n_L) \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A^\perp \parr B \vdash \Delta, \Delta'}\,(\parr_L) \qquad\qquad \dfrac{\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A^\perp, B, \Delta}\,(n_R)}{\Gamma \vdash A^\perp \parr B, \Delta}\,(\parr_R)$$

As a consequence, by translating each intuitionistic connective by the corresponding classical one, and $I \multimap J$ as $I^\perp \parr J$, each intuitionistic proof of a sequent $\Gamma \vdash I$ is translated into a classical proof of $\Gamma \vdash I$.

However the converse is not true. Some intuitionistic formulas not provable in intuitionistic Linear Logic are nevertheless provable once embedded in classical Linear Logic. This is the case of $((((\mathbf{0} \multimap \mathbf{1}) \multimap \mathbf{1}) \multimap \mathbf{0}) \multimap \mathbf{0}) \multimap \mathbf{1}$ which is not provable in ILL but whose associate classical formula $((((\top \parr \mathbf{1}) \otimes \bot) \parr \mathbf{0}) \otimes \top) \parr \mathbf{1}$ is provable in LL:

more about conservativity — Olivier

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \top, \mathbf{1}, \mathbf{0}}\,(\top_R)}{\vdash \top \parr \mathbf{1}, \mathbf{0}}\,(\parr_R) \qquad \dfrac{}{\vdash \bot, \mathbf{1}}\,(ax)}{\vdash (\top \parr \mathbf{1}) \otimes \bot, \mathbf{0}, \mathbf{1}}\,(\otimes_R)}{\vdash ((\top \parr \mathbf{1}) \otimes \bot) \parr \mathbf{0}, \mathbf{1}}\,(\parr_R) \qquad \dfrac{}{\vdash \top}\,(\top_R)}{\vdash (((\top \parr \mathbf{1}) \otimes \bot) \parr \mathbf{0}) \otimes \top, \mathbf{1}}\,(\otimes_R)}{\vdash ((((\top \parr \mathbf{1}) \otimes \bot) \parr \mathbf{0}) \otimes \top) \parr \mathbf{1}}\,(\parr_R)$$

3.7.3 Mix rules

It is quite common to consider mix rules:

$$\dfrac{}{\vdash}\,(mix_0) \qquad\qquad \dfrac{\Gamma \vdash \Delta \qquad \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,(mix_2)$$

They immediately impact provability since the empty sequent $\vdash$ becomes provable. While this could lead to a degeneracy of the system if structural rules were fully available, it has not such a deep impact on Linear Logic. Let us consider two (different) notions of consistency.

Definition 3.7.1 (Strong Consistency)**.** A variant of Linear Logic (in which the cut rule is admissible) is **strongly consistent** if one of the following equivalent properties holds:

- $\vdash$ is not provable
- $\vdash \bot$ is not provable
- $\mathbf{1} \vdash \bot$ is not provable
- there is no formula A such that both $\vdash A$ and $\vdash A^\perp$ are provable

- there is no formula A such that $\vdash A \otimes A^{\perp}$ is provable

Definition 3.7.2 (Weak Consistency)**.** A variant of Linear Logic (in which the cut rule is admissible) is **weakly consistent** if one of the following equivalent properties holds:

- $\vdash \mathbf{0}$ is not provable
- there exists a formula which is not provable
- there exists a formula A such that $\vdash A \otimes A^{\perp}$ is not provable

While LL is strongly consistent (Theorem 3.5.11), its extension with mix rules is only weakly consistent. To prove this we also rely on cut elimination. The proof described in Section 3.5 can be easily adapted by simply adding the following new reduction cases:

$$\dfrac{\dfrac{\pi_1 : \vdash \Gamma, C \qquad \pi_2 : \vdash \Gamma'}{\vdash \Gamma, C, \Gamma'}\,(mix_2) \qquad \pi_3 : \vdash C^{\perp}, \Delta}{\vdash \Gamma, \Delta, \Gamma'}\,(cut)$$

$$\xrightarrow{\mathsf{comm}(mix)} \qquad \dfrac{\dfrac{\pi_1 : \vdash \Gamma, C \qquad \pi_3 : \vdash C^{\perp}, \Delta}{\vdash \Gamma, \Delta}\,(cut) \qquad \pi_2 : \vdash \Gamma'}{\vdash \Gamma, \Delta, \Gamma'}\,(mix_2)$$

(and the symmetric one if C belongs to the right premise of the (mix_2) rule).

Now weak consistency comes from the fact that, in a cut-free proof, there is no rule with possible conclusion $\vdash \mathbf{0}$.

3.7.4 Dyadic sequent calculus

We consider here an alternative to the one-sided sequent calculus of Linear Logic which relies on a different management of the (restricted) structural rules. Sequents are built from two lists of formulas and denoted $\vdash \Sigma \mid \Gamma$. Intuitively elements of Σ are prefixed with an implicit ? connective. The rules are presented on Table 3.6.

Theorem 3.7.3 (Dyadic provability)**.** *The sequent $\vdash \Gamma \mid \Delta$ is provable in the dyadic system if and only if the sequent $\vdash ?\Gamma, \Delta$ is provable in the one-sided system of Table 3.4.*

Proof. The embedding from the dyadic system to usual one-sided rules goes directly since the translation of each rule happens to be derivable. In the converse direction, one starts from a proof in the usual system and commutes weakening and contraction rules as far as possible towards the leaves of the proof. The obtained proof can be easily turned into a dyadic one. □

Identity group

$$\frac{}{\vdash \Sigma \mid F^{\perp}, F}\ (ax) \qquad \frac{\vdash \Sigma \mid F, \Gamma \quad \vdash \Sigma \mid F^{\perp}, \Delta}{\vdash \Sigma \mid \Gamma, \Delta}\ (cut)$$

Multiplicative group

$$\frac{\vdash \Sigma \mid F, \Gamma \quad \vdash \Sigma \mid G, \Delta}{\vdash \Sigma \mid F \otimes G, \Gamma, \Delta}\ (\otimes) \qquad \frac{}{\vdash \Sigma \mid \mathbf{1}}\ (\mathbf{1}) \qquad \frac{\vdash \Sigma \mid F, G, \Gamma}{\vdash \Sigma \mid F \parr G, \Gamma}\ (\parr) \qquad \frac{\vdash \Sigma \mid \Gamma}{\vdash \Sigma \mid \perp, \Gamma}\ (\perp)$$

Additive group

$$\frac{\vdash \Sigma \mid F_i, \Gamma}{\vdash \Sigma \mid F_1 \oplus F_2, \Gamma}\ (\oplus_i) \qquad \frac{\vdash \Sigma \mid F, \Gamma \quad \vdash \Sigma \mid G, \Gamma}{\vdash \Sigma \mid F \,\&\, G, \Gamma}\ (\&) \qquad \frac{}{\vdash \Sigma \mid \top, \Gamma}\ (\top)$$

Quantifier group

In the rule $(\forall^1)$ (resp. $(\forall^2)$), the variable x (resp. X) must not occur free in Σ and Γ.

$$\frac{\vdash \Sigma \mid F[t/x], \Gamma}{\vdash \Sigma \mid \exists x.F, \Gamma}\ (\exists^1) \qquad \frac{\vdash \Sigma \mid F[B/X], \Gamma}{\vdash \Sigma \mid \exists X.F, \Gamma}\ (\exists^2)$$

$$\frac{\vdash \Sigma \mid F, \Gamma}{\vdash \Sigma \mid \forall x.F, \Gamma}\ (\forall^1) \qquad \frac{\vdash \Sigma \mid F, \Gamma}{\vdash \Sigma \mid \forall X.F, \Gamma}\ (\forall^2)$$

Exponential group

$$\frac{\vdash \Sigma \mid F}{\vdash \Sigma \mid !F}\ (!) \qquad \frac{\vdash \Sigma, F \mid \Gamma}{\vdash \Sigma \mid ?F, \Gamma}\ (?)$$

Structural group

$$\frac{\vdash \Sigma \mid \Gamma, F, G, \Delta}{\vdash \Sigma \mid \Gamma, G, F, \Delta}\ (ex) \qquad \frac{\vdash \Sigma, F, \Sigma' \mid F, \Gamma}{\vdash \Sigma, F, \Sigma' \mid \Gamma}\ (sel)$$

Table 3.6: Inference rules for dyadic sequent calculus

3.8 Embedding into classical logic

Linear Logic provides a refinement of classical logic by constraining structural rules. An key consequence is the emergence of the distinction between multiplicative and additive connectives related through the exponential ones. It is possible to see that this refinement is faithful to classical provability since if one decides to put back structural rules and to forget the distinction between multiplicative and additive connectives, one can map proofs of Linear Logic into valid proofs of classical logic. More formally we define the notion of **skeleton** of linear formulas and Linear Logic proofs.

The skeleton $\mathrm{sk}(A)$ of a formula A is obtained by applying the following translation table to each linear connective:

$$\begin{array}{lll} \bot & \mapsto & \neg \\ \otimes & \mapsto & \wedge \\ \parr & \mapsto & \vee \\ \mathbf{1} & \mapsto & \top \\ \bot & \mapsto & \bot \\ ! & \mapsto & \\ \forall & \mapsto & \forall \end{array} \qquad \begin{array}{lll} \& & \mapsto & \wedge \\ \oplus & \mapsto & \vee \\ \top & \mapsto & \top \\ \mathbf{0} & \mapsto & \bot \\ ? & \mapsto & \\ \exists & \mapsto & \exists \end{array}$$

Note the exponential connectives are simply erased.

Theorem 3.8.1 (Skeleton)**.** *Given a (two-sided) proof π of the sequent $\Gamma \vdash \Delta$, there exists a classical proof (see Table 3.7) of* $\mathrm{sk}(\Gamma) \vdash \mathrm{sk}(\Delta)$.

Proof. We prove for each rule of Table 3.2 that its image under sk is derivable using the rules of Table 3.7:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \dots \quad \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta} \quad \mapsto \quad \frac{\mathrm{sk}(\Gamma_1) \vdash \mathrm{sk}(\Delta_1) \quad \dots \quad \mathrm{sk}(\Gamma_n) \vdash \mathrm{sk}(\Delta_n)}{\mathrm{sk}(\Gamma) \vdash \mathrm{sk}(\Delta)}$$

It is immediate for most of the rules. Let us focus on the richer case of the $\otimes$ rules:

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\ (\otimes_L) \quad \mapsto \quad \dfrac{\dfrac{\dfrac{\mathrm{sk}(\Gamma), \mathrm{sk}(A), \mathrm{sk}(B) \vdash \mathrm{sk}(\Delta)}{\mathrm{sk}(\Gamma), \mathrm{sk}(A) \wedge \mathrm{sk}(B), \mathrm{sk}(B) \vdash \mathrm{sk}(\Delta)}\ (\wedge_{L1})}{\mathrm{sk}(\Gamma), \mathrm{sk}(A) \wedge \mathrm{sk}(B), \mathrm{sk}(A) \wedge \mathrm{sk}(B) \vdash \mathrm{sk}(\Delta)}\ (\wedge_{L2})}{\mathrm{sk}(\Gamma), \mathrm{sk}(A) \wedge \mathrm{sk}(B) \vdash \mathrm{sk}(\Delta)}\ (c_L)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\ (\otimes_R)$$

$$\mapsto \quad \dfrac{\dfrac{\dfrac{\mathrm{sk}(\Gamma) \vdash \mathrm{sk}(A), \mathrm{sk}(\Delta)}{\mathrm{sk}(\Gamma), \mathrm{sk}(\Gamma') \vdash \mathrm{sk}(A), \mathrm{sk}(\Delta)}\ (w_L)}{\mathrm{sk}(\Gamma), \mathrm{sk}(\Gamma') \vdash \mathrm{sk}(A), \mathrm{sk}(\Delta), \mathrm{sk}(\Delta')}\ (w_R) \quad \dfrac{\dfrac{\mathrm{sk}(\Gamma') \vdash \mathrm{sk}(B), \mathrm{sk}(\Delta')}{\mathrm{sk}(\Gamma), \mathrm{sk}(\Gamma') \vdash \mathrm{sk}(B), \mathrm{sk}(\Delta')}\ (w_L)}{\mathrm{sk}(\Gamma), \mathrm{sk}(\Gamma') \vdash \mathrm{sk}(B), \mathrm{sk}(\Delta), \mathrm{sk}(\Delta')}\ (w_R)}{\mathrm{sk}(\Gamma), \mathrm{sk}(\Gamma') \vdash \mathrm{sk}(A) \wedge \mathrm{sk}(B), \mathrm{sk}(\Delta), \mathrm{sk}(\Delta')}\ (\wedge_R)$$

□

Identity and negation group

$$\frac{}{\Gamma, A \vdash A, \Delta}\ (ax) \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ (cut) \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ (n_L)$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ (n_R)$$

Logical group

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ (\vee_L) \qquad \frac{}{\Gamma, \bot \vdash \Delta}\ (\bot_L) \qquad \frac{\Gamma \vdash A_i, \Delta}{\Gamma \vdash A_1 \vee A_2, \Delta}\ (\vee_{Ri})$$

$$\frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \wedge A_2 \vdash \Delta}\ (\wedge_{Li}) \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ (\wedge_R) \qquad \frac{}{\Gamma \vdash \top, \Delta}\ (\top_R)$$

Quantifier group

In the rules $(\exists^1_L)$ (resp. $(\exists^2_L)$) and $(\forall^1_R)$ (resp. $(\forall^2_R)$),
the variable x (resp. X) must not occur free in Γ nor in Δ.

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists x.A \vdash \Delta}\ (\exists^1_L) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma, \exists X.A \vdash \Delta}\ (\exists^2_L) \qquad \frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x.A, \Delta}\ (\exists^1_R) \qquad \frac{\Gamma \vdash A[B/X], \Delta}{\Gamma \vdash \exists X.A, \Delta}\ (\exists^2_R)$$

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x.A \vdash \Delta}\ (\forall^1_L) \qquad \frac{\Gamma, A[B/X] \vdash \Delta}{\Gamma, \forall X.A \vdash \Delta}\ (\forall^2_L) \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x.A, \Delta}\ (\forall^1_R) \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall X.A, \Delta}\ (\forall^2_R)$$

Structural group

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}\ (ex_L) \qquad \frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2}\ (ex_R)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ (w_L) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ (w_R) \qquad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ (c_L) \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ (c_R)$$

Table 3.7: Inference rules for two-sided Classical Logic sequent calculus

create a chapter about translations of LK and LJ into LL, the only existing part is translation of the λ-calculus into proof-nets in Chapter 4 — Olivier

General comments:

- use lists and not multisets in order not to blur computational content. Introduce derivable rules:introduction + exchange -> OK
- let's be a bit more precise on the structure of proofs, right? finite trees labelled by a pair of sequent and inference rule. -> OK or alternative suggestion by Michele
- presentation of the sequent calculus: distinguish logical group from structural and identity group. -> OK, present ?w and ?c as structural rules.
- make uniform and consistent choice for one-sided vs two-sided. -> OK
- section cut-elim and consequence: as stated, this should be referred to as cut-admissibility. -> ask the question. either cut-elim = cut-admissibility and one speak about weak/strong normalization or cut-elim has more constructive content than cut-admissibility.
- some basic notions are required: descendents, derivable vs admissible rules, isomorphism, etc.
- reversibility: why not using cut-elimination result?
- focusing: why not a general result for the exponential as well?
- add final section with bibliographic discussion? -> OK plus comment in the main body. Build a file for gouping bibliographic comments.
- discuss fragments (in particular ILL)
- what about decidability questions?

Notations:

- why naming the rules in table 2?
- introduce notation pi: Gamma vdash Delta

— Alexis

Chapter 4

Proof nets

We give some basic results of the theory of proof nets for multiplicative linear logic and multiplicative exponential linear logic with mix. The relation between proof nets and the lambda-calculus is precisely described.

Warning! *Only the case of multiplicative proof nets is currently in a satisfactory state, although some introductory material and pictures are missing. The treatment of* MELL *is under construction.*

4.1 Introduction

Proof nets are one of the most striking contributions of linear logic. At first sight, they are the analogue of natural deduction for linear logic: a logical syntax avoiding the bureaucracy of sequent calculus. More precisely, they can be considered as a representation of proofs in the one-sided fragment of sequent calculus — see Section 3.3.5 — up to some inessential commutations of rules.

Due to their graphical nature, and to the many (essentially equivalent but technically distinctive) choices one can make to define them, proof nets are also one of the less consensual topics within the theory of linear logic. Their definition varies throughout the literature, which makes the reuse of results somehow fragile. And, up to this point, there is no published reference text covering the details of the most basic definitions and fundamental results, starting from the ideal case of MLL without units to the simulation of the β-reduction of the λ-calculus in MELL proof nets, *via* confluence and strong normalizability of each considered fragment. The present chapter aims precisely to address this demand.

A few words about our graph theoretical terminology: We assume basic knowledge in graph theory. In particular, the notion of paths and cycles will play a prominent rôle. We provide a complete review of the definitions and

results we use in Appendix A, but for the most part it will suffice to make the setting and terminology precise:

- we consider directed graphs only (that we simply call *graphs*), which are given by a set of nodes $\mathcal{N}$, a set of arrows $\mathcal{A}$, and two functions $\mathtt{s}$ (source) and $\mathtt{t}$ (target) from arrows to nodes;
- we call *edge* the data of an arrow and a direction (forward or backward) in which the arrow is crossed, and we denote a^+ (resp. a^-) for the arrow a crossed forward (resp. backward);
- *paths* are undirected in general, and are given by a source node and a sequence of edges (one obtains a directed path if all the underlying arrows are crossed forward);
- a path is *simple* if each arrow is crossed at most once (in whatever direction) and a *cycle* is a non-empty simple *closed* path.

To the newcomer: The reader intending to discover proof nets by reading this chapter should jump directly to Section 4.2, which presents multiplicative proof nets. These form the most satisfactory version of proof nets, especially in the unit-free case: nodes in a proof net correspond exactly with inference rules in a proof tree, cut elimination is strongly confluent and terminating, and the translation of proof trees is exactly characterized by a graphical correctness criterion.

Every other notion of proof net is only an attempt to recover the features of this ideal case in a more general setting.

To the more knowledgeable reader: Our choice of definition of proof nets is guided by our desire to rely on standard results of graph theory. We follow the "rules as nodes/formulas as arrows" (rather than "formulas as nodes/rules as hyperarrows") approach. We also avoid the introduction of half edges or dangling wires corresponding to conclusions: for that reason, we have explicit conclusion nodes.

Typing is not mandatory: the basic definition of proof structure is untyped, and typing is seen as a labelling of arrows. We call proof nets the structures satisfying the Danos–Regnier acyclicity criterion, and connected proof nets those that satisfy the acyclicity-and-connectedness criterion. In the unit-free multiplicative case, a proof tree is thus translated into a typed connected proof net.

The most peculiar feature of our presentation is the proof of sequentialization. It relies on the so-called *Bungee Jumping* method, to first obtain an order on arrows, such that the target of a maximal (non-conclusion) arrow is either terminal $⅋$-node or a splitting $\otimes$-node Section 4.2.4.4. This method leverages general results on *switching paths* (a path in a proof structure is a switching path if is also a path in some switching graph of the structure), that we develop in Section 4.2.4.3. The Bungee Jumping method can be applied to proof nets, without the connectedness assumption, to sequentialize in the presence of (*mix*)

rules. We first treat the connected case, though, so that newcomers can get acquainted with the subject in the most ideal case.

4.2 Multiplicative Proof Nets

We first consider multiplicative proof nets without units, *i.e.* proof nets for the MLL_v fragment — see Section 3.4. These will represent proofs in the one-sided fragment of sequent calculus — see Section 3.3.5 — up to some inessential commutations of rules. Note in particular that we consider formulas up to De Morgan's laws so that linear negation $A \mapsto A^\perp$ is an involution — see Section 3.3.5 again.

4.2.1 Proof structures

4.2.1.1 Forgetting Sequential Structure

The following two proofs:

$$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax) \quad \dfrac{}{\vdash B, B^\perp}\,(ax)}{\vdash A, A^\perp \otimes B, B^\perp}\,(\otimes) \quad \dfrac{}{\vdash C, C^\perp}\,(ax)}{\vdash A, A^\perp \otimes B, B^\perp \otimes C, C^\perp}\,(\otimes)$$

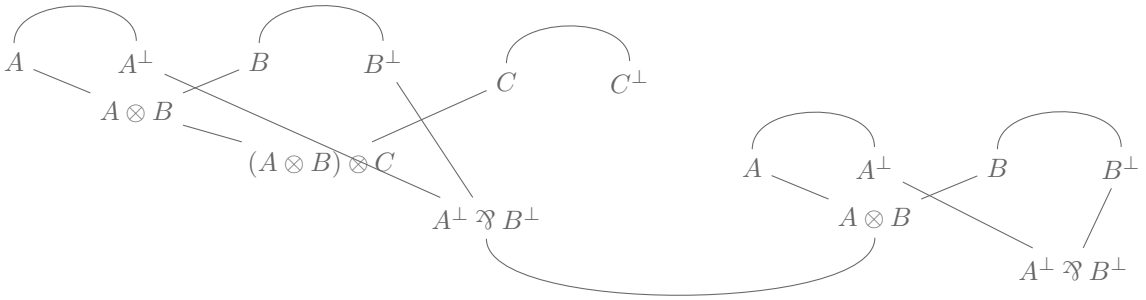
and

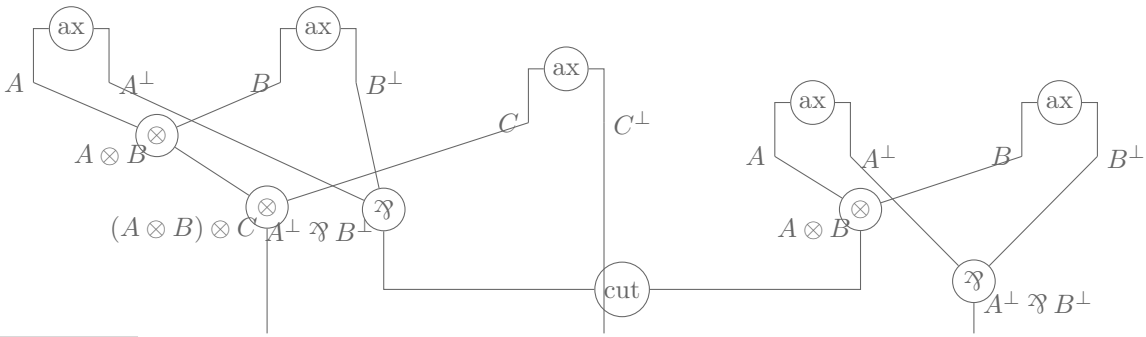
$$\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax) \quad \dfrac{\dfrac{}{\vdash B, B^\perp}\,(ax) \quad \dfrac{}{\vdash C, C^\perp}\,(ax)}{B, B^\perp \otimes C, C^\perp}\,(\otimes)}{\vdash A, A^\perp \otimes B, B^\perp \otimes C, C^\perp}\,(\otimes)$$

differ only by the order in which we apply the two $(\otimes)$ rules.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax) \quad \dfrac{}{\vdash B, B^\perp}\,(ax)}{\vdash A \otimes B, A^\perp, B^\perp}\,(\otimes) \quad \dfrac{}{\vdash C, C^\perp}\,(ax)}{\vdash (A \otimes B) \otimes C, A^\perp, B^\perp, C^\perp}\,(\otimes)}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}\,(\parr) \quad \dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax) \quad \dfrac{}{\vdash B, B^\perp}\,(ax)}{\vdash A \otimes B, A^\perp, B^\perp}\,(\otimes)}{\vdash (A \otimes B) \otimes C, A^\perp, B^\perp, C^\perp}\,(cut)}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}\,(\parr)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}(ax) \quad \dfrac{}{\vdash B, B^\perp}(ax)}{\vdash A \otimes B, A^\perp, B^\perp}(\otimes) \quad \dfrac{}{\vdash C, C^\perp}(ax)}{\vdash (A \otimes B) \otimes C, A^\perp, B^\perp, C^\perp}(\otimes)}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}(\parr) \quad \dfrac{\dfrac{}{\vdash A, A^\perp}(ax) \quad \dfrac{}{\vdash B, B^\perp}(ax)}{\vdash A \otimes B, A^\perp, B^\perp}(\otimes)}{\vdash (A \otimes B) \otimes C, A^\perp, B^\perp, C^\perp}(cut)}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}(\parr)$$





TODO

4.2.1.2 Definition of proof structures

Recall that the definitions and abstract properties of graphs we are going to use can be found in Appendix A.

An **untyped multiplicative proof structure** $\mathcal{S}$ is given by:

- a *directed acyclic graph* $\mathcal{G}(\mathcal{S})$ — the *incoming arrows* of a node are called its **premises**, the *outgoing arrows* are its **conclusions**;
- a labelling of the nodes of $\mathcal{G}(\mathcal{S})$ with labels in $\{ax, cut, \otimes, \parr, \bullet\}$ — a node with label k is called a k-node — such that:
  - each ax-node has exactly two conclusions and no premise;

  - each *cut*-node has exactly two premises and no conclusion;
  - each $\otimes$-node or $\parr$-node has exactly two premises and one conclusion;
  - each $\bullet$-node has exactly one premise and no conclusion;
- an ordering of the premises of each $\otimes$-node and of each $\parr$-node — the first premise is called the left premise of the node, and the second one is the right premise;
- a linear ordering on the $\bullet$-nodes, which we call the **interface**, or **conclusion sequence**, of $\mathcal{S}$.

A $\bullet$-node is called a **conclusion node**.

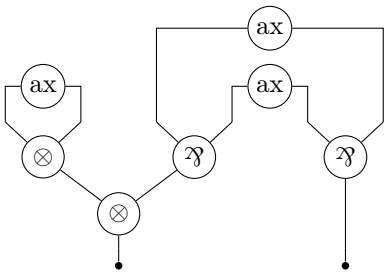
Some notions do not depend on the ordering of edges (premises of $\otimes$- and $\parr$-nodes) nor of conclusion nodes (interface): this is in particular the case for the graphical notion of correctness that we will study in Sections 4.2.2 and 4.2.4. We will call **unordered proof structure** the object obtained by forgetting the interface and the ordering of premises in a proof structure: this is simply given by a directed acyclic graph, together with a labelling of nodes, satisfying the same constraints as those for proof structures.

Nodes which are not conclusion nodes are called **internal nodes** of the proof structure. A **premise node** of some node N is any internal node N_0 such that one conclusion of N_0 is also a premise of N. Nodes that are not premises of internal nodes (or, equivalently, which are premises of $\bullet$-nodes only) are called **terminal**. Note that *cut*-nodes are both internal and terminal. By definition a non-empty proof structure must contain at least one *ax* node and at least one terminal node.

An arrow is called a **conclusion arrow** (resp. an **internal arrow** or **terminal arrow**), when its target a conclusion (resp. internal or terminal) node. Moreover, we may indifferently call conclusion of the proof structure any of a conclusion node or arrow, depending on the context.

In the graphical representation of a proof structure, we do not mention explicitly the direction of arrows, but we draw them in such a way that direction is represented in a top-down way, which is always possible thanks to directed acyclicity. Nodes are depicted as circles, each with its node label, except for conclusion nodes which are simply represented as bullets. We use the convention that the interface is given by the order in which the conclusions appear at the bottom, from left to right.

Example 4.2.1 (Untyped Proof Structure). Consider the untyped proof structure



draw it also with directed edges, ordered premises, ordered conclusions since it is the first example — Olivier

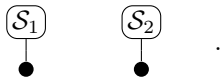
which has 9 nodes (7 internal ones), 2 conclusions and 2 terminal nodes above these conclusions: a $\otimes$-node and a $\parr$-node.

This graphical depiction makes evident that we should actually consider proof structures up to "renaming" of internal nodes. More precisely we say two proof structures $\mathcal{S}$ and $\mathcal{S}'$ are **isomorphic** if there exists a graph isomorphism $\Phi : \mathcal{G}(\mathcal{S}) \xrightarrow{\sim} \mathcal{G}(\mathcal{S}')$, preserving the labelling of nodes, the ordering of premises and the interface. In the following, unless explicitly stated, we will identify isomorphic proof structures: it should be clear that all the notions we define in the next sections (typing, correctness, cut elimination, *etc.*) are compatible with such isomorphisms.

The **empty proof structure** is the only proof structure whose underlying graph is empty. Given two proof structures $\mathcal{S}_1$ and $\mathcal{S}_2$, we write $\mathcal{S}_1 + \mathcal{S}_2$ for the **sum of proof structures**, with $\mathcal{G}(\mathcal{S}_1 + \mathcal{S}_2) = \mathcal{G}(\mathcal{S}_1) + \mathcal{G}(\mathcal{S}_2)$ and such that: the labelling of nodes and the order of premises is inherited from those of $\mathcal{S}_1$ and $\mathcal{S}_2$; and the interface is the concatenation of those of $\mathcal{S}_1$ and $\mathcal{S}_2$.

We may depict an arbitrary structure $\mathcal{S}$ by

where the larger conclusion dot stands for an arbitrary number of conclusions. Then the sum $\mathcal{S}_1 + \mathcal{S}_2$ is naturally depicted by



.

Many notions and results about proof structures involve the notion of path: a **path in** $\mathcal{S}$ is just a *path* in $\mathcal{G}(\mathcal{S})$. Depending on the context, we will in fact consider various restrictions of that notion. For instance, we will call **descent path** any path in $\mathcal{S}$ that is a *directed path* of $\mathcal{G}(\mathcal{S})$: graphically, these are the paths going downwards. Since $\mathcal{G}(\mathcal{S})$ is directed acyclic, it is immediate that:

- no descent path is a cycle;
- every descent path is a simple path;
- the length of descent paths is bounded.

A **full descent path** is a descent path that cannot be extended downwards (it is not a strict prefix of another descent path): the target of a full descent path must be a *cut*-node or a conclusion. Note that each arrow in a proof structure is the first arrow of exactly one full descent path: except for *ax*-nodes, which have no premise arrow, each node has at most one conclusion.

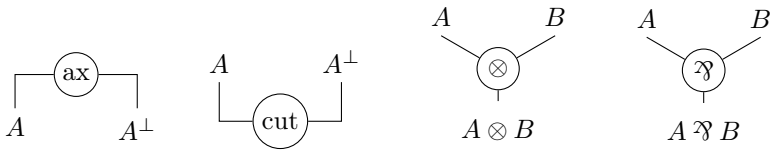


Figure 4.1: Typing of proof structures.

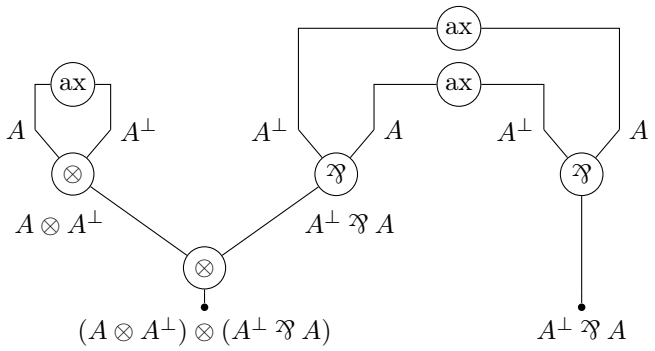
4.2.1.3 Typed proof structures

A **typing** of an untyped proof structure is a labelling of its arrows with formulas of MLL_v — the label of an arrow is called its **type** — such that:

- the conclusions of an *ax*-node (resp. the premises of a *cut*-node) have dual types;
- if the left premise of a $\otimes$-node (resp. a $\parr$-node) has type A and its right premise has type B then its conclusion has type $A \otimes B$ (resp. $A \parr B$).

We depict these constraints in Fig. 4.1. A **typed proof structure** is given by an untyped proof structure together with such a typing. With the conclusion sequence of a typed proof structure $\mathcal{S}$ is associated the **conclusion sequent** of $\mathcal{S}$, i.e. the sequence of the types of its conclusions (in the order given by the interface of $\mathcal{S}$). Note that the empty graph is a typed proof structure, whose conclusion is the empty sequent $\vdash$.

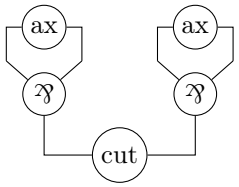
Example 4.2.2 (Typed Proof Structure). We can type the previous example of untyped proof structure as follows

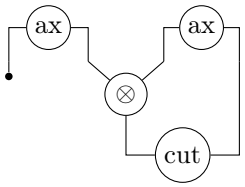


yielding a typed proof structure with conclusion $\vdash (A \otimes A^\perp) \otimes (A^\perp \parr A), A^\perp \parr A$.

Example 4.2.3 (Non typeable Proof Structure). It is easy to check that the

following proof structures do not admit any typing:

$\mathcal{S}_{clash} =$ 

and $\mathcal{S}_{fix} =$ 

.

Indeed, in $\mathcal{S}_{clash}$, it is impossible to type the premises of the *cut*-node with dual types, because the main connective must be $⅋$ on both sides. And in $\mathcal{S}_{fix}$, the conclusion of the $\otimes$-node must be typed with some formula of the form $A \otimes B$, but the constraints on *ax*- and *cut*-nodes impose that $B = A \otimes B$, which is impossible.

Note that the previous two examples of untypeable proof structures involve *cut*-nodes. Indeed, any cut-free proof structure is typeable: it suffices to pick a suitable typing of the conclusions of *ax*-nodes, and to propagate the typing from top to bottom. More precisely:

Proposition 4.2.4 (Typings of proof structures)**.** *Given a proof structure $\mathcal{S}$, each typing of $\mathcal{S}$ is uniquely determined by its value on the conclusions of ax-nodes. Moreover, if $\mathcal{S}$ is cut-free, any choice of dual formulas for the conclusions of each ax-node of $\mathcal{S}$ induces a typing of the whole proof structure.*

Proof. Since $\mathcal{G}(\mathcal{S})$ is a directed acyclic graph, we can:

- start from a typing of the conclusions of each *ax*-nodes with dual types;
- deduce the type of the conclusion of each $\otimes$- or $⅋$-node from that of its premises.

This determines uniquely a type for all the arrows of $\mathcal{S}$. If the types thus assigned to premises of *cut*-nodes are dual formulas, then we obtain a typing of the proof structure; otherwise, there is no typing with the choice of axioms we started with. □

Remark 4.2.5. In presence of typing, the directed acyclicity requirement on $\mathcal{G}(\mathcal{S})$ is redundant: typing conditions are sufficient to ensure that there is no directed cycle. Indeed the only nodes with both premises and conclusions (*i.e.* incoming and outgoing arrows) are those labelled $\otimes$ and $⅋$: in this case the definition imposes that premises are typed with an immediate subformula of the conclusion.

In the more general setting of multiplicative-exponential proof nets to be introduced later, this will no longer hold and directed acyclicity must be required explicitly. Moreover, the notion of untyped structure is also relevant *per se*: for instance, typing plays no rôle in the correctness criteria nor in the definition of cut elimination.

Insert a reference to a figure where we see why in presence of exponentials directed acyclicity is required. — Lorenzo

In the following, we will simply call *proof structure* any untyped proof structure, that might or might not be associated with a typed proof structure.

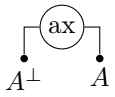
4.2.1.4 Translation of proof trees into typed proof structures

Recall that the (one-sided) rules of the sequent calculus MLL_v are those of the identity and multiplicative groups in Table 3.2, plus the exchange rule (ex).

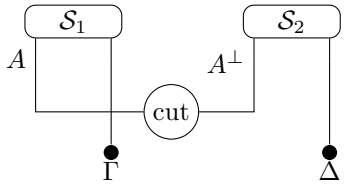
A proof π of MLL_v can be translated into a typed proof structure $\mathtt{ps}(\pi)$ with the same conclusion sequent. The internal nodes of $\mathtt{ps}(\pi)$ correspond with the rules of π (except for exchange rules), and each node is labelled with the name of its corresponding rule. The premises and conclusions of each node are labelled with active formulas of the corresponding rule; in particular the conclusion of a $\parr$- or $\otimes$-node is labelled with the principal formula. It is important that we treat the exchange rule explicitly, to keep the interface of the proof structure consistent with the occurrences of formulas in the conclusion sequent: the translation of the (ex) amounts to reordering the interface.

Definition 4.2.6 (Desequentialization of proof trees)**.** To each proof π of MLL_v, we associate a typed proof structure $\mathtt{ps}(\pi)$ by induction on the structure of π:

- An (ax) rule $\dfrac{}{\vdash A^\perp, A}\,(ax)$ is translated into an ax-node with conclusions labelled $A^\perp$ and A which have $\bullet$-nodes as targets.[1]

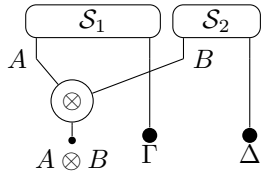


- If π_1 is translated into $\mathcal{S}_1 = \mathtt{ps}(\pi_1)$ and π_2 is translated into $\mathcal{S}_2 = \mathtt{ps}(\pi_2)$, then with the proof $\dfrac{\pi_1 \quad \pi_2}{\vdash A, \Gamma \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\,(cut)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1 + \mathcal{S}_2$ by removing the $\bullet$-nodes with premises a_1 labelled A and a_2 labelled $A^\perp$, and by introducing a new cut-node with premises a_1 and a_2.

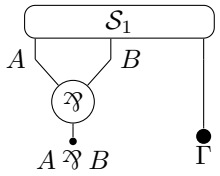


[1] To define the proof structure formally we should also give the interface: we order the conclusions to match the sequent $\vdash A^\perp, A$. In the remaining cases, we keep this information implicit, as it is univocally guided by the shape of inference rules, and it is faithfully reflected in our depictions of the resulting structures. We apply the same policy to the ordering of premises of $\otimes$- and $\parr$-nodes.

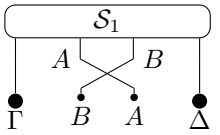
- If π_1 is translated into $\mathcal{S}_1 = \mathtt{ps}(\pi_1)$ and π_2 is translated into $\mathcal{S}_2 = \mathtt{ps}(\pi_2)$, then with the proof $\dfrac{\overset{\pi_1}{\vdash A, \Gamma} \quad \overset{\pi_2}{\vdash B, \Delta}}{\vdash A \otimes B, \Gamma, \Delta}\ (\otimes)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1 + \mathcal{S}_2$ by removing the $\bullet$-nodes with premises a_1 labelled A and a_2 labelled B, and by introducing a new $\otimes$-node with premises a_1 and a_2 and with conclusion a new arrow labelled $A \otimes B$ which is itself the premise of a new $\bullet$-node.



- If π_1 is translated into $\mathcal{S}_1 = \mathtt{ps}(\pi_1)$, then with the proof $\dfrac{\overset{\pi_1}{\vdash A, B, \Gamma}}{\vdash A \parr B, \Gamma}\ (\parr)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1$ by removing the $\bullet$-nodes with premises a_1 labelled A and a_2 labelled B, and by introducing a new $\parr$-node with premises a_1 and a_2 and with conclusion a new arrow labelled $A \parr B$ which is itself the premise of a new $\bullet$-node.



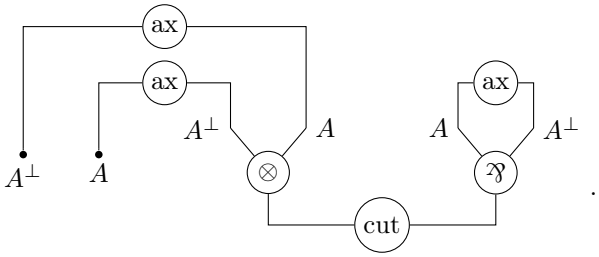
- If π_1 is translated into $\mathcal{S}_1 = \mathtt{ps}(\pi_1)$, then with the proof $\dfrac{\overset{\pi_1}{\vdash \Gamma, A, B, \Delta}}{\vdash \Gamma, B, A, \Delta}\ (ex)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1$ by exchanging the $\bullet$-nodes with premises labelled A and B in the conclusion sequence (thus only the interface is modified).



Exercise 4.2.7. Compute $\mathtt{ps}(\pi)$ with

$$\pi := \dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\,(ax) \quad \dfrac{}{\vdash A, A^\perp}\,(ax)}{\vdash A^\perp \otimes A, A, A^\perp}\,(\otimes) \qquad \dfrac{\dfrac{}{\vdash A, A^\perp}\,(ax)}{\vdash A \parr A^\perp}\,(\parr)}{\dfrac{\vdash A, A^\perp}{\vdash A^\perp, A}\,(ex)}\,(cut)$$

and check that it is isomorphic to



.

Exercise 4.2.8. Compute $\mathtt{ps}(\pi)$ and $\mathtt{ps}(\pi')$ with

$$\pi := \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi_1 \quad \pi_2}{\vdash A, C, D, \Gamma_1 \quad \vdash B, E, \Gamma_2}\,(\otimes)}{\vdash A \otimes B, C, D, \Gamma_1, E, \Gamma_2}\,(ex)}{\vdash C, D, A \otimes B, \Gamma_1, E, \Gamma_2}\,(\parr)}{\vdash C \parr D, A \otimes B, \Gamma_1, E, \Gamma_2}\,(ex)}{\vdash E, C \parr D, A \otimes B, \Gamma_1, \Gamma_2} \qquad \pi_3 \quad \vdash E^\perp, \Gamma_3}\,(cut)}{\vdash C \parr D, A \otimes B, \Gamma_1, \Gamma_2, \Gamma_3}$$

and

$$\pi' := \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\vdash A, C, D, \Gamma_1}\,(ex)}{\vdash C, D, A, \Gamma_1}\,(\parr)}{\vdash C \parr D, A, \Gamma_1}\,(ex)}{\vdash A, C \parr D, \Gamma_1} \qquad \pi_2 \quad \vdash B, E, \Gamma_2}\,(\otimes)}{\vdash A \otimes B, C \parr D, \Gamma_1, E, \Gamma_2}\,(ex)}{\vdash E, C \parr D, A \otimes B, \Gamma_1, \Gamma_2} \qquad \pi_3 \quad \vdash E^\perp, \Gamma_3}\,(cut)}{\vdash C \parr D, A \otimes B, \Gamma_1, \Gamma_2, \Gamma_3}$$

where π_1, π_2 and π_3 are proof trees with the appropriate conclusions. Check that $\mathtt{ps}(\pi)$ and $\mathtt{ps}(\pi')$ are isomorphic and find a third proof with the same translation.

It is then natural to try to analyse the kernel of the translation $\mathtt{ps}$ by understanding when two different sequent calculus proofs are mapped to the same typed proof structure. One can prove that it is the case if and only if one can transform one of the two proofs to the other by some permutations of the order of application of rules.

We should prove this result formally. Step 1: list the commutation rules to consider. Step 2: investigate the cases in which they can be applied (a $⅋$ can always move below an independent connective; independent tensors can be swapped; a $\otimes$ can be moved below a $⅋$ as soon as it mapped to a splitting node). Step 3: if $\mathtt{ps}(\pi)$ has a terminal $⅋$ then $\pi \sim \pi'$ whose last rule is a $⅋$. Step 4: if $\mathtt{ps}(\pi)$ has a splitting, terminal $\otimes$ then $\pi \sim \pi'$ whose last rule is a $\otimes$. Step 5: prove, by induction on the number of rules that if $\mathtt{ps}(\pi) = \mathtt{ps}(\pi')$ then $\pi \sim \pi'$.
All this must wait until after we have discussed splitting nodes. — Lionel

In a cut-free sequent calculus proof or typed proof structure, the formulas used in the axiom rules or nodes are occurrences of sub-formulas of the conclusions of the proof or the typed proof structure. Two proofs are mapped to the same typed proof structure if and only if the pairing of such occurrences of formulas given by *ax* rules are the same in the two proofs.

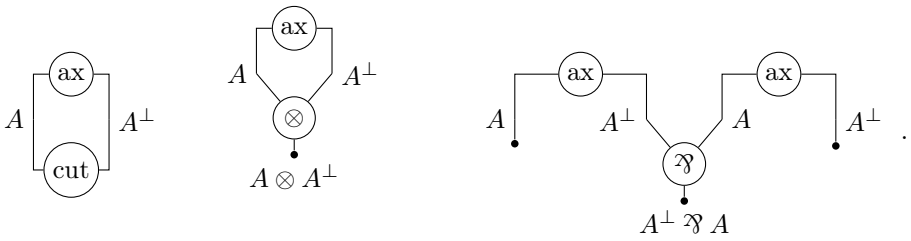
Ref? — Lionel

extend with cuts — Olivier

4.2.2 Proof nets

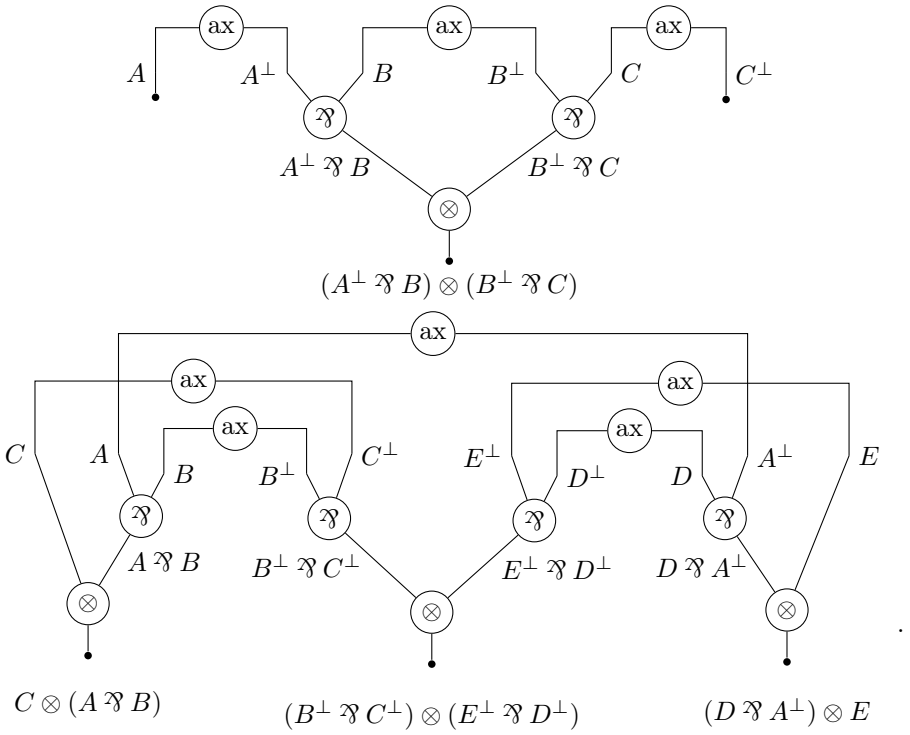
Not all typed proof structures represent (or are the translation of) proofs in the sequent calculus MLL_v.

Exercise 4.2.9. Prove that there is no proof tree whose translation yields one of the following three proof structures:



Determining whether a proof structure is indeed the translation of a proof tree is in general not so obvious.

Example 4.2.10. Here are a two other examples of typed proof structures which do not correspond to any proof of MLL_v (this will be proved in Exercise 4.2.14 below):



This leads to the study of *correctness criteria* to try to delineate a sub-set of "valid" proof structures which belong to the image of the translation `ps`.

4.2.2.1 Switching graphs

Given a proof structure $\mathcal{S}$, let $\mathcal{N}_{⅋}(\mathcal{S})$ be the set of its ⅋-nodes. A **switching** of $\mathcal{S}$ is a function φ defined on $\mathcal{N}_{⅋}(\mathcal{S})$ and such that, for each ⅋-node P, $\varphi(P)$ is one of its premises. The **switching graph** $\mathcal{S}^{\varphi}$ associated with φ is the graph obtained from $\mathcal{G}(\mathcal{S})$ by keeping only the premise $\varphi(P)$ for each ⅋-node P: formally, we modify the target of the other premise into a new node $P^{\bullet}$, as in Fig. 4.2 — where the new node $P^{\bullet}$ is depicted as a $\bullet$-node.

More explicitly:

- the nodes of $\mathcal{S}^{\varphi}$ are those of $\mathcal{S}$, plus one node $P^{\bullet}$ for each $P \in \mathcal{N}_{⅋}(\mathcal{S})$;
- the set of arrows of $\mathcal{S}^{\varphi}$ is the same as that of $\mathcal{S}$;
- the source function $\mathtt{s}$ in $\mathcal{S}^{\varphi}$ is the same as in $\mathcal{S}$;
- the target function $\mathtt{t}$ in $\mathcal{S}^{\varphi}$ is the same as in $\mathcal{S}$, except that each premise a of a ⅋-node P with $a \neq \varphi(P)$ is now mapped to $P^{\bullet}$.

Observe that $\mathcal{S}^{\varphi}$ is not the graph of a proof structure in general, because its ⅋-nodes have only one premise. Moreover, no typing information is involved

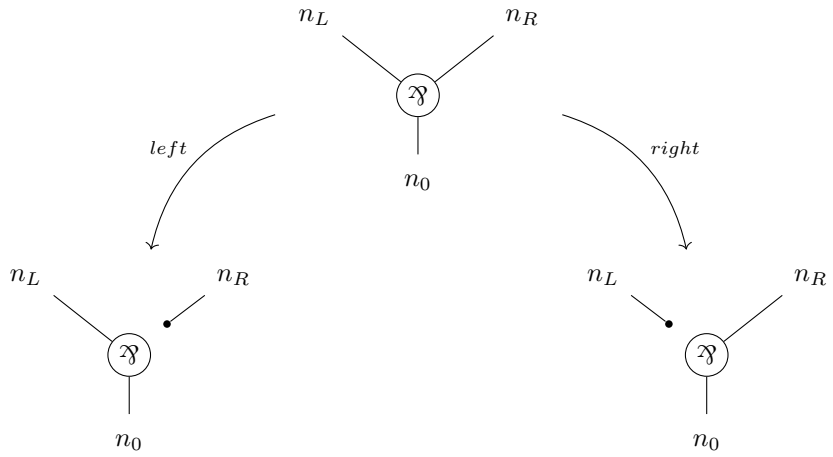


Figure 4.2: Switching a $\parr$-node.

to define switching graphs, which are in fact defined solely from the underlying unordered proof structure. A proof structure with p $\parr$-nodes induces 2^p switchings.

Definition 4.2.11 (Proof nets)**.** A **switching cycle** in a proof structure is a cycle in some of its switching graphs. A proof structure is **acyclic** if it has no switching cycle, *i.e.* if its switching graphs do not contain any cycle, An acyclic proof structure is called a **proof net**.

Note that the empty proof structure has exactly one switching, which is the empty function, and the associated switching graph is the empty graph, which is acyclic: the empty proof structure is thus a proof net. Moreover, a sum of proof structures $\mathcal{S}_1 + \mathcal{S}_2$ is a proof net iff each $\mathcal{S}_i$ is a proof net: indeed, a switching graph of $\mathcal{S}_1 + \mathcal{S}_2$ is the sum of a switching graph of $\mathcal{S}_1$ and a switching graph of $\mathcal{S}_2$.

Note that every descent path can be considered as a path inside some switching graph:

Lemma 4.2.12. *For every descent path γ of $\mathcal{S}$, there exists a switching φ such that γ is also a directed path of $\mathcal{S}^\varphi$.*

Proof. Since each descent path is a simple path, we can chose φ such that each premise of a $\parr$-node crossed by γ is in the image of φ. Then each edge of γ has the same source and target in $\mathcal{S}^\varphi$ as in $\mathcal{G}(\mathcal{S})$. Since γ is a directed path in $\mathcal{G}(\mathcal{S})$, it is also a directed path in $\mathcal{S}^\varphi$. □

In other words, every descent path is a *switching path*: the latter notion will be formally defined and thoroughly studied in Section 4.2.4.

Lemma 4.2.13 (Connected Components)**.** *All the switching graphs of a proof net have the same number of connected components.*

Proof. Let $\mathcal{S}$ be a proof net. If n is the number of nodes of $\mathcal{S}$, p its number of $\parr$ nodes and a its number of arrows, any switching graph of $\mathcal{S}$ is acyclic and has $n+p$ nodes and a arrows. By Lemma A.2.3, any such acyclic graph has $n+p-a$ connected components. □

Given a proof net $\mathcal{R}$, we write $\#_{cc}(\mathcal{R})$ for the number of connected components of its switching graphs.

A proof net $\mathcal{R}$ is *connected* if all its switching graphs have exactly one connected component: $\#_{cc}(\mathcal{R}) = 1$. Thanks to the previous lemma, this is equivalent to checking that one switching graph is connected.

Exercise 4.2.14. Inspect all possible switchings of the proof structures of Exercise 4.2.9 and Example 4.2.10. Which of these structures are proof nets? Show that none of them is a connected proof net.

Notice that since the unique switching graph of the empty proof structure has no connected components, the empty proof structure is a proof net which *is not* connected. Similarly, a sum $\mathcal{S}_1 + \mathcal{S}_2$ of structures is a connected proof net iff one is a connected proof net and the other is empty.

4.2.2.2 Soundness

We have thus introduced two **correctness criteria**: first the acyclicity of switching graphs, characterizing proof nets; then the stronger requirement that switching graphs are moreover connected, yielding connected proof nets. In the following, we will establish that the translations of proof trees are exactly the typed connected proof nets (Corollary 4.2.40). The weaker notion of proof net is nonetheless relevant: we will see that acyclicity is sufficient to expose interesting structures and properties when we study the sequentialization process in Section 4.2.4; and beyond a mere technical artifact, we will see that dropping the connectedness requirement yields a major simplification of the theory when we consider units (Section 4.3) or exponential modalities (Section 4.4).

We first establish that our correctness criteria are sound: the translation of a proof tree is correct.

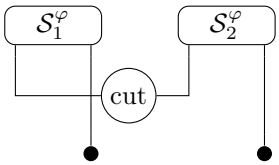
Proposition 4.2.15 (Soundness of Correctness)**.** *The translation* $\mathtt{ps}(\pi)$ *of a sequent calculus proof* π *of* MLL_v *is a typed connected proof net.*

Proof. By definition, the translation $\mathtt{ps}(\pi)$ of a sequent calculus proof π of MLL_v is a typed proof structure.

We prove that $\mathtt{ps}(\pi)$ is a proof net by induction on the structure of the MLL_v proof π. Let $\mathcal{S}$ be the proof structure associated with π, and we also need to consider two sub-proofs π_1 and π_2 of π with associated proof structures $\mathcal{S}_1$ and $\mathcal{S}_2$.

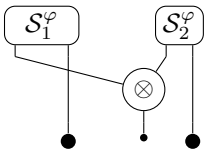
- The proof structure below has a unique switching graph which has no cycle.

- If π is obtained from π_1 and π_2 with a (cut) rule, every switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ is obtained by connecting through a *cut*-node a switching graph $\mathcal{S}_1^\varphi$ of $\mathcal{S}_1$ and a switching graph $\mathcal{S}_2^\varphi$ of $\mathcal{S}_2$.



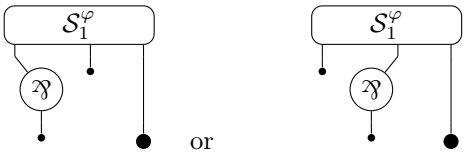
  We can deduce that no switching graph of $\mathcal{S}$ contains a cycle.

- If π is obtained from π_1 and π_2 with a $(\otimes)$ rule, any switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ is obtained by connecting through a $\otimes$-node a switching graph $\mathcal{S}_1^\varphi$ of $\mathcal{S}_1$ and a switching graph $\mathcal{S}_1^\varphi$ of $\mathcal{S}_2$.



  We can deduce that no switching graph of $\mathcal{S}$ contains a cycle.

- If π is obtained from π_1 with a $(\parr)$ rule, any switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ is obtained by putting a $\parr$-node connected to a $\bullet$-node instead of a $\bullet$-node in some switching graph $\mathcal{S}_1^\varphi$ of $\mathcal{S}_1$.



  We can deduce that no switching graph of $\mathcal{S}$ contains a cycle.

- If π is obtained from π_1 with an (ex) rule, the underlying unordered proof structure is the same.

It remains only to prove that all the switching graphs of $\mathtt{ps}(\pi)$ are connected. One can easily check, by induction on π, that $n+p-a=1$, where n (resp. p) is the number of nodes (resp. $\parr$ nodes) of $\mathtt{ps}(\pi)$, and a is the number of arrows of $\mathtt{ps}(\pi)$. Similarly to the proof of Lemma 4.2.13, this entails that every switching graph of $\mathtt{ps}(\pi)$ has exactly one connected component. □

We will establish the converse of this property in Section 4.2.4: each typed connected proof net is the translation of a proof. Before that, however, we first establish that proof nets enjoy a cut elimination procedure.

4.2.3 Cut Elimination

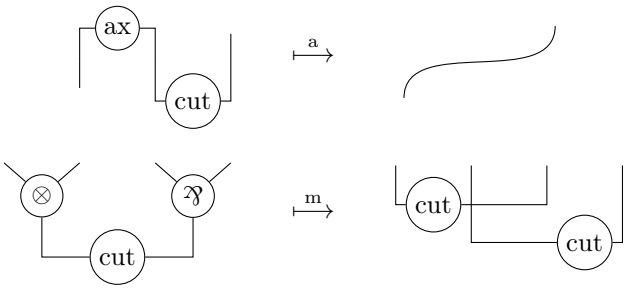
If we propose proof nets as an alternative to sequent calculus to study proofs in (multiplicative) linear logic, we need to be able to deal with cut elimination in this new syntax without referring to the sequent calculus.

Cut elimination in proof nets is defined as a graph rewriting procedure, which acts through local transformations of the proof net.

We first define the transformation on proof structures, but we will restrict immediately after to the case of proof nets.

4.2.3.1 Reductions Steps

We consider two reductions steps:



specifying local transformations of proof structures.

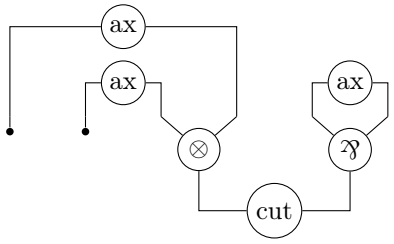
Applying a reduction step in a proof structure $\mathcal{S}$ amounts to:

- selecting a set of nodes and arrows in $\mathcal{G}(\mathcal{S})$, matching the shape of the left hand side of a reduction rule, and obeying the labelling constraint on nodes (the **redex**);
- replacing these nodes and arrows with the right hand side (its **reduct**);
- inferring the rest of the structure (the labelling of nodes, the order of premises, the interface of the structure) from that of $\mathcal{S}$.

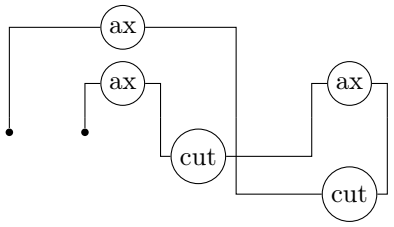
Example 4.2.16.

Utiliser des couleurs pour mettre en évidence les redex. — Lionel

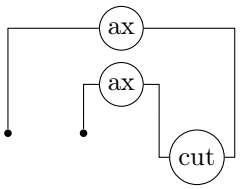
Consider the proof net $\mathcal{R}$:



containing a single $\overset{\text{m}}{\longmapsto}$ redex. Reducing this redex, we obtain



which contains *four* $\overset{\text{a}}{\longmapsto}$ redexes: one for each premise of a *cut*-node that is also a conclusion of an *ax*-node. Reducing the redex involving the right premise of the rightmost *cut*-node, we obtain

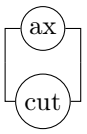


which again reduces to

by firing the redex associated with the left premise of the cut.

As intuitive as it might seem, the definition of cut elimination does require some care, as shown by the next two examples.

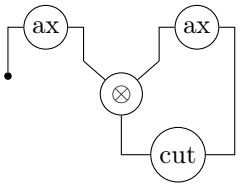
Example 4.2.17. Consider the proof structure:

$\mathcal{S}_{loop} =$ 

.

This seems to contain a redex, but it is not clear what the result of the reduction should be.

Example 4.2.18. Eliminating the only cut in the proof structure $\mathcal{S}_{fix}$



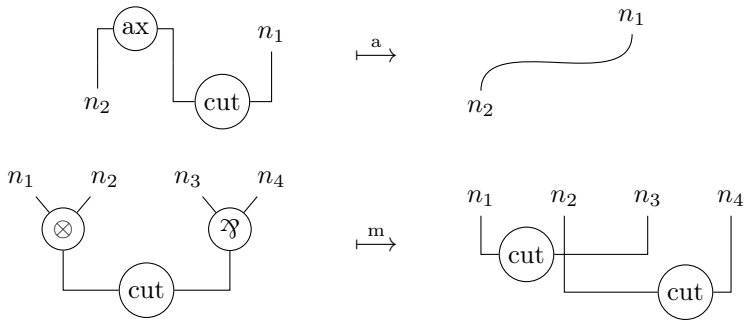
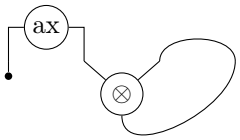


Figure 4.3: Cut elimination steps in multiplicative proof structures

yields



which is not a proof structure, because the underlying directed graph is not acyclic.

To give a precise definition of cut elimination, we need to make explicit the way a redex is matched in a proof structure. Note that some sources and targets of arrows in the depiction of reduction steps were left unspecified: these form the **interface** of the reduction step, and the drawing implicitly designates a one-to-one correspondence between the **interface nodes** of the left hand side and those of the right hand side. In Fig. 4.3, we have reproduced the two reduction rules, giving names to the interface nodes, to make this correspondence explicit. We will call **active nodes** the nodes that are actually depicted in a reduction step. By contrast with active nodes, the interface nodes are left unchanged by the reduction step, and are not part of the redex nor of the reduct: they rather serve as placeholders for the non-active targets and sources of the arrows in the redex or the reduct.

In particular, to make sense of the above procedure describing the application of a reduction step, we implicitly require that the interface nodes of a redex are distinct from the active nodes: with this requirement, the proof structure $\mathcal{S}_{loop}$ of Example 4.2.17 has no redex. Note that, for the multiplicative reduction step $\overset{\mathrm{m}}{\longmapsto}$, this is automatically satisfied, because the interface nodes occur as premises of active nodes only.

Finally, to ensure that eliminating a cut in a proof structure does yield a proof structure, we will restrict the application of the cut elimination rule $\overset{\mathrm{a}}{\longmapsto}$. More precisely, we will rule out situations such as that of Example 4.2.18, by forbidding the application of rule $\overset{\mathrm{a}}{\longmapsto}$ inside a proof structure $\mathcal{S}$ when there is

a descent path from n_2 to n_1 in $\mathcal{G}(\mathcal{S})$ (where n_1 and n_2 are the interface nodes, following the notations of Fig. 4.3). With this restriction, it is clear that the application of any reduction step in a proof structure preserves the directed acyclicity of the underlying graph, and the rest of the constraints ensure that the result is indeed a proof structure.

On the other hand, we do not make the assumption that the interface nodes n_1 to n_4 in the $\overset{\text{m}}{\longmapsto}$ step of Fig. 4.3 are pairwise distinct: even in the translation of a proof tree, the premises of a $\parr$-node might very well be conclusions of the same *ax*-node, and we must be able to eliminate a cut involving such a $\parr$-node. This was in particular the case in the first reduction step of Example 4.2.16.

We are now ready to state the definition of cut elimination in proof structures precisely:

Definition 4.2.19. The **reduction of multiplicative proof structures** $\overset{\text{am}}{\longrightarrow}$ is the union of:

- the **multiplicative reduction** $\overset{\text{m}}{\longrightarrow}$ induced on proof structures by the application of the reduction step $\overset{\text{m}}{\longmapsto}$ of Fig. 4.3;
- the **axiom reduction** $\overset{\text{a}}{\longrightarrow}$ induced on proof structures by the application of reduction step $\overset{\text{a}}{\longmapsto}$ of Fig. 4.3, under the condition that there is no descent path from the n_2 to n_1.[2]

A **redex in a proof structure** $\mathcal{S}$ is a set of nodes and arrows in $\mathcal{G}(\mathcal{S})$ matching the left hand side of a rule in Fig. 4.3, such that the corresponding reduction step can be applied.

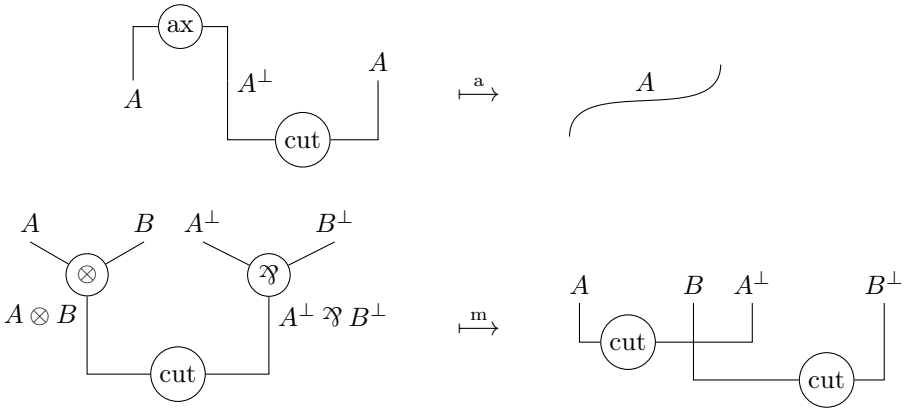
Note that any pattern matching the left hand side of $\overset{\text{m}}{\longmapsto}$ in a proof structure $\mathcal{S}$ is always a redex, while in the case of $\overset{\text{a}}{\longmapsto}$ it is considered a redex in $\mathcal{S}$ only if there is no descent path from n_2 to n_1 in $\mathcal{G}(\mathcal{S})$. Depending on the constraints we put on proof structures, we thus obtain the following situations:

1. In the general case, we can find proof structures, (such as those of Examples 4.2.17 and 4.2.18) in which a *cut*-node might have a premise which is a conclusion of an *ax*-node, but does not belong to any redex. The next point shows that such proof structures cannot be proof nets.
2. If $\mathcal{R}$ is a proof net, then each *cut*-node having as premise a conclusion of an *ax*-node belongs to a redex involving those two nodes. Indeed, every descent path belongs to a switching graph (Lemma 4.2.12): with the notations of Fig. 4.3, if there was a descent path from n_2 to n_1 in $\mathcal{G}(\mathcal{R})$, we would obtain a cycle in the associated switching graph *via* three edges of the redex.
3. If $\mathcal{R}$ is a proof net that cannot be typed, one might still have a cut which does not belong to any redex (e.g. a cut between two $\otimes$-nodes).

[2] Recall that, as explained above, we always assume that interface nodes are distinct from active nodes: here, n_1 is not the depicted *ax*-node, and n_2 is not the depicted *cut*-node.

4. If $\mathcal{R}$ is a typeable proof net, then every cut belongs to a redex. Indeed, given a *cut*-node c: either one premise node of c is an *ax*-node, and Item 2 ensures this forms a redex; or both premise nodes of c are $\otimes$- or $\parr$-nodes. In the latter case, due to the typing constraints, one must be a $\otimes$-node and the other a $\parr$-node, which yields a redex.

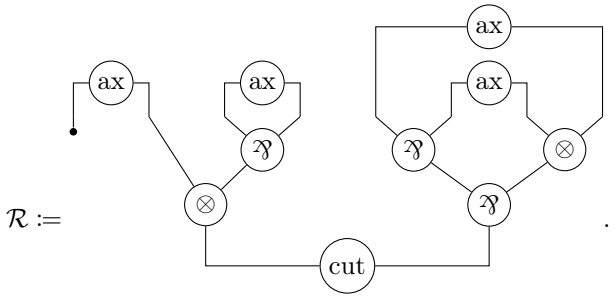
Moreover notice that applying reduction steps inside a typed structure preserves typing. Since both typing and cut elimination are defined locally, it suffices to observe how a typing is transformed by a reduction step:



which ensures that the typing constraints imposed at the level of interface nodes are the same in the redex and in the reduct — the rest of the typing is left unchanged.

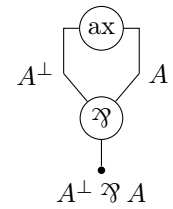
As a consequence, normal forms for the reduction of typed proof nets are exactly cut-free typed proof nets.

Exercise 4.2.20. Consider the proof net



Find a typing of $\mathcal{R}$ and check that the conclusion sequent is of the form $\vdash A^\perp \parr A$. This typing being fixed, find a sequence of reduction steps from $\mathcal{R}$ to a *cut*-free

typed net. Check that the result is isomorphic to

$\mathcal{R}_{id} \coloneqq$ 

.

It is in fact obvious that all possible reduction paths from the proof net $\mathcal{R}$ of Example 4.2.16 yield the same *cut*-free result (up to isomorphism of proof structures). We will later establish (see Section 4.2.3.3) that this is a general fact: cut elimination is *convergent*. Before that, we show that proof nets are stable under cut elimination.

4.2.3.2 Preservation of Correctness

Lemma 4.2.21 (Preservation of Acyclicity)**.** *If $\mathcal{R}$ is a proof net and $\mathcal{R} \xrightarrow{\text{am}} \mathcal{R}'$ then $\mathcal{R}'$ is a proof net.*

Proof. We consider the two steps:

- Through an a step, a switching graph of the reduct can be turned into a switching graph of the redex by replacing an edge crossing the new arrow with a path of length 3 going through the *ax* node and through the *cut* node. Then one of these two switching graphs is acyclic if and only if the other one is.
- Through an m step, a switching graph $\mathcal{S}^\varphi$ of the reduct gives rise to two switching graphs $\mathcal{S}^{\varphi_1}$ and $\mathcal{S}^{\varphi_2}$ in the redex depending on the choice of a premise a_1 or a_2 of the $\parr$ node which disappears through the reduction. Assume there is a cycle in $\mathcal{S}^\varphi$. It must go through at least one of the cuts otherwise it is a cycle in $\mathcal{S}^{\varphi_1}$ and $\mathcal{S}^{\varphi_2}$.

  If it goes only through the cut with premise a_i in the reduct, the premises of this cut are connected in $\mathcal{S}^\varphi$ (without using the cut) and then we have a cycle in $\mathcal{S}^{\varphi_i}$, hence a contradiction.

  If it uses both cuts: either the premises of the $\otimes$-node are connected in $\mathcal{S}^\varphi$ (without using those cuts) and we have a cycle in both $\mathcal{S}^{\varphi_1}$ and $\mathcal{S}^{\varphi_2}$; or a_1 is connected to a premise of the $\otimes$-node, and we have a cycle in $\mathcal{S}^{\varphi_1}$.

Draw pictures — Lionel

□

Remember that, thanks to Lemma 4.2.13, all the switching graphs of a proof net have the same number of connected components.

Lemma 4.2.22 (Preservation of Connected Components)**.** *If $\mathcal{R}$ is a proof net and $\mathcal{R} \xrightarrow{\text{am}} \mathcal{R}'$ then the number of connected components of the switching graphs of $\mathcal{R}'$ is the same as for the switching graphs of $\mathcal{R}$.*

Proof. The switching graphs are acyclic in both $\mathcal{R}$ and $\mathcal{R}'$ (see Lemma 4.2.21). We can thus use Lemma A.2.3. We consider the two reduction steps. In each case, in every switching graph we loose two nodes and two arrows thus the number of connected components is not modified. □

In particular a reduct of a connected proof net is a connected proof net.

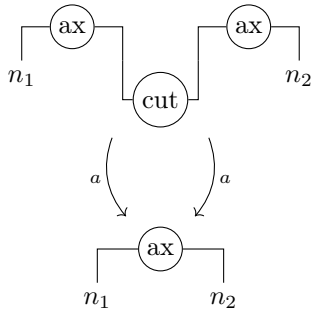
4.2.3.3 Normalization

If we consider cut elimination as a computational process on proof nets, the two key properties we want to prove about it are termination and uniqueness of the result. If the existence of a terminating reduction strategy (weak normalization) allowing to reach a cut-free proof net from any typed proof net is enough from the point of view of logical consistency, it is more satisfactory from a computational point of view to prove that any reduction will eventually terminate (strong normalization). It turns out that, in the multiplicative case, this does not require any typing condition nor any correctness condition.

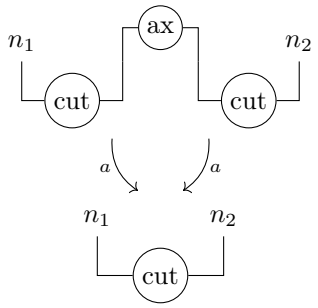
Lemma 4.2.23 (Sub-Confluence)**.** *The reduction of proof structures* $\xrightarrow{\text{am}}$ *is sub-confluent*[3]*: more precisely, if* $\mathcal{R} \xrightarrow{\text{am}} \mathcal{R}_1$ *and* $\mathcal{R} \xrightarrow{\text{am}} \mathcal{R}_2$ *then* $\mathcal{R}_1 = \mathcal{R}_2$ *or there exists* $\mathcal{R}'$ *such that* $\mathcal{R}_1 \xrightarrow{\text{am}} \mathcal{R}'$ *and* $\mathcal{R}_2 \xrightarrow{\text{am}} \mathcal{R}'$*.*

Proof. There are two kinds of critical pairs:

- a/a (shared *cut*)

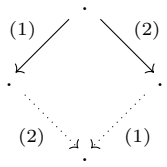


- a/a (shared *ax*)



[3]The vocabulary and main properties of reduction systems are recalled in Appendix B.

In all the other situations, two different reductions from a given proof net commute:



since they cannot overlap. □

Proposition 4.2.24 (Convergence)**.** *The reduction of proof structures is* convergent*.*

Proof. Confluence is obtained by Proposition B.2.1 and Lemma 4.2.23. Moreover, the number of nodes is reduced in each reduction step. □

Since each cut of a typed proof net is involved in at least one redex, we obtain:

Corollary 4.2.25 (Normalization of typed proof nets)**.** *The reduction of typed proof nets is* convergent*, and the unique normal form of a typed proof net is cut-free.*

4.2.4 Sequentialization

We want to associate an MLL_v proof with each typed connected multiplicative proof net. This is called the sequentialization process, for it requires to turn the graph structure of proof nets into the more sequential structure of proofs trees. A proof tree π such that $\mathtt{ps}(\pi) = \mathcal{R}$ is called a **sequentialization** of $\mathcal{R}$.

4.2.4.1 Overview

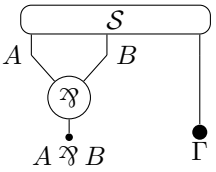
There are various approaches to sequentialization. We provide a survey later in this chapter. Here we chose to follow what seems like the most natural strategy: we focus on terminal nodes and investigate the conditions under which the translation process described in Section 4.2.1.4 can be reversed.

More precisely, fixing a typed connected proof net $\mathcal{R}$, we can consider the following cases:

- If $\mathcal{R}$ has a terminal ax-node, then it must be reduced to this node and its conclusions (otherwise its switching graphs have more than one connected component), and it is thus the translation of an (ax) rule.

- If $\mathcal{R}$ has a terminal $\parr$-node, then we can assume (up to reordering the conclusions, which amounts to apply (ex) rules at the bottom of the sequentialization under construction) that $\mathcal{R}$ has the shape obtained in the

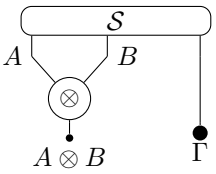
translation of a ($\parr$) rule, namely:

$\mathcal{R} =$ 

.

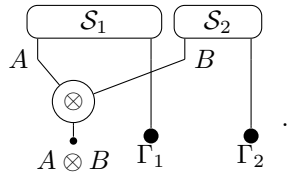
The structure $\mathcal{S}$ obtained from $\mathcal{R}$ by removing the $\parr$-node (also removing its conclusion, and adding two new $\bullet$-nodes as conclusions of the arrows typed A and B) is necessarily a typed connected proof net: by sequentializing $\mathcal{S}$ inductively, we obtain a proof tree π_1 with conclusion $\vdash A, B, \Gamma$, and we construct a sequentialization of $\mathcal{R}$ by applying a ($\parr$) rule to π_1.

- It remains only to consider the case where all the terminal nodes of $\mathcal{R}$ are $\otimes$-nodes or *cut*-nodes. For the purpose of sequentialization, we can in fact assume that $\mathcal{R}$ is cut-free: indeed, we can replace a *cut*-node with premises of type A and $A^\perp$ with a $\otimes$-node with conclusion $A \otimes A^\perp$, without changing the required properties of switching graphs (we make this argument formal below: see Lemma 4.2.27) nor the typing constraints. We are thus left with terminal $\otimes$-nodes only. Again, if we fix a terminal $\otimes$-node, up to reordering the conclusions, we can assume $\mathcal{R}$ has the shape:

$\mathcal{R} =$ 

but note that this is not the shape of a translation of a ($\otimes$) rule!

A crucial step in the sequentialization process is thus to find a terminal $\otimes$-node T that is moreover **splitting**: it is not part of a cycle in $\mathcal{G}(\mathcal{R})$. Indeed, assuming the $\otimes$-node T in the previous depiction of $\mathcal{R}$ is splitting, since there is no path between its premises other than through T itself, we can split $\mathcal{S}$ into $\mathcal{S}_1$ with conclusion $\vdash A, \Gamma_1$ and $\mathcal{S}_2$ with conclusion $\vdash B, \Gamma_2$ (further assuming, up to an additional reordering of conclusions, that $\Gamma = \Gamma_1, \Gamma_2$) to obtain:

$\mathcal{R} =$ 

.

Then we can inductively sequentialize $\mathcal{S}_1$ and $\mathcal{S}_2$ separately, and apply a ($\otimes$) rule to obtain a sequentialization of $\mathcal{R}$.

Our objective is thus to show that, in a cut-free connected proof net which is not reduced to an *ax*-node, there must exist a terminal $\parr$-node, or a splitting terminal $\otimes$-node: this is the crux of the sequentialization process.

In order to help the reuse of some of the results in later sections, we consider a simple generalization of proof structures where *ax*-nodes are replaced with *hyp*-nodes:

- each node labelled *hyp* has an arbitrary number of conclusions (at least one) and no premise;
- in a typing for a proof structure with *hyp*-nodes, we require that for each *hyp*-node with conclusions $A_1, \ldots, A_n$ (in arbitrary order), the sequent $\vdash A_1, \ldots, A_n$ is derivable in MLL_v.

The original *ax*-nodes are clearly a particular case of these new *hyp*-nodes since $\vdash A, A^\perp$ is provable for any A in MLL_v by means of an (ax) rule.

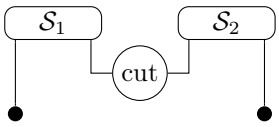
Moreover, typing plays no rôle in finding an appropriate terminal node, as we only rely on geometrical properties of switching graphs. To reduce the "noise" associated with the management of types, we introduce an untyped version of sequentiality in the next section. Again, this will also help us to reuse some results in later sections, where more general forms of typing are considered.

4.2.4.2 Connected sequential structures

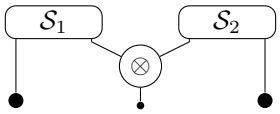
We say that an unordered proof structure $\mathcal{S}$ is **connected sequential** if one of the following holds, assuming inductively that $\mathcal{S}_1$ and $\mathcal{S}_2$ are connected sequential unordered proof structures:

(S1) $\mathcal{S}$ is reduced to an *hyp*-node with its conclusion arrows and respective $\bullet$-nodes;

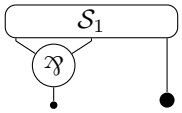
(S2) $\mathcal{S}$ is obtained from $\mathcal{S}_1 + \mathcal{S}_2$ by removing one $\bullet$-node in each of $\mathcal{S}_1$ and $\mathcal{S}_2$, with premises a_1 in $\mathcal{S}_1$ and a_2 in $\mathcal{S}_2$ respectively, and by introducing a new *cut*-node with premises a_1 and a_2;



(S3) $\mathcal{S}$ is obtained from $\mathcal{S}_1 + \mathcal{S}_2$ by removing one $\bullet$-node in each of $\mathcal{S}_1$ and $\mathcal{S}_2$, with premises a_1 in $\mathcal{S}_1$ and a_2 in $\mathcal{S}_2$, and by introducing a new terminal $\otimes$-node with premises a_1 and a_2, together with its conclusion arrow and node;



(S4) $\mathcal{S}$ is obtained from $\mathcal{S}_1$ by removing two $\bullet$-nodes in $\mathcal{S}_1$, with premises a_1 and a_2, and by introducing a new terminal $\parr$-node with premises a_1 and a_2 together with its conclusion arrow and node.



And we say that a proof structure $\mathcal{S}$ is **connected sequential** if the underlying unordered proof structure is.

Note that we took advantage of the irrelevance of the ordering of conclusions to illustrate the cases (S2) to (S4) without crossing edges.

Lemma 4.2.26 (Untyped correctness)**.** *Any connected sequential proof structure is a connected proof net.*

Proof. By induction on the definition of connected sequential proof structures: cases (S1) to (S4) are treated exactly as in the proof of Proposition 4.2.15. □

In the following, we will establish the converse: every connected proof net is connected sequential. We will then deduce the sequentialization theorem (Theorem 4.2.39) from the easy observation that a typed proof structure is the translation of a proof as soon as it is connected sequential.

To establish the correspondence between connected sequential structures and connected proof nets, it will be sufficient to consider *cut*-free structures. Indeed, for any proof structure $\mathcal{S}$, we write $\mathcal{S}[\otimes/cut]$ for the proof structure obtained by replacing each *cut*-node with a $\otimes$-node — with conclusion pointing to a fresh $\bullet$-node. We obtain:

Lemma 4.2.27. *For any proof structure $\mathcal{S}$:*

- *$\mathcal{S}$ is connected sequential iff $\mathcal{S}[\otimes/cut]$ is;*
- *$\mathcal{S}$ is a connected proof net iff $\mathcal{S}[\otimes/cut]$ is.*

Proof. The first item is direct by induction on the definition of connected sequential structures. For the second item, it is sufficient to observe that:

- the $\parr$-nodes of $\mathcal{S}[\otimes/cut]$ are those of $\mathcal{S}$;
- given a switching φ of $\mathcal{S}$ (equivalently, of $\mathcal{S}[\otimes/cut]$), any path of $\mathcal{S}^\varphi$ is also a path of $\mathcal{S}[\otimes/cut]^\varphi$, with the same endpoints, and any simple path of $\mathcal{S}[\otimes/cut]^\varphi$ with endpoints in $\mathcal{S}^\varphi$ is also a path of $\mathcal{S}^\varphi$.

□

4.2.4.3 Switching paths and bridges

Given a proof structure $\mathcal{S}$, recall that any switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ has an additional node $P^\bullet$ for each $\parr$-node P, and the same edges as $\mathcal{S}$, except that the target of the premise different from $\varphi(P)$ of each $\parr$-node P is changed to $P^\bullet$.

We call **switching path** of $\mathcal{S}$ any path γ in $\mathcal{G}(\mathcal{S})$ that never crosses both premises of a $\parr$-node: equivalently, γ is a switching path if there exists a switching φ such that, for each premise a of $\parr$-node P, $\varphi(P) = a$ as soon as γ crosses a. In particular γ is also a path in the *switching graph* $\mathcal{S}^\varphi$. Conversely, a path in $\mathcal{S}^\varphi$ is a switching path of $\mathcal{S}$, as soon as none of the new nodes $P^\bullet$ occur in γ. In particular a *switching cycle* is the same thing as a cyclic switching path:

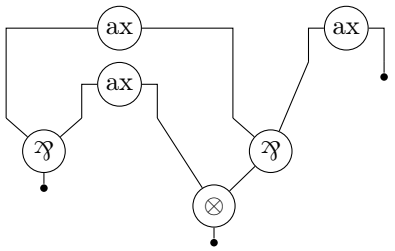
Lemma 4.2.28. *A cycle in a proof structure is a switching cycle iff it is a switching path.*

Proof. A switching cycle in a structure $\mathcal{S}$ is a cycle in some switching graph $\varphi^\mathcal{S}$: since it is a simple path, it cannot cross the premise of any node $P^\bullet$ in $\varphi^\mathcal{S}$, hence it is also a path in $\mathcal{G}(\mathcal{S})$, which crosses at most one premise of each $\parr$-node.

Conversely, given a cyclic switching path γ, we have already seen that γ is also a path in some $\varphi^\mathcal{S}$. □

Given a path γ, a **bridge** of γ is any subpath $a_1^+ a_2^-$ of γ, such that a_1 and a_2 are both premises of some $\parr$-node P, which we call the **pier** of the bridge. Of course, a switching path has no bridge, but the converse does not hold in general.

Example 4.2.29. In the structure



the only $\otimes$-node is the source of exactly one cycle, up to the choice of an orientation. This cycle has exactly one bridge, whose pier is the terminal $\parr$-node. By contrast, the cycle whose source is the terminal $\parr$-node has no bridge — although it uses the same edges as the previous one, up to a circular permutation.

We generalize the situation observed in the previous example as follows: a **proper cycle** of a $\parr$-node P is any bridge-free cycle of the shape $a_1^- \gamma' a_2^+$ where a_1 and a_2 are both premises of P.

Lemma 4.2.30 (Bridge-free cycles)**.** *Any bridge-free cycle in a proof structure is either a switching cycle, or a proper cycle of some $\parr$-node.*

Proof. Assume γ is a bridge-free cycle in a proof net, that is not a switching cycle: γ crosses both premises of some $⅋$-node. Given such a $⅋$-node P, write a_1 and a_2 for its premises, and then write $\gamma = \gamma_1 e_1 \gamma' e_2 \gamma_2$, with $e_1 = a_1^+$ or a_1^- and $e_2 = a_2^+$ or a_2^-. We can moreover chose P so that γ' is of minimal length: γ' is then a switching path.

Now observe that, in each of the following cases, we obtain a contradiction:

- if $e_1 = a_1^+$ and $e_2 = a_2^-$, either γ' crosses the conclusion of P twice (but γ is simple), or it is empty and we obtain a bridge of γ (but γ is bridge-free);
- if $e_1 = a_1^+$ and $e_2 = a_2^+$, then the first edge of γ' must be a_0^+ (because γ' is not closed, it cannot be empty, and must start with an outgoing edge of $\mathsf{t}(a_1) = P$, and moreover γ is simple so γ' cannot cross a_1 nor a_2); hence γ_2 is empty ($\mathsf{s}(\gamma_2) = P$ but γ is simple so γ_2 does not cross any of the a_i's) and then γ_1 must also be empty ($\mathsf{s}(\gamma_1) = P$ but, again, γ is simple), hence $\mathsf{s}(\gamma) = \mathsf{s}(a_1) \neq P = \mathsf{t}(\gamma)$ (but γ is closed);
- if $e_1 = a_1^-$ and $e_2 = a_2^-$, we reason symmetrically to the previous case.

Hence we must have: $\gamma = \gamma_1 a_1^- \gamma' a_2^+ \gamma_2$. Then $\mathsf{t}(\gamma_1) = P = \mathsf{s}(\gamma_2)$ so one of γ_1 and γ_2 must be empty (again, because γ is simple and already crosses a_1 and a_2). In particular, $P = \mathsf{s}(\gamma)$: this forces both γ_1 and γ_2 to be closed simple paths of source P and since they cannot cross a_1 nor a_2, both must be empty. □

We say a path γ is **weak** if its first edge is a^- with a a premise of a $⅋$-node. We say γ is **strong** if it is neither empty nor weak. We say γ is **coweak** (resp. **costrong**) if $\overline{\gamma}$ is weak (resp. strong). Lemma 4.2.30 can be reformulated equivalently as follows:

Corollary 4.2.31 (Strong bridge-free cycles)**.** *A proof structure is a proof net iff it has no strong bridge-free cycle.*

Proof. We show that in any proof structure, a cycle is a switching path iff it is bridge-free, and strong or costrong.

Given a switching cycle γ, we have already observed that γ is bridge-free, hence so is $\overline{\gamma}$. Moreover, one of γ or $\overline{\gamma}$ must be strong, as otherwise γ is both weak and coweak, *i.e.* the source of γ is a $⅋$-node P and the first edge and last edge of γ cross the premises of P, which contradicts the fact that γ is switching.

Conversely, a bridge-free cycle that is strong or costrong cannot be a proper cycle, hence it is a switching cycle by Lemma 4.2.30. □

Moreover, a direct application of the definitions yields:

Lemma 4.2.32 (Concatenation of Strong and Bridge-Free Paths)**.** *Let γ and γ' be two composable paths:*

- *if γ is strong (resp. weak) then so is $\gamma\gamma'$;*
- *if γ' is costrong (resp. coweak) then so is $\gamma\gamma'$;*

- *if γ is costrong or γ' is strong, and both γ and γ' are bridge-free, then $\gamma\gamma'$ is bridge-free.*

In particular, if γ and γ' are both strong and bridge-free, then so is then $\gamma\gamma'$.

Definition 4.2.33 (Two relations on arrows)**.** Let a and b be arrows, and n a node, in a proof structure. We write $\gamma : a \frown n$ whenever γ is a simple, bridge-free and strong path from $\mathtt{t}(a)$ to n, not starting with the edge a^-. We then write $\gamma : a \frown b$ whenever, in addition, the last edge of γ is b^+, so that $n = \mathtt{t}(b)$. We simply write $a \frown n$ (resp. $a \frown b$) whenever such a path $\gamma : a \frown n$ (resp. $\gamma : a \frown b$) exists.

We moreover write $\gamma : a \prec b$ whenever $\gamma : a \frown b$, and there is no node n of γ such that $b \frown n$. Again, we may simply write $a \prec b$ whenever $\gamma : a \prec b$.

drawings! — Lionel

Lemma 4.2.34 (Composing $\prec$ and $\frown$)**.** *Whenever $\gamma : a \prec b$ and $\delta : b \frown n$, we have $\gamma\delta : a \frown n$. And if moreover $\delta : b \frown c$, then $\gamma\delta : a \frown c$.*

Proof. First observe that γ and δ are disjoint: otherwise we contradict $\gamma : a \prec b$ by considering any non-empty prefix δ' of δ (which is automatically simple, bridge-free and strong) such that $\mathtt{t}(\delta')$ occurs in γ. It follows that $\gamma\delta$ is also simple. It is moreover bridge-free and strong by Lemma 4.2.32, and its first edge is that of γ, which is not a^-: we thus have $\gamma\delta : a \frown n$.

If moreover $\delta : b \frown c$, then the last edge of $\gamma\delta$ is that of δ, which is c^+, and we have $\gamma\delta : a \frown b$. □

Lemma 4.2.35 (A strict partial order on arrows)**.** *In any proof net $\mathcal{R}$, the relation $\prec$ is a strict partial order relation on the arrows of $\mathcal{R}$.*

Proof. We first prove that $\frown$ is irreflexive: assuming $\gamma : a \frown a$, γ would be a strong and bridge-free cycle, which contradicts Corollary 4.2.31. It follows that $\prec$ is irreflexive too.

For transitivity, assume $\gamma : a \prec b$ and $\delta : b \prec c$. By the previous Lemma, we also have $\gamma\delta : a \frown c$. It remains only to show that there is no node n of $\gamma\delta$ such that $c \frown n$. Towards a contradiction, assume we have $\rho : c \frown n$ with n occurring in $\gamma\delta$. If n occurs in δ we contradict $\delta : b \prec c$ directly. Hence n occurs in γ, and we contradict $\gamma : a \prec b$ because $\delta\rho : b \frown n$ by the previous Lemma. □

Since the set of arrows of a proof net is finite, it follows that a net with at least one internal arrow has a $\prec$-maximal one (among internal arrows). And we can already establish that such a maximal internal arrow is necessarily a premise of a terminal node:

Lemma 4.2.36 (Maximal arrows are terminal)**.** *Let a be an internal arrow of a proof net, that is maximal for $\prec$ among internal arrows. Then a is terminal.*

Proof. We reason by contraposition. Assume $n \coloneqq \mathsf{t}(a)$ is not terminal. Necessarily, n has an internal conclusion arrow, that we write b. We then show that $a \prec b$. The descent path reduced to b^+ is obviously strong, simple and bridge-free, hence $b^+ : a \frown b$. Now assume towards a contradiction that we have a path $\gamma : b \frown p$ with p occurring in b^+. Since the first edge of γ does not cross b we can assume w.l.o.g. that γ does not cross b (otherwise we pick a suitable non-empty prefix). Then there are only two cases:

- $p = \mathsf{t}(b)$, hence γ is a strong, bridge-free cycle;
- $p = n$, and then, again, $b^+\gamma$ is a strong, bridge-free cycle;

and both cases contradict Corollary 4.2.31. □

In the next section we show that if the target of a maximal internal arrow is a $\otimes$-node, the latter is necessarily splitting.

4.2.4.4 Bungee jumping and splitting $\otimes$-nodes

Given some node n in a proof structure, we write $\mathcal{B}(n)$ for the set of strong or costrong cycles with source n, whose number of bridges is minimal. Note that $\gamma \in \mathcal{B}(n)$ iff $\overline{\gamma} \in \mathcal{B}(n)$.

Lemma 4.2.37 (Bungee Jumping)**.** *Let n be a node in a proof net, let $\omega \in \mathcal{B}(n)$, and let a be the conclusion of κ, the pier of a bridge of ω. If γ is a simple, bridge-free and strong path with first edge a^+, then $\mathsf{t}(\gamma)$ does not occur in ω.*

Proof. Write $p \coloneqq \mathsf{t}(\gamma)$ and assume that p occurs in ω. It is sufficient to show the existence of a strong bridge-free cycle to contradict Corollary 4.2.31.

We can assume $\kappa \neq p$: otherwise we already have such a cycle. Note that ω cannot cross the first edge of γ, which is the conclusion of κ, because ω is simple and already crosses both premises of κ at the bridge. In particular, $\kappa \neq n$. We can thus assume that γ is disjoint from ω, up to taking a (non empty) prefix. Noting that ω satisfies the hypotheses iff $\overline{\omega}$ does, we can reverse ω if necessary to ensure that: in case $p = n$, then ω is strong; and otherwise the occurrence of p in ω is after the pier κ. We can thus write $\omega = \omega_1\delta\omega_2$, so that $\mathsf{s}(\delta) = \kappa$ and $\mathsf{t}(\delta) = p$.

Now consider the closed path $\pi = \omega_1\gamma\omega_2$: it is not empty since $\kappa \neq p$; it is moreover simple, as we have already ensured that ω and γ are disjoint. Hence π is a cycle, which is moreover strong or costrong: if $p = n$, we have made sure ω_1 is strong; and otherwise, ω_1 is a non empty prefix of ω and ω_2 is a non empty suffix, hence ω_1 is strong or ω_2 is costrong. By the definition of $\mathcal{B}(n)$, it follows that π has at least as many bridges as ω. Since γ is bridge-free, and π cannot have a bridge at κ (because γ is strong), there must be no bridge in δ, and there is exactly one newly formed bridge in π, with pier p: hence γ is coweak and ω_2 is weak, which leaves only the conclusion of p for the last edge of δ. By Lemma 4.2.32, $\gamma\overline{\delta}$ is a strong bridge-free cycle. □

Lemma 4.2.38 (Finding a splitting $\otimes$-node)**.** *If a is an internal arrow in a proof net, that is maximal for $\prec$ among internal arrows, and moreover the (necessarily terminal) node $n \coloneqq \mathsf{t}(a)$ is a $\otimes$-node, then the latter is splitting.*

Proof. Write a_1 and a_2 for the premises of n, so that $a = a_2$. Assume, towards a contradiction, that n is not splitting: necessarily, the set $\mathcal{B}(n)$ is not empty, and contains a cycle ω whose first edge is a_1^-. Consider the first bridge $b_1^+ b_2^-$ of ω: write κ for its pier, with premises b_1 and b_2. Then write γ for the prefix of ω with last edge b_1^+: this is strong (its first edge is a_1^-), simple (it is a subpath of a cycle), and bridge-free (by construction). We obtain $\gamma : a \frown b_1$.

Moreover, there is no path $\delta : b_1 \frown p$ with p occurring in γ (hence in ω). Indeed, the first edge of such a strong path δ cannot be b_1^- nor b_2^-, so it must be b_0^+ with b_0 the conclusion of κ: this would contradict the Bungee Jumping Lemma. We thus obtain $a \prec b_1$, contradicting the maximality of a since b_1 is internal. □

4.2.4.5 Sequentialization of connected proof nets

We are now ready to establish the announced sequentialization result.

Theorem 4.2.39 (Connected sequentialization)**.** *Any connected proof net $\mathcal{R}$ is connected sequential. If moreover $\mathcal{R}$ is typed then it is the translation of a sequent calculus proof of MLL_v.*

Proof. Suppose $\mathcal{R}$ is a connected proof net: by Lemma 4.2.27, $\mathcal{R}$ is connected sequential as soon as $\mathcal{R}[\otimes/cut]$ is and $\mathcal{R}[\otimes/cut]$ is a connected proof net as soon as $\mathcal{R}$ is a connected proof net. For the first part, it is thus sufficient to prove that if $\mathcal{R}$ is a *cut*-free connected proof net, then $\mathcal{R}$ is connected sequential. We reason by induction on the number of internal nodes of $\mathcal{R}$.

Since $\mathcal{R}$ is a connected proof net, it must be non-empty, so it has at least one internal node.

If the only internal nodes of $\mathcal{R}$ are *hyp*-nodes, then they must be terminal (there is no internal node with premises). By connectedness, $\mathcal{R}$ is reduced to one *hyp*-node and its conclusions: we conclude by (S1).

Otherwise, $\mathcal{R}$ must contain at least one internal arrow. We pick one, a_1, that is maximal for $\prec$. Write $n \coloneqq \mathsf{t}(a)$, which is either a $\parr$-node or a $\otimes$-node, and a_2 for the other premise of n. By Lemma 4.2.36, n is terminal.

If n is a $\parr$-node, then:

- we consider the proof structure $\mathcal{R}'$ obtained from $\mathcal{R}$ by replacing n and its conclusion with two fresh $\bullet$-nodes with premises a_1 and a_2;
- $\mathcal{R}'$ is also a connected proof net, which yields a connected sequential structure by induction hypothesis;
- we conclude by (S4).

If n is a $\otimes$-node, then it is splitting by Lemma 4.2.38:

- we consider the proof structure $\mathcal{R}'$ obtained from $\mathcal{R}$ by replacing n and its conclusion with two fresh $\bullet$-nodes, with premises a_1 and a_2;
- since n is splitting, $\mathcal{R}'$ is made of two connected components, $\mathcal{R}_1$ containing a_1 and $\mathcal{R}_2$ containing a_2, each being a connected proof structure;
- the induction hypothesis can be applied to $\mathcal{R}_1$ and to $\mathcal{R}_2$ to yield connected sequential structures;
- we conclude by (S3).

For the second part, assuming $\mathcal{R}$ is typed, we construct a suitable proof of MLL_v by a straightforward induction on $\mathcal{R}$ as a connected sequential structure.

We should do this proof in detail (and then maybe refer to it for the version with jumps). — Lionel

□

Corollary 4.2.40 (Characterization of the translation of MLL_v proof trees)**.** *A typed proof structure $\mathcal{R}$ is the translation of a proof tree iff it is a connected proof net.*

4.3 Multiplicative units and the *mix* rules

4.3.1 Multiplicative units and jumps

TODO: We postpone the discussion on the impossibility of having a satisfactory graphical characterization of sequential nets to the final section, where we should mention the literature on the subject. — Lionel

Until this point, multiplicative proof nets only covered the fragment without units MLL_v. To cover the propositional fragment MLL_0, it remains only to translate the two rules (**1**) and ($\bot$). A natural idea is to extend the definition of proof structures with two new node labels **1** and $\bot$, and require that each **1**-node (resp. each $\bot$-node) has no premise and exactly one conclusion, of type **1** (resp. $\bot$).

Then the translation $\mathtt{ps}$ from proof trees to proof structures is extended as follows:

- the proof
$$\frac{}{\vdash \mathbf{1}}\ (\mathbf{1})$$
is translated to the proof structure with a single **1**-node;
- the translation of
$$\dfrac{\begin{array}{c}\pi\\ \vdots\\ \Gamma\end{array}}{\vdash \Gamma, \bot}\ (\bot)$$
is $\mathtt{ps}(\pi)$ with an additional $\bot$-node.

Note that the case of $(\bot)$ generates a new connected component in the underlying graph: the proof structure associated with an MLL_0 proof is not necessarily connected. In particular, a cut between a $\bot$-node and a $\mathbf{1}$-node forms a connected component: eliminating this cut simply amounts to removing this component. We write $\xrightarrow{\mathrm{u}}$ for the reduction defined by eliminating such $\bot/\mathbf{1}$ cuts, and $\xrightarrow{\mathrm{amu}}$ for the reduction obtained as the union of $\xrightarrow{\mathrm{am}}$ and $\xrightarrow{\mathrm{u}}$.

Proposition 4.3.1 (Acyclicity of MLL_0 proofs)**.** *The translation* $\mathtt{ps}(\pi)$ *of a sequent calculus proof* π *of* MLL_0 *is a typed proof net.*

Proof. The proof is the same as that of Proposition 4.2.15, except that we drop the connectedness requirement, which allows to treat the translation of the $(\bot)$ rule. Concretely, we prove that $\mathtt{ps}(\pi)$ is acyclic by induction on the structure of the MLL_0 proof π, and contrary to the proof of Proposition 4.2.15 we don't need to check any equation on the number of nodes and arrows. □

Of course, the converse does not hold: consider for instance the proof structure whose only internal node is a $\bot$-node. If one wants to recover a correctness criterion as in Section 4.2, one possible fix is the introduction of jumps, restoring the connectivity of $\bot$-nodes.

4.3.1.1 Proof structures with jumps

A **jump function** on an MLL_0-proof structure $\mathcal{S}$ is a function mapping each $\bot$-node $B \in \mathcal{N}_\bot(\mathcal{S})$ to some internal node $j(B) \in \mathcal{N}(\mathcal{S})$, where $\mathcal{N}(\mathcal{S})$ (resp. $\mathcal{N}_\bot(\mathcal{S})$) is the set of nodes (resp. $\bot$-nodes) of $\mathcal{S}$. A **proof structure with jumps** is a pair $(\mathcal{S}, j)$, where $\mathcal{S}$ is an MLL_0-proof structure and j is a jump function on $\mathcal{S}$. Given a switching φ of $\mathcal{S}$, the switching graph $(\mathcal{S}, j)^\varphi$ is obtained as previously, with the addition of an arrow from $j(B)$ to B for each $\bot$-node B.

A **proof net with jumps** is a proof structure with jumps such that each switching graph is acyclic. A proof net with jumps is **connected** if all its switching graphs are connected.

Proposition 4.3.2 (Soundness of Correctness with Units)**.** *The translation* $\mathtt{ps}(\pi)$ *of a sequent calculus proof* π *of* MLL_0 *can be equipped with a jump function to obtain a connected proof net with jumps.*

Proof. We reason by induction on π, the only interesting case being that of the $(\bot)$ rule. In this case, it is sufficient to apply the induction hypothesis and observe that the immediate subproof π_1 of π involves at least one rule: then, attaching a new $\bot$-node to any node of $\mathtt{ps}(\pi_1)$ *via* a jump edge does not introduce cycles and preserves the number of connected components of switching graphs. □

drawing — Lionel

Let us insist on the fact that the jump function thus obtained is not defined uniquely by π: the existence of jumps making switching graphs connected acyclic should be considered as a side condition rather than as part of the structure.

example — Lionel

4.3.1.2 Sequentialization of connected proof nets with jumps

As a converse to Proposition 4.3.2, we will show that if a typed MLL_0-proof structure $\mathcal{S}$ can be equipped with a jump function j making $(\mathcal{S}, j)$ a connected proof net with jumps, then $\mathcal{S}$ is the translation of a proof tree of MLL_0. For that purpose, we will first establish that the image of the jump function can be restricted to *ax*- and **1**-nodes.

Given a proof structure with jumps $(\mathcal{S}, j)$ we consider the graph $\mathcal{G}(\mathcal{S}, j)$ obtained from the underlying graph $\mathcal{G}(\mathcal{S})$ by adding an arrow from $j(B)$ to B for each $\bot$-node B. We call **initial node** any *ax*- and **1**-node of $\mathcal{S}$: initial nodes are exactly those nodes without incoming arrow in $\mathcal{G}(\mathcal{R}, j)$. We say a jump function is **initial** if each $j(B)$ is an initial node.

Lemma 4.3.3. *If $(\mathcal{R}, j)$ is a proof net with jumps then $\mathcal{G}(\mathcal{R}, j)$ is directed acyclic.*

Proof. Assume otherwise that there is a directed cycle π in $\mathcal{G}(\mathcal{R}, j)$: this must contain a subpath π' that is also a directed cycle, with the additional property that no node of $\mathcal{R}$ occurs twice as the target of an arrow of π'. In particular, if P is a $\parr$-node of $\mathcal{R}$, π' crosses at most one of the premises of P. It follows that π' is also a cycle in some switching graph of $(\mathcal{R}, j)$ which yields a contradiction. □

If $(\mathcal{R}, j)$ is a proof net with jumps and N is a node of $\mathcal{R}$, we can thus define $d_{\mathcal{R},j}(N)$ to be the maximum length of a path γ in $\mathcal{G}(\mathcal{R}, j)$ with target N.

Lemma 4.3.4. *If $(\mathcal{R}, j)$ is a connected proof net with jumps, then there exists an initial jump function j_0 on $\mathcal{R}$ such that $(\mathcal{R}, j_0)$ is also a connected proof net with jumps.*

Proof. We prove the result by induction on $\sum_{B \in \mathcal{N}_\bot(R)} d_{\mathcal{R},j}(B)$. If j is not initial, we select some $\bot$-node B such that $j(B)$ is not initial, and we define a jump function j' which is the same as j except for its value on B: we set $j'(B)$ to be the source of an incoming arrow of $j(B)$ in $\mathcal{G}(\mathcal{R}, j)$. More explicitly: if $j(B)$ is a $\parr$- or $\otimes$- or *cut*-node, then we set $j'(B)$ to be the source of any premise of $j(B)$; and if $j(B)$ is a $\bot$-node, then we set $j'(B)$ to be $j(j(B))$. This transformation does not introduce cycles and preserves the number of connected components of switching graphs, and then we can apply the induction hypothesis. □

Theorem 4.3.5 (Sequentialization with Units)**.** *For any typed connected proof net with jumps $(\mathcal{R}, j)$, the underlying structure $\mathcal{R}$ is the translation of a sequent calculus proof of MLL_0.*

Proof. Let $(\mathcal{R}, j)$ be a connected MLL_0 proof net with jumps. By the previous result, we can assume j to be initial. Consider the (jump-free) structure $\mathcal{R}'$ obtained from $\mathcal{R}$ as follows:

- the nodes of $\mathcal{G}(\mathcal{R}')$ are those of $\mathcal{G}(\mathcal{R})$ minus those that were $\bot$-nodes;
- the labels of nodes in $\mathcal{R}'$ are the same as in $\mathcal{R}$, except for initial nodes which become *hyp*-nodes, as introduced in Section 4.2.4;

- the arrows of $\mathcal{G}(\mathcal{R}')$ are those of $\mathcal{G}(\mathcal{R})$, with the same source and target, except for the source of those arrows that were conclusions of $\bot$-node: for each b with source B a $\bot$-node of $\mathcal{R}$, we set $\mathtt{s}_{\mathcal{R}'}(b) \coloneqq j(B)$ (which is an *hyp*-node in $\mathcal{R}'$ because $j(B)$ is initial);
- the order of premises and the interface of $\mathcal{R}'$ are the same as those of $\mathcal{R}$.

drawing and example — Lionel

In short, $\mathcal{R}'$ is obtained by merging each jump arrow with the conclusion of the corresponding $\bot$-node.

Observe that any switching path in $\mathcal{R}'$ induces a path in $\mathcal{R}$ with the same endpoints (identifying each *ax*- or **1**-node in $\mathcal{R}$ with the corresponding *hyp*-node in $\mathcal{R}'$). Conversely, any switching path in $\mathcal{R}$ without $\bot$-node as an endpoint, induces a switching path in $\mathcal{R}'$ with the same endpoints. Hence $\mathcal{R}'$ is a connected proof net and we can apply Theorem 4.2.39: $\mathcal{R}'$ is connected sequential.

If moreover $\mathcal{R}$ is typed, we construct an MLL_0-proof π such that $\mathtt{ps}(\pi) = \mathcal{R}$, by induction on the connected sequentiality of $\mathcal{R}'$.

(S1) If $\mathcal{R}'$ is reduced to an *hyp*-node, then $\mathcal{R}$ is reduced to a number of $\bot$-nodes $B_1, \dots, B_n$, plus one node N such that $j(B_i) = N$ for $1 \leq i \leq n$, and N is either a **1**-node, or an *ax*-node with conclusions typed A and $A^\bot$. Then we can set π to be either:

$$\frac{}{\mathbf{1}}\ (\mathbf{1})$$

or

$$\frac{}{A, A^\bot}\ (ax)$$

followed by n applications of the $(\bot)$ rule, and appropriate exchange rules to match the interface of $\mathcal{R}$.

(S2) If $\mathcal{R}'$ is obtained as a cut C between two proof structures $\mathcal{R}'_1$ and $\mathcal{R}'_2$, where $\mathcal{R}'_1$ and $\mathcal{R}'_2$ are connected sequential structures, then we can freely assume that the cut is between the first conclusion of $\mathcal{R}'_1$ and the first conclusion of $\mathcal{R}'_2$. Moreover, up to a simultaneous reordering of the conclusions of $\mathcal{R}'$ and $\mathcal{R}$ (which amounts to apply (ex) rules at the bottom of the proof under construction), we can assume that the conclusion Γ of $\mathcal{R}$ is of the shape Γ_1, Γ_2, where each Γ_i matches the interface of $\mathcal{R}'_i$ minus its last conclusion.

Then $\mathcal{R}$ is necessarily obtained as the same cut C between the first conclusion of $\mathcal{R}_1$ and the first conclusion of $\mathcal{R}_2$, where $\mathcal{R}_1$ and $\mathcal{R}_2$ are connected proof nets such that each $\mathcal{R}'_i$ is obtained from $\mathcal{R}_i$ as above. Moreover $\mathcal{R}_1$ and $\mathcal{R}_2$ are typed with conclusion sequents A, Γ_1 and $A^\bot, \Gamma_2$, so that the premises of C in $\mathcal{R}$ have dual types A and $A^\bot$. The induction hypothesis yields π_1 and π_2 such that $\mathtt{ps}(\pi_i) = \mathcal{R}_i$, and we can set

$$\pi \coloneqq \frac{\overset{\pi_1}{\vdots}\ A, \Gamma_1 \qquad \overset{\pi_2}{\vdots}\ A^\bot, \Gamma_2}{\Gamma_1, \Gamma_2}\ (cut)$$

Cases (S3) and (S4) are treated similarly to (S2). □

Observe that, given a proof π in MLL_0, the proof obtained by sequentializing $\mathtt{ps}(\pi)$ might be quite different from π, as the rule $(\bot)$ is only applied immediately below (ax) or $(\mathbf{1})$: this is because we have restricted the sequentialization process to initial jumps, and we could in fact inline this requirement into the translation provided by Proposition 4.3.2. On the other hand, allowing jumps from arbitrary nodes makes it easier to describe the preservation of connected proof nets with jumps under cut elimination.

4.3.1.3 Jumps and cut elimination

Recall that cut elimination in MLL_0-proof structures is the same as in MLL_v-proof structures, with the addition of the $\mathbf{1}/\bot$ case, which amounts to removing any connected component made of a cut between a $\mathbf{1}$-node and a $\bot$-node. This preserves connected sequentiality with jumps, in the following sense:

Lemma 4.3.6. *If $(\mathcal{S}, j)$ is a connected proof net with jumps, and $\mathcal{S} \xrightarrow{\text{amu}} \mathcal{S}'$ then there exists a jump function j' on $\mathcal{S}'$ making $(\mathcal{S}', j')$ a connected proof net with jumps.*

Proof. By Lemma 4.3.4, we can assume j to be initial. Then we define j' to be the same as j except on those $\bot$-nodes B such that $j(B)$ is a premise node of the *cut*-node C, eliminated in the step $\mathcal{S} \xrightarrow{\text{amu}} \mathcal{S}'$. If C is a cut between a $\mathbf{1}$-node P_0 and a $\bot$-node B_0, then for each $\bot$-node B such that $j(B) = P_0$, we set $j'(B) \coloneqq j(B_0)$. And if C is a cut between an ax-node A and some other node N, then for each $\bot$-node B such that $j(B) = A$, we set $j'(B) \coloneqq N$.

drawing — Lionel

drawing — Lionel

In both cases, this transformation does not introduce cycles in switching graphs and it preserves their number of connected components. It follows that $(\mathcal{S}, j')$ is also a connected proof net with jumps. Observe that the proofs of Lemmas 4.2.21 and 4.2.22 only rely on transformations of switching graphs that are local to the eliminated cut. They can thus be adapted straightforwardly to the reduction $\mathcal{S} \xrightarrow{\text{amu}} \mathcal{S}'$, considering the switching graphs of $(\mathcal{S}, j)$ and of $(\mathcal{S}', j')$. □

Tracing the rewriting of jump functions through cut elimination steps is tedious, and we have already explained that jumps are not really intended to be part of the structure of proof nets: they only exist because we need to relax the connectedness condition to obtain a sequentialization result with $\bot$-nodes. In most cases, the additional technicality is not worth the effort: a much simpler course is to drop connectedness, thus considering proof nets rather than connected proof nets. On the logical side, this amounts to augment the sequent calculus with so-called *mix* rules.

4.3.2 The *mix* rules and sequentialization without connectedness

Note that neither the Bungee Jumping Lemma (Lemma 4.2.37) nor the partial order $\prec$ on arrows (Lemma 4.2.35) depend on the connectedness of the proof net under consideration. Moreover, the only place where we use the connectedness assumption in the proof of Theorem 4.2.39 is in the case without internal arrows: then we have a sum of connected components reduced to *hyp*-nodes with their conclusions, and we use connectedness to ensure there is exactly one such component.

We can thus apply the very same sequentialization technique for general proof nets as for connected proof nets, provided we include rules to form arbitrary sums of nets. This is precisely the behaviour of the *mix* rules.

4.3.2.1 The *mix* rules as proof net constructions

Recall from 3.7.3 that the two *mix* rules are the nullary *mix* rule (mix_0) and the binary *mix* rule (mix_2):

$$\frac{}{\vdash}\ (mix_0) \qquad \text{and} \qquad \frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}\ (mix_2)\ .$$

We can interpret these two rules as proof structure constructions. The (mix_0) rule is translated into the empty proof structure. The (mix_2) rule applied to two proofs π_1 and π_2 which translate into the proof structures $\mathcal{S}_1$ and $\mathcal{S}_2$ leads to the disjoint union of $\mathcal{S}_1$ and $\mathcal{S}_2$:

.

Note that eliminating a cut against a (mix_2) rule does not change the associated proof structure.

In presence of both (mix_0) and (mix_2), the whole family of n-ary *mix* rules:

$$\frac{\vdash \Gamma_1 \quad \cdots \quad \vdash \Gamma_n}{\vdash \Gamma_1, \ldots, \Gamma_n}\ (mix_n)$$

are derivable. We may write (mix) for any (mix_n) and, e.g., $\mathsf{MLL}_0 + (mix)$ for $\mathsf{MLL}_0 + (mix_0) + (mix_2)$.

Note that, in $\mathsf{MLL}_0 + (mix)$, we can construct a proof whose translation is a single $\bot$-node:

$$\frac{\dfrac{}{\vdash}\ (mix_0)}{\vdash \bot}\ (\bot)\ .$$

In the remaining, we allow typed *hyp*-nodes whose conclusions induce a derivable sequent in $\mathsf{MLL}_0 + (mix)$, so that *hyp*-nodes subsume *ax*-nodes, but also **1**- and $\bot$-nodes, even in the typed case.

4.3.2.2 Sequentialization of proof nets

The proof of sequentialization now follows the same path as in the connected case. We first relax the notion of sequentiality, to allow for arbitrary sums of sequential nets.

We say that an unordered proof structure $\mathcal{S}$ is **sequential** if, assuming inductively that $\mathcal{S}_1$ and $\mathcal{S}_2$ are sequential unordered proof structures, either one of the conditions (S1) to (S4) holds, or one of the following holds:

(S5) $\mathcal{S}$ is the empty proof structure;

(S6) $\mathcal{S}$ is the sum of $\mathcal{S}_1$ and $\mathcal{S}_2$.

And we say that a proof structure $\mathcal{S}$ is **sequential** if the underlying unordered proof structure is. Note that (S1) includes the case of a single **1**- or $\bot$-node.

Lemma 4.3.7 (Untyped correctness)**.** *Any sequential proof structure is a proof net.*

Proof. By induction on the definition of sequential proof structures: again, cases (S1) to (S4) are treated exactly as in the proof of Proposition 4.2.15; case (S5) is trivial, and (S6) follow straightforwardly from the induction hypotheses. □

The analogue of Lemma 4.2.27 is obtained similarly:

Lemma 4.3.8. *For any proof structure $\mathcal{S}$:*

- *$\mathcal{S}$ is sequential iff $\mathcal{S}[\otimes/cut]$ is.*
- *$\mathcal{S}$ is a proof net iff $\mathcal{S}[\otimes/cut]$ is.*

Theorem 4.3.9 (Sequentialization with (mix))**.** *Any proof net $\mathcal{R}$ is sequential. If moreover $\mathcal{R}$ is typed (in $\mathsf{MLL}_0 + (mix)$) then it is the translation of a sequent calculus proof of $\mathsf{MLL}_0 + (mix)$.*

Proof. Suppose $\mathcal{R}$ is a proof net. As in the proof of Theorem 4.2.39, we can assume that $\mathcal{R}$ is *cut*-free, and we reason by induction on the number of internal nodes of $\mathcal{R}$.

If $\mathcal{R}$ has no internal node, then it must be empty and we apply (S5).

If the only internal nodes of $\mathcal{R}$ are *hyp*-nodes, then they must be terminal (there is no internal node with premises). Pick one: its connected component is sequential by (S1); the remaining components form a sequential structure by induction hypothesis; and we conclude by (S6).

Otherwise, $\mathcal{R}$ must contain at least one internal arrow. We pick one that is maximal for $\prec$ (among internal arrows), and write n for its target, which is terminal by Lemma 4.2.36. We can then conclude as in the proof of Theorem 4.2.39: if n is a $\parr$-node then we apply (S4); otherwise it is a splitting $\otimes$-node by Lemma 4.2.38 and we apply (S3).

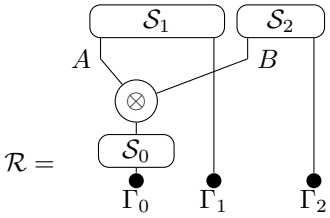
For the second part, assuming $\mathcal{R}$ is typed, we construct a suitable proof of $\mathsf{MLL}_0 + (mix)$ by a straightforward induction on $\mathcal{R}$ as a sequential structure. □

4.3.3 Concluding remarks on sequentialization

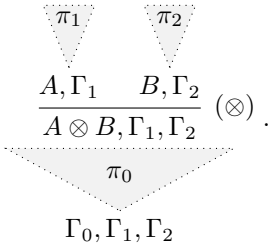
The sequentialization process is an important technical step in any introduction to proof nets. In the previous sections we have chosen to stick to a minimalistic and quite narrow point of view: we have introduced just enough notions to establish Theorem 4.3.9. In this paragraph, we try provide a bigger picture of the subject, and relate our proof method with other techniques from the literature.

4.3.3.1 On splitting nodes

Recall that, although we have focused our procedure on the notion of splitting terminal $\otimes$-node, we have defined *splitting* $\otimes$*-nodes* more generally, as those $\otimes$-nodes that are not part of a cycle in the structure. A splitting $\otimes$-node T in a typed structure $\mathcal{S}$ with conclusion Γ induces three components in $\mathcal{G}(\mathcal{S})$ that are connected only through T: one for each premise, and one for the conclusion. Up to a reordering of Γ, we can then depict $\mathcal{S}$ as:

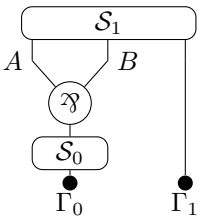


with $\Gamma = \Gamma_0, \Gamma_1, \Gamma_2$. Note that, whereas $\mathcal{S}_1$ an $\mathcal{S}_2$ are indeed typed structures whose conclusions typed A and B have been merged into T, $\mathcal{S}_0$ is not really a proof structure in the sense we have considered. Rather, $\mathcal{S}_0$ could be considered as an *open proof structure*: we will not define the notion formally, but it is the direct analogue of *open proof trees* in the setting of proof structures, and it is easy to extend the definitions so that open proof trees translate to typed open proof nets. Then, provided $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are the translations of proof trees, say π_0, π_1 and π_2 respectively, it is easy to check that $\mathcal{S}$ is also the translation of a proof tree:

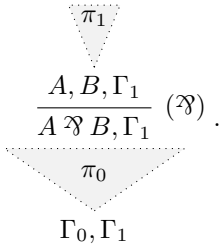


Note that we could define splitting *cut*-nodes in the same way, and treat them in the exact same manner, except that there is no third component associated with the conclusion: this is an alternative to considering *cut*-free nets up to the transformation $-[\otimes/cut]$.

A similar approach can be applied to $\parr$-nodes. We say a $\parr$-node P in a structure $\mathcal{S}$ is **splitting** if *its conclusion arrow* is not part of a cycle in $\mathcal{G}(\mathcal{S})$: in other words, there is no path between the conclusion of P and any of its premises, other than through P. A splitting $\parr$-node P in a typed structure $\mathcal{S}$ with conclusion Γ now induces two components in $\mathcal{G}(\mathcal{S})$ that are connected only through P: one containing the premises, and one containing the conclusion. Up to a reordering of Γ, we can then depict $\mathcal{S}$ as:

$\mathcal{R} =$ 

with $\Gamma = \Gamma_0, \Gamma_1$. Again, provided $\mathcal{S}_0$ and $\mathcal{S}_1$ are the translations of (possibly open) proof trees, say π_0 and π_1 respectively, $\mathcal{S}$ is also the translation of a proof tree:



So, instead of focusing on terminal nodes, a sequentialization strategy could be as follows: if a non empty proof net has a terminal *hyp*-node, this component is sequential and we proceed inductively with the rest; otherwise it contains a $\parr$-node or a $\otimes$-node, and it is then sufficient to show that one is splitting to reason inductively. To show that proof nets are sequential, it is thus sufficient to show that a (*cut*-free) proof net that is not reduced to terminal *hyp*-nodes contains a splitting ($\otimes$- or $\parr$-) node.

It turns out that Bungee Jumping also allows to follow this strategy:

Lemma 4.3.10. *Let $\mathcal{R}$ be a non empty proof net:*

- *if $\mathcal{R}$ contains a $\parr$-node, then one of them is splitting;*
- *if all the terminal nodes of $\mathcal{R}$ are $\otimes$-nodes (this is in particular the case if $\mathcal{R}$ has no $\parr$-node, and no terminal hyp-node) then one of them is splitting.*

Proof. The second item is a direct consequence of Lemmas 4.2.36 and 4.2.38.

The proof of the first item is similar to that of Lemma 4.2.38. If $\mathcal{R}$ contains a $\parr$-node, then we pick an arrow a_1 that is a premise of some $\parr$-node P, and such that a_1 is maximal for $\prec$ among premises of $\parr$-nodes. We show that P is splitting.

Assume, towards a contradiction, that P is not splitting: we pick a cycle ω in $\mathcal{B}(P)$, with a_0^+ as first edge. Consider the first bridge $b_1^+ b_2^-$ of ω: write κ for its pier, with premises b_1 and b_2. Then write γ for the prefix of ω with last edge b_1^+: this is strong (its first edge is a_0^+), simple (it is a subpath of a cycle), and bridge-free (by construction). We obtain $\gamma : a_1 \frown b_1$.

Moreover, there is no path $\delta : b_1 \frown p$ with p occurring in γ (hence in ω): this would contradict the Bungee Jumping Lemma. We thus obtain $a_1 \prec b_1$, contradicting the maximality of a_1 since b_1 is a premise of a $\parr$-node. □

Remark 4.3.11. We have argued that, to establish that proof nets are sequential, it is sufficient to show that any (*cut*-free) proof net with at least a $\otimes$- or $\parr$-node admits a splitting node. Conversely, observe that any sequential proof net $\mathcal{R}$ not reduced to a sum of terminal *hyp*-nodes contains a terminal splitting node: starting from the root of a derivation of sequentiality of $\mathcal{R}$, we split $\mathcal{R}$ into disjoint connected components *via* applications of (S6) until we reach an application of (S2), (S3) or (S4), yielding either a splitting *cut*-node, or a terminal and splitting $\otimes$-node, or a terminal (hence splitting) $\parr$-node.

Hence proving sequentialization for proof nets (connected or not) is essentially equivalent to establishing the existence of splitting nodes.

4.3.3.2 Sequentialization in the literature

In the first paper on the subject [20], Girard proves sequentialization for MLL_v using a correctness criterion called "the longtrip criterion": the proof was based on the existence of a splitting $\otimes$-node. Shortly after Girard's discovery of this criterion, Danos and Regnier found a simplification and proved that it was equivalent to the acyclicity and connectedness of switching graphs: this is the criterion we presented in this chapter. Danos and Regnier's proof of sequentialization was based on the existence of a splitting $\parr$-node (called "section" in [9]).

The complexity of deciding whether a proof structure $\mathcal{S}$ is a proof net by testing the acyclicity of all switching graphs is exponential in the number of $\parr$-nodes of $\mathcal{S}$. Several alternative criteria have thus been designed, including ones that can be checked in linear time: we refer to the work of Jacobé de Naurois and Mogbil [38] for a review of the subject. The latter authors moreover give a lower bound, establishing that the decision problem is NL-complete.

Note that, in MLL_v, once we know that a given typed proof structure is a proof net, deciding whether it is a connected proof net (and thus the translation of a proof tree) is easily done in linear time: as in the proof of Lemma 4.2.13, it is sufficient to count the numbers n of nodes, p of $\parr$-nodes and a of arrows, and to check that $n + p = a + 1$.

4.3.3.3 Proof nets with units

In presence of multiplicative units, we have considered two variants: one which introduces non canonical information (jumps), but allows to characterize efficiently those proof structures that are translations of proof trees in MLL_u; and

one which does not introduce jumps but relaxes the correctness criterion, leading to the introduction of (mix) rules.

Note that the MLL_u proofs

$$\dfrac{\dfrac{\dfrac{}{\mathbf{1}}(\mathbf{1})}{\bot,\mathbf{1}}(\bot) \quad \dfrac{\dfrac{}{\mathbf{1}}(\mathbf{1})}{\bot,\mathbf{1}}(\bot)}{\dfrac{\dfrac{\bot\otimes\bot,\mathbf{1},\mathbf{1}}{\mathbf{1},\mathbf{1},\bot\otimes\bot}(ex(2,3,1))}{\mathbf{1}\parr\mathbf{1},\bot\otimes\bot}(\otimes)}(\parr) \qquad \text{and} \qquad \dfrac{\dfrac{\dfrac{}{\mathbf{1}}(\mathbf{1})}{\bot,\mathbf{1}}(\bot) \quad \dfrac{\dfrac{}{\mathbf{1}}(\mathbf{1})}{\bot,\mathbf{1}}(\bot)}{\dfrac{\dfrac{\bot\otimes\bot,\mathbf{1},\mathbf{1}}{\mathbf{1},\mathbf{1},\bot\otimes\bot}(ex(3,2,1))}{\mathbf{1}\parr\mathbf{1},\bot\otimes\bot}(\otimes)}(\parr)$$

are translated to the same proof net, but induce different jumps. Forgetting about jumps identifies those two proof trees, but it is easy to check that this identification cannot be obtained by the application of local commutations between independent inferences.[4]

We should define these at some point. In fact we should discuss proof equivalence more in detail: see the todos in the section on correctness. — Lionel

It is then natural to ask whether one can introduce another notion of proof nets for MLL_u which would be canonical:[5] two proof trees are translated to the same proof net iff they are equivalent up to local commutations. It turns out that there is a strong, theoretical obstacle to the existence of such a notion. Indeed, Heijltjes and Houston [29] have established that proof equivalence in MLL_u is PSPACE-complete. It follows that, for such a canonical notion proof nets, if the translation from proof trees to proof nets is tractable, then the problem of deciding whether two proof nets are equal cannot be (and *vice versa*).

So, to work with units (and the same will apply in presence of exponential modalities), one must chose to either overspecify the structure of proof nets, *e.g.* with jumps, or to accept the (mix) rules and drop the connectedness condition. Most of the time we will opt for the latter, simpler solution.

4.4 Multiplicative Exponential Proof Nets

We introduce now the exponential connectives which provide linear logic with real expressive power. The rewriting theory of proof nets becomes much richer.

[4] Another manifestation of this discrepancy, that goes beyond the scope of the present chapter, is that there are denotational models of MLL_u, *i.e.* $*$-autonomous categories (see Section 6.1.1.1), that do not equate the above two proof trees.

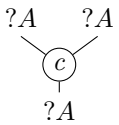
[5] In other words, such a notion should be a syntactic presentation of $*$-autonomous categories.

4.4.1 Boxes

A first, naïve idea to define proof structures for MELL, is to simply introduce one new node for each new rule — (!), (?), (c) and (w):

for $\dfrac{?\Gamma, A}{?\Gamma, !A}\ (!)$

for $\dfrac{\Gamma, A}{\Gamma, ?A}\ (?)$



for $\dfrac{\Gamma, ?A, ?A}{\Gamma, ?A}\ (c)$

for $\dfrac{\Gamma}{\Gamma, ?A}\ (w)$

but it turns out that too much information is lost w.r.t. sequent calculus. Indeed, the promotion rule

$$\frac{?\Gamma, A}{?\Gamma, !A}\ (!)$$

imposes a constraint on the shape of the context, where each formula must be of the shape $?B$: so we must be able to impose a similar constraint when adding a !-node to a proof structure. Even more importantly, the dynamics of cut elimination on exponentials is not defined locally. Indeed, the key cut elimination steps from Fig. 3.5 might involve the duplication or erasure of a whole proof sub-tree.

We must thus be able to identify, within a proof structure, a sub-structure associated with each !-node. And this requires additional information:

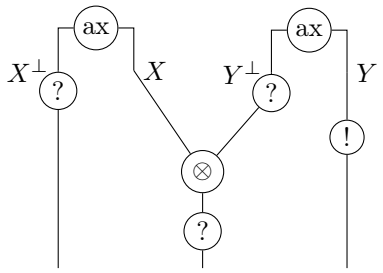
Example 4.4.1. Consider the two proof trees:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{X^\perp, X}\ (ax)}{?X^\perp, X}\ (?) \qquad \dfrac{\dfrac{}{Y^\perp, Y}\ (ax)}{?Y^\perp, Y}\ (?)}{?X^\perp, X \otimes ?Y^\perp, Y}\ (\otimes)}{?X^\perp, ?(X \otimes ?Y^\perp), Y}\ (?)}{?X^\perp, ?(X \otimes ?Y^\perp), !Y}\ (!)$$

and

$$\dfrac{\dfrac{\dfrac{}{X^\perp, X}\,(ax)}{?X^\perp, X}\,(?) \qquad \dfrac{\dfrac{\dfrac{}{Y^\perp, Y}\,(ax)}{?Y^\perp, Y}\,(?)}{?Y^\perp, !Y}\,(!)}{\dfrac{?X^\perp, X \otimes ?Y^\perp, !Y}{?X^\perp, ?(X \otimes ?Y^\perp), !Y}\,(?)}\,(\otimes)$$

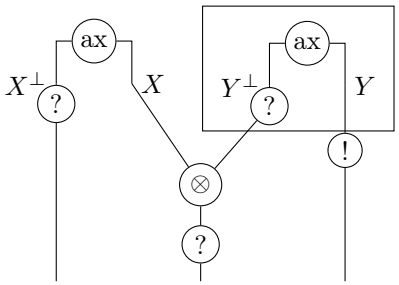
which differ only by the level at which we apply the promotion rule. By the previous translation, both are mapped to the structure:



whereas the two proofs behave very differently from each other. Indeed, if the $!Y$ conclusion is cut against a contraction or a weakening, different sub-trees must be duplicated or erased — in particular, the axiom on X will be involved in this transformation for the former proof tree, but not for the latter.

An MELL proof structure must thus come with additional information: the standard technique is to equip proof structures with a notion of **promotion box**, identifying the whole sub-structure corresponding to the proof tree associated with the application of a promotion rule.

Example 4.4.2. Graphically, the proof structure associated with the second proof of Example 4.4.1 might be depicted as:



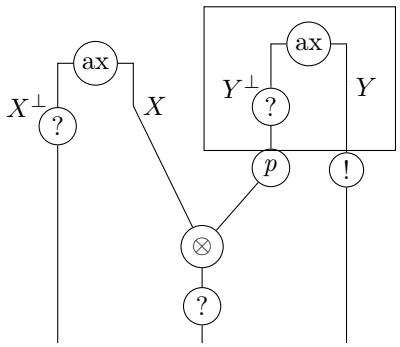
where the bounding box delineates the sub-structure associated with the proof tree

$$\dfrac{\dfrac{}{Y^\perp, Y}\,(ax)}{?Y^\perp, Y}\,(?)\ .$$

For technical reasons, it is in fact more practical to add specific nodes for the auxiliary ports of promotions boxes, *i.e.* those arrows which cross the boundary of the box. This way, given a box in a proof structure, we can distinguish between:

- the arrows and nodes inside the box, which are arrows and internal nodes of the proof structure associated with this box — the *content* of the box;
- the !-node and the auxiliary ports of the box, which are on the border of the box, and correspond with the conclusion nodes of the content, but are also internal nodes of the top level proof structure;
- the nodes and arrows outside the box, which can be conclusions of nodes on the border of the box, but are never directly connected with nodes inside the box.

Example 4.4.3. Again, the proof structure associated with the second proof of Example 4.4.1 might be depicted as:



where we have added an auxiliary port node marking the border of the box, where it was crossed by the conclusion of the ?-node inside the box.

Note that the promotion boxes of a proof structure must be correctly nested: given two !-nodes, either the associated boxes are disjoint, or one of the boxes is included in the other. This means that boxes are organised into a tree, that we will call the *box tree* of the proof structure. In the next section we make that definition formal, and we show that it is equivalent to considering boxes as new *hyp*-nodes, to which we inductively associate proof structures.

4.4.2 Proof Structures for MELL

An **untyped MELL proof structure** $\mathcal{S}$ is given by:

- a *directed acyclic graph* $\mathcal{G}(\mathcal{S})$;
- a labelling of the nodes of $\mathcal{G}(\mathcal{S})$ with labels in $\{ax, cut, \otimes, \parr, !, d, w, c, p, \bullet\}$ such that:

  - each ax-node has two conclusions and no premise;
  - each cut-node has two premises and no conclusion;
  - each $\otimes$-node, $\parr$-node or c-node has two premises and one conclusion;
  - each d-node or p-node has one premise and one conclusion;
  - each w-node has no premise and one conclusion;
  - each $\bullet$-node has one premise and no conclusion;
- an ordering of the premises of each $\otimes$-node, $\parr$-node or c-node;
- an ordering of the $\bullet$-nodes;
- a rooted tree $\mathcal{B}_{\mathcal{S}}$, called the **box tree** of $\mathcal{S}$, whose nodes are the !-nodes of $\mathcal{S}$, plus the root node that we denote by $\star_{\mathcal{S}}$ (the **top level** of $\mathcal{S}$);
- a map, also denoted $\mathcal{B}_{\mathcal{S}}(-)$, associating with each node or arrow of $\mathcal{G}(\mathcal{S})$ a node of $\mathcal{B}_{\mathcal{S}}$ — the **box level** of the former node or arrow — such that:
  - all $\bullet$-nodes are mapped to the root of $\mathcal{B}_{\mathcal{S}}$;
  - for each !-node n, $\mathcal{B}_{\mathcal{S}}(n)$ is the parent of n in $\mathcal{B}_{\mathcal{S}}$, and the unique premise of n is mapped to n;
  - the unique premise of each p-node n is mapped to a child of $\mathcal{B}_{\mathcal{S}}(n)$;
  - except for the premises of !-nodes and p-nodes, the premises and conclusions of each node n are mapped to $\mathcal{B}_{\mathcal{S}}(n)$.

Note that the conclusions of a proof structure are always at top level. Moreover observe that the function $\mathcal{B}_{\mathcal{S}}(-)$ is uniquely defined by its value on cut-nodes and on the premises of p-nodes: in all the other cases, the box level of a node or arrow can be deduced from that of its conclusion or target respectively.

We will often omit the subscript $\mathcal{S}$, when the proof structure under consideration is clear from the context.

Given a !-node n in a proof structure $\mathcal{S}$, we refer to the following data as the **promotion box** associated with n in $\mathcal{S}$:

- the **main door** of the box is n;
- the **auxiliary doors** of the box are the p-nodes of $\mathcal{S}$ whose premises are at box level n — and whose conclusion is thus at box level $\mathcal{B}(n)$;
- the **border** of the box is the set of its main door and auxiliary doors — note that these are the nodes having a premise at box level n and a conclusion at box level $\mathcal{B}(n)$;
- the **level** of the box is $\mathcal{B}(n)$ — which is the common level of its doors;
- the **content** of the box is the subgraph of $\mathcal{G}$, made of the border of the box, plus those nodes and edges whose box level occurs in the subtree of $\mathcal{B}$ rooted at n.

We will often refer to the box of n just as n.

Given an enumeration I of the border of n, we can define the **proof structure of** n, denoted by $\beta(n, I)$ as follows:

- $\mathcal{G}(\beta(n, I))$ is the content of n;
- the labelling of nodes is the same as in $\mathcal{S}$, except for border nodes, which become $\bullet$-nodes;
- the ordering of premises is inherited from that in $\mathcal{S}$;
- the interface, *i.e.* the order of the $\bullet$-nodes, is given by I;
- $\mathcal{B}_{\beta(n,I)}$ is the subtree of $\mathcal{B}_{\mathcal{S}}$ rooted at n;
- the map $\mathcal{B}_{\beta(n,I)}(-)$ is defined by restricting $\mathcal{B}_{\mathcal{S}}(-)$ except for $\bullet$-nodes which are now at top level $\star_{\beta(n,I)} = n$.

We will often keep the interface implicit and just write $\beta(n)$. In particular in pictures, I is given by the order of doors from left to right unless explicitly noted. We might also abusively refer to $\beta(n)$ as the content of n, since all the structure is canonically defined as soon as the interface is fixed.

Show that this is equivalent to black boxes. — Lionel

The ?-*tree* of an edge of type ?_ is defined inductively by:

- If the edge is conclusion of an *ax* node, its ?-tree is empty.
- If the edge is conclusion of a d node, its ?-tree is this d node.
- If the edge is conclusion of a w node, its ?-tree is this w node.
- If the edge is conclusion of a c node, its ?-tree is this c node together with the ?-trees of the two premises of the c node.
- If the edge is conclusion of a p node, its ?-tree is this p node together with the ?-tree of the its premise.

The *size* of a ?-tree is its number of nodes.

descent path (bis): from a node downwards to a conclusion or to a cut or to a premise of ! node (that is we do not continue down through an ! node)

4.4.3 Multiplicative Exponential Linear Logic with Mix

We consider the multiplicative exponential fragment of propositional linear logic without units: MELL_v, in its one-sided version. Recall — from Section 3.3.5 — that the formulas are:

$$X \qquad X^{\perp} \qquad A \otimes B \qquad A \parr B \qquad !A \qquad ?A$$

where X ranges over positive atomic formulas and A, B range over (DM-normal) formulas, and that linear negation is the involution defined inductively by $X^{\perp\perp} \coloneqq X$, $(A \otimes B)^\perp \coloneqq A^\perp \parr B^\perp$ and $(!A)^\perp \coloneqq ?A^\perp$.

Note that the weakening rule:

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ (w)$$

behaves exactly like the $(\perp)$ rule:

$$\frac{\vdash \Gamma}{\vdash \Gamma, \perp}\ (\perp)$$

by adding a formula to a derivable sequent. This means that, to design proof nets for MELL, we will face the same connectedness issues as with multiplicative units. And these can be solved similarly: either by introducing jumps, or by relaxing the connectedness requirement and adding the (mix) rules to the system. We chose the latter path, which requires less technicality.

Remark 4.4.4. In passing, note that multiplicative units can be encoded into MELL_v, even in the absence of second order quantification.[6] Indeed, given any atomic formula X, we have a linear equivalence $\mathbf{1} \multimap\!\!\circ\ !(X \multimap X)$. More precisely, writing $\mathbf{1}' \coloneqq !(X \multimap X)$ and $\perp' \coloneqq ?(X \otimes X^\perp) = (!(X \multimap X))^\perp$, not only can we prove $\mathbf{1} \vdash \mathbf{1}'$ (*i.e.* $\vdash \perp, \mathbf{1}'$) and $\mathbf{1}' \vdash \mathbf{1}$ (*i.e.* $\vdash \perp', \mathbf{1}$), but we can derive analogues of rules $(\mathbf{1})$ and $(\perp)$ for $\mathbf{1}'$ and $\perp'$ respectively. Namely:

$$\frac{\dfrac{\dfrac{}{\vdash X^\perp, X}\ (ax)}{\vdash X^\perp \parr X}\ (\parr)}{\vdash !(X^\perp \parr X)}\ (!) \qquad \text{and} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?(X \otimes X^\perp)}\ (?)\ .$$

Moreover, cutting the first proof against the conclusion $?(X \otimes X^\perp)$ of the rule $(?)$, the cut elimination process yields the underlying proof of $\vdash \Gamma$: this mimics the cut elimination rule between $(\mathbf{1})$ and $(\perp)$. It follows that, as far as provability, cut elimination or the geometry of proof nets are concerned, MELL behaves the same with or without multiplicative units, and this whole section could be adapted to MELL_0 instead of MELL_v.

Instead of $\mathsf{MELL}_v + (mix)$, we consider the sequent calculus MELL_{mix} obtained from the rules of MLL_v by adding the (mix) rules, together with:

$$\frac{}{\vdash ?A}\ (w_0) \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ (c) \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ (?) \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ (!)$$

Due to the presence of mix rules, our presentation of the weakening rule (w_0) is equivalent to the original one $\dfrac{\vdash \Gamma}{\vdash \Gamma, ?A}\ (w)$. Indeed, the two rules are inter-derivable:

$$\frac{\vdash \Gamma \qquad \dfrac{}{\vdash ?A}\ (w_0)}{\vdash \Gamma, ?A}\ (mix_2) \qquad \frac{\dfrac{}{\vdash}\ (mix_0)}{\vdash ?A}\ (w)$$

[6] This contrasts with what was done in Section 3.3.3.3.

and, with the obvious cut elimination rules for (mix_2) and (w_0), cut elimination behaves the same in MELL_{mix} and in $\mathsf{MELL}_v + (mix)$. We keep this simulation statement informal, as the focus is on proof nets, and it should already be clear that the translations induced by both versions of weakening are the same: (w) should add a terminal weakening node to an already constructed proof net, while (w_0) should be translated as a single weakening node; and the (mix) rules allow to turn one construction into the other.

4.4.4 Correctness Criterion

acyclicity

sequentialization

4.4.5 Reduction of MELL proof nets

In this section we define the reduction steps for proof nets together with their action on paths and give some general results.

4.4.5.1 Notations and conventions

We briefly recall here some maybe nonstandard conventions. See appendix A on graphs for details.

Proof nets will be considered typed or untyped; by untyped we mean that formulas may be quotiented by a type equation such as $o = !o \multimap o$ which is used to translate call-by-name lambda-calculus into proof nets, or $o = !(o \multimap o)$ which is used to translate call-by-value lambda-calculus into proof nets, see section 4.5.4.

Recall that proof nets are oriented graphs, the arrows of which are oriented topdown in pictures. Incoming arrows in a node are its premises, while outgoing arrows are its conclusions. An *edge* is an arrow together with a *direction* ± 1: we write $e = a^+$ and say that e is a *downard edge* when the direction of e is 1 in which case we define the source and target of e to be the same as the source and target of the arrow a; we write $e = a^-$ and say that e is an *upward edge* when the direction of e is -1 in which case we define the source and target of e to be respectively the target and source of a.

A path γ of length N in a proof net is a pair $\gamma = (\mathsf{n}_0, (e_i)_{1 \leq i \leq N})$ where n_0 is a node and (e_i) is a sequence of composable edges: for each $1 \leq i \leq N$, the source of e_i is the node n_{i-1} and the target of e_i is the node n_i (thus source of e_{i+1}). The nodes n_0 and n_N are the source and target of the path which we will denote by $\gamma : \mathsf{n}_0 \to \mathsf{n}_N$. We say γ *crosses a node* n if there is an $0 < i < N$ such that $\mathsf{n} = \mathsf{n}_i$; the crossing is *downard* if e_{i-1} and e_i are downard edges, *upward* if e_{i-1} and e_i are upward edges. Note that *ax* and *cut* nodes cannot be crossed neither downardly nor upwardly. Similarly we say that γ starts (ends) upwardly (downardly) if e_1 (e_N) is upward (downard).

Except for the empty path $\epsilon_{\mathsf{n}} = (\mathsf{n}, ())$ at node n, paths will be represented as words of edges $e_1 \dots e_N$, leaving the nodes implicit. Two nonempty paths $\gamma =$

$e_1 \dots e_N$ and $\delta = e_{N+1} \dots e_M$ are composable if e_N and e_{N+1} are composable in which case their composition is $\gamma\delta = e_1 \dots e_M$. The empty path ϵ_{n} is composable on the left with any δ sourced on n and then $\epsilon_{\mathsf{n}}\delta = \delta$ and on the right with any γ targeted on n and then $\gamma\epsilon_{\mathsf{n}} = \gamma$. Being concatenation, path composition is clearly associative, thus nodes and paths in a proof net $\mathcal{R}$ form a category that we will denote by $\mathcal{R}^*$. Note that if $\gamma : \mathsf{n} \to \mathsf{n}'$ and $\delta : \mathsf{n}' \to \mathsf{n}''$ are two composable paths we denote their composition by $\gamma\delta : \mathsf{n} \to \mathsf{n}''$ whereas categorical convention would rather write: $\delta \circ \gamma : \mathsf{n} \to \mathsf{n}''$.

If γ and γ_0 are paths we write $\gamma_0 \sqsubset \gamma$ (or $\gamma \sqsupset \gamma_0$) for γ_0 is a supbath of γ, that is if there are paths γ' and γ'' such that $\gamma = \gamma'\gamma_0\gamma''$. If γ' (resp. γ'') is the empty path we say that γ_0 is a prefix (resp. a suffix) of γ and write $\gamma_0 \sqsubset_p \gamma$ (resp. $\gamma \ {}_s\!\sqsupset \gamma_0$).

Finally let us (re)define two particular form of paths in a proof net that we will use latter on. Firstly recall that, according to definition **??**, a descent path is a downward path, that is a path of the form $a_1^+ \dots a_n^+$ for some composable arrows $a_1, \dots, a_n$. We will say that the descent path is **down-maximal** when it cannot be extended downardly, that is there is no arrow a such that $a_1^+ \dots a_n^+ a^+$ is a path. Thus a down-maximal descent path ends in a conclusion of a cut node.

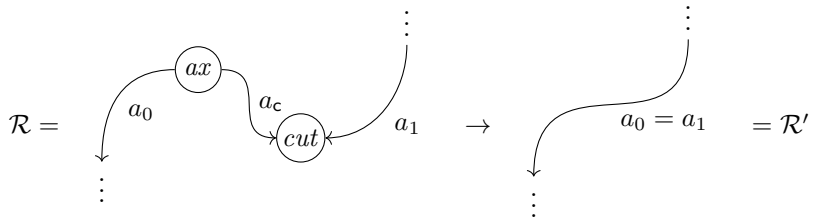
A particular case of descent path is when each arrow is labelled by a same ?-formula. When it is maximal with this property we call it an **exponential branch**. Therefore an exponential branch is a descent path starting from the conclusion of a weakening or derelection node, crossing only contraction and auxiliary door nodes, and ending on a non-? or dereliction node.

4.4.5.2 Reduction steps and the lifting functor

Let $\mathcal{R}$ be a proof net, c a *cut*-node in $\mathcal{R}$; we define by case on the nature of the cut c the proof net $\mathcal{R}'$ obtained by applying the one step reduction associated to c, that we view as a rewriting of $\mathcal{R}$ together with the map $\mathcal{L}_{\mathsf{c}}$ associating to each arrow of $\mathcal{R}'$ a path in $\mathcal{R}$.

Axiom cut case: let a_0 and a_{c} be the two conclusions of the *ax* node, a_{c} and a_1 be the two premises of the cut c.

The retract $\mathcal{R}'$ is obtained by identifying a_0 and a_1 into an arrow denoted $a_0 = a_1$, removing a_{c}, the *cut* node c and the *ax* node, all other part of the graph being unchanged.

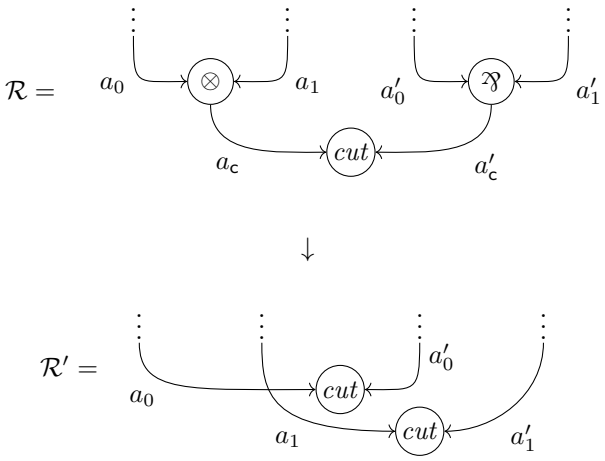


If a is any arrow in $\mathcal{R}'$ we define:

- $\mathcal{L}_{\mathsf{c}} a = a^+$ if $a \neq a_0 = a_1$ and,
- $\mathcal{L}_{\mathsf{c}}(a_0 = a_1) = a_1^+ a_{\mathsf{c}}^- a_0^+$.

Multiplicative cut case: let a_0, a_1 be respectively the left and right premise of the $\otimes$, a'_0 and a'_1 the left and right premise of the $\parr$-node, a_{c} and a'_{c} the conclusion respectively of the $\otimes$ and the $\parr$ node and the premises of the cut c.

The retract $\mathcal{R}'$ is obtained by replacing c by two cut nodes c_0 and c_1, retargetting a_i and a'_i on c_i for $i = 0, 1$, removing a_{c}, a'_{c}, the $\otimes$ and the $\parr$ nodes, leaving all other part of the graph unchanged.



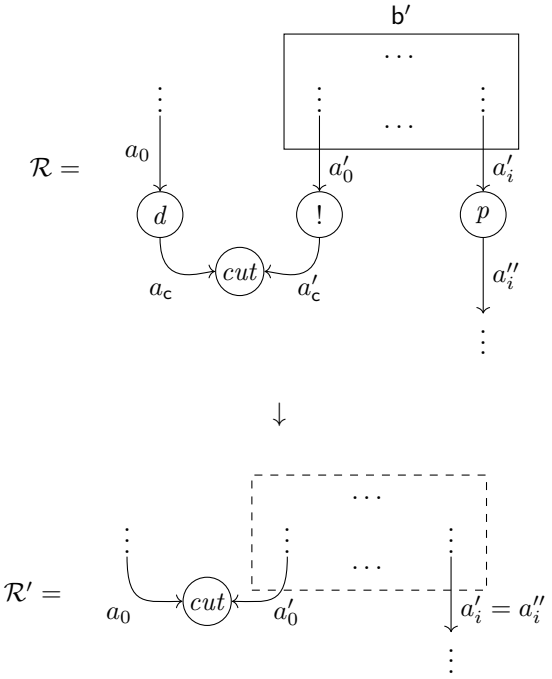
We define:

- $\mathcal{L}_{\mathsf{c}} a_\epsilon = a_\epsilon^+ a_{\mathsf{c}}^+$ for $\epsilon = 0, 1$;
- $\mathcal{L}_{\mathsf{c}} a'_\epsilon = {a'_\epsilon}^+ {a'_{\mathsf{c}}}^+$ for $\epsilon = 0, 1$;
- $\mathcal{L}_{\mathsf{c}} a = a^+$ for all other arrows in $\mathcal{R}$.

If a is any arrow in $\mathcal{R}'$ we define: $\mathcal{L}_{\mathsf{c}} a = a^+$. Note that a_i^+ and ${a'_i}^-$ are composable in $\mathcal{R}'$ but not in $\mathcal{R}$.

Dereliction cut case: let a_0 be the premise of the d-node, a_{c} be its conclusion which is also one premise of the cut c, let a'_{c} be the other premise which is therefore conclusion of a !-node associated to a box b'; let a'_0 be the premise of the !-node, a'_1, a''_1, …, a'_k, a''_k be the premises and conclusions of the p nodes of b'.

The retract $\mathcal{R}'$ is obtained by removing the d and ! nodes, all the p nodes of b', the arrows a_{c} and a'_{c}, setting c as the new target of a_0 and a'_0, for $i = 1, \ldots, k$ identifying a'_i and a''_i into an arrow $a'_i = a''_i$ the source (goal) of which is the source (goal) of a'_i (a''_i) and finally removing b' from the box structure of $\mathcal{R}$.

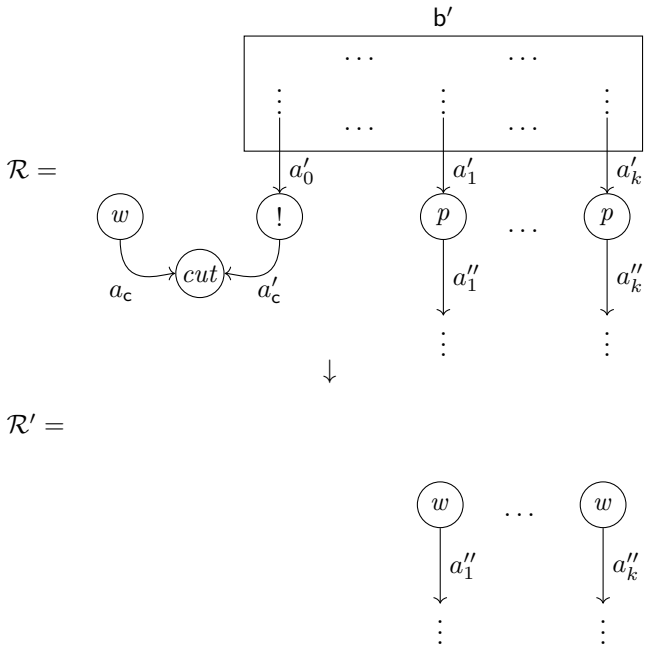


We define:

- $\mathcal{L}_{\mathsf{c}} a_0 = a_0^+ a_{\mathsf{c}}^+$;
- $\mathcal{L}_{\mathsf{c}} a'_0 = {a'_0}^+ {a'_{\mathsf{c}}}^+$;
- $\mathcal{L}_{\mathsf{c}}(a'_i = a''_i) = {a'_i}^+ {a''_i}^+$;
- $\mathcal{L}_{\mathsf{c}} a = a$ for all other arrows in $\mathcal{R}$.

Weakening cut case: let a_{c} be the conclusion of the w-node, a'_{c} be the other premise of the cut c and the conclusion of the !-node, b' be the box associated, and as before a'_i and a''_i be the premise and conclusion of each p-node of b'.

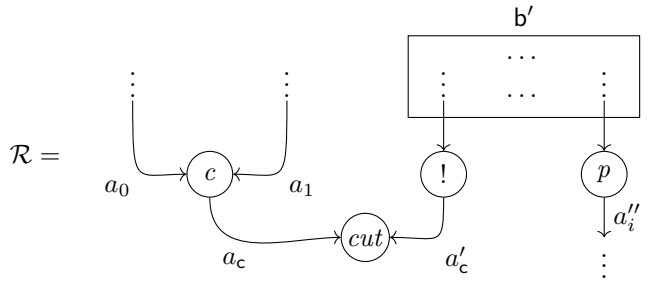
The retract $\mathcal{R}'$ is obtained by removing the w-node, the cut c, the !-node, all the nodes and arrows inside b' including a'_0 and the a'_i's and retyping the p-nodes into w nodes.

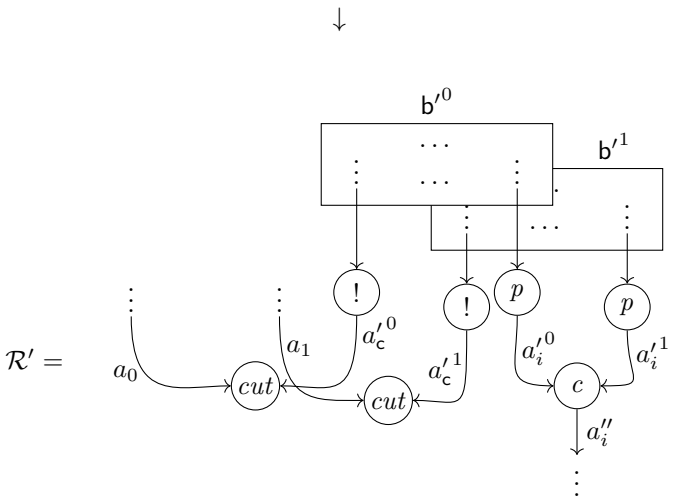


We define $\mathcal{L}_{\mathsf{c}}a = a^+$ for all arrows a in $\mathcal{R}'$.

Contraction cut case: let a_0, a_1 be the premises of the c-node, a_{c} its conclusion which is premise of the cut node c, a'_{c} be the other premise of c which if conclusion of a !-node associated to a box b', let a''_1, ..., a''_k be the conclusions of the p-nodes p_1, ... p_k of b'.

The retract $\mathcal{R}'$ is obtained by duplicating the entire content of the box b', including its ! and p nodes into two copies ${\mathsf{b}'}^0$ and ${\mathsf{b}'}^1$, replacing the c node by two cut nodes c_0 and c_1, removing the c-node, retargetting a_0 on c_0, a_1 on c_1, replacing the arrow a'_{c} by two arrows ${a'_{\mathsf{c}}}^0$ and ${a'_{\mathsf{c}}}^1$ sourced respectively on each copy of the !-node and targeted respectively on c_0 and c_1, adding k c-nodes n_1, ..., n_k and $2k$ new arrows ${a'_1}^0$, ${a'_1}^1$, ..., ${a'_k}^0$, ${a'_k}^1$, the source of ${a'_i}^\epsilon$ being the copy p_i^ϵ of p_i in ${\mathsf{b}'}^\epsilon$ for $\epsilon = 0, 1$, the target being n_i, and finally resourcing a''_i on n_i.



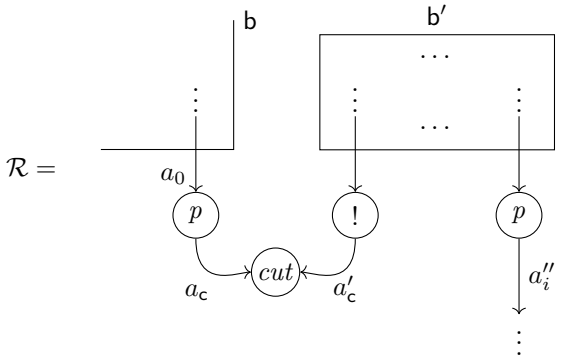


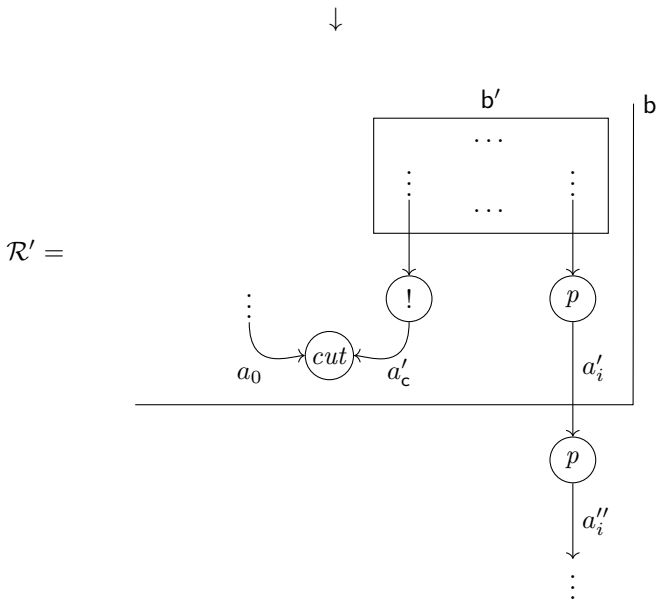
We define:

- $\mathcal{L}_{\mathsf{c}} a_\epsilon = a_\epsilon^+ a_{\mathsf{c}}^+$ for $\epsilon = 0, 1$;
- $\mathcal{L}_{\mathsf{c}} a'^{\epsilon}_{\mathsf{c}} = a'^{+}_{\mathsf{c}}$;
- $\mathcal{L}_{\mathsf{c}} a'^{\epsilon} = a'^{+}$ for any arrow a'^{ϵ} in the ϵ-copy b'^{ϵ} of b', copy of an arrow a' in b';
- $\mathcal{L}_{\mathsf{c}} a'^{\epsilon}_i = \epsilon_{p_i}$ the empty path at the p-node p_i of b';
- $\mathcal{L}_{\mathsf{c}} a''_i = a''^{+}_i$;
- $\mathcal{L}_{\mathsf{c}} a = a^+$ for any other arrow in $\mathcal{R}'$.

Commutative cut case: let a_0 be the premise of a p-node p associated to a box b, a_{c} be the conclusion of p and the premise of the cut c, a'_{c} be the other premise of c and conclusion of a !-node associated to a box b', p'_1, $\ldots$, p'_k the p-nodes of b' and a''_1, $\ldots$, a''_k the conclusions of p'_1, $\ldots$, p'_k.

The retract $\mathcal{R}'$ is obtained by removing p and a_{c}, retargetting a_0 on c, moving all the box b' inside the box b, adding k p-node p_1, $\ldots$, p_k as new doors of b and k arrows a'_1,$\ldots a'_k$ sourced and targeted respectively on p'_i and p_i and resourcing each a''_i from p_i.





We define:

- $\mathcal{L}_{\mathsf{c}} a_0 = a_0^+ a_{\mathsf{c}}^+$;
- $\mathcal{L}_{\mathsf{c}} a'_{\mathsf{c}} = {a'_{\mathsf{c}}}^+$;
- $\mathcal{L}_{\mathsf{c}} a'_i = \epsilon_{p'_i}$;
- $\mathcal{L}_{\mathsf{c}} a''_i = {a''_i}^+$;
- $\mathcal{L}_{\mathsf{c}} a = a^+$ for any other arrow a in $\mathcal{R}'$.

The map $\mathcal{L}_{\mathsf{c}}$ is extended to edges by setting $\mathcal{L}_{\mathsf{c}} a^+ = \mathcal{L}_{\mathsf{c}} a$ and $\mathcal{L}_{\mathsf{c}} a^- = \overline{\mathcal{L}_{\mathsf{c}} a}$. The definition is made so that we immediately check:

Proposition 4.4.5. *If e and f are two composable edges in $\mathcal{R}'$, then $\mathcal{L}_{\mathsf{c}} e$ and $\mathcal{L}_{\mathsf{c}} f$ are two composable paths in $\mathcal{R}$.*

This allows us to extend $\mathcal{L}_{\mathsf{c}}$ into the **lifting functor** $\mathcal{L}_{\mathsf{c}} : \mathcal{R}'^* \to \mathcal{R}^*$ from the category of paths on $\mathcal{R}'$ to the category of paths on $\mathcal{R}$. If n is the target of an edge e (or the source of an edge f) we set $\mathcal{L}_{\mathsf{c}}\mathsf{n}$ to be the target node of $\mathcal{L}_{\mathsf{c}} e$ (or the source node of $\mathcal{L}_{\mathsf{c}} f$). Note that by acyclicity n cannot be source and target of a same edge, and if n is target of e and source of f, then e and f are composable thus by the proposition above the target of $\mathcal{L}_{\mathsf{c}} e$ and the source of $\mathcal{L}_{\mathsf{c}} f$ are the same, so that $\mathcal{L}_{\mathsf{c}}\mathsf{n}$ is well defined.

Finally we turn $\mathcal{L}_{\mathsf{c}}$ into a functor by setting $\mathcal{L}_{\mathsf{c}} \epsilon_{\mathsf{n}} = \epsilon_{\mathcal{L}_{\mathsf{c}}\mathsf{n}}$ and $\mathcal{L}_{\mathsf{c}}(e_1 \dots e_n) = \mathcal{L}_{\mathsf{c}} e_1 \dots \mathcal{L}_{\mathsf{c}} e_n$. Note that $\mathcal{L}_{\mathsf{c}}$ is compatible with path reversion: $\mathcal{L}_{\mathsf{c}} \overline{\gamma} = \overline{\mathcal{L}_{\mathsf{c}} \gamma}$ for any path γ in $\mathcal{R}'^*$.

4.4.5.3 Path residuals

The **residuals of a path γ in $\mathcal{R}$** are the paths γ' in $\mathcal{R}'$ of minimal length such that γ is a subpath of $\mathcal{L}_{\mathsf{c}}\gamma'$.

One could think of defining a residual as a γ' such that $\mathcal{L}_{\mathsf{c}}\gamma' = \gamma$; unfortunately this doesn't work because of examples like $\gamma = a_0^+$ in the axiom cut reduction case (p. 115); clearly γ has a single residual $a_0^+ = a_1^+$ but $\mathcal{L}_{\mathsf{c}}a_0^+ = a_1^+ a_{\mathsf{c}}^- a_0^+ \neq \gamma$. We will denote $pathresiduals_{\mathsf{c}}(\gamma)$ the set of residuals of γ and by $\gamma \rhd_{\mathsf{c}} \gamma'$ for $\gamma' \in \mathsf{Res}_{\mathsf{c}}(\gamma)$.

The definition of residual is extended to sets of paths: the residual of the set Γ of paths in $\mathcal{R}$ is the union of the sets of residuals of each $\gamma \in \Gamma$, that is $\mathsf{Res}_{\mathsf{c}}(\Gamma) = \bigcup_{\gamma\in\Gamma} \mathsf{Res}_{\mathsf{c}}(\gamma)$. Note that if Γ is finite, then so is $\mathsf{Res}_{\mathsf{c}}(\Gamma)$, althought it might be bigger, thanks to the contraction reduction.

The most important point is that a path may have no residual along the reduction of c in the two following cases:

- In the multiplicative cut case (p. 116) let $\gamma = a_i^+ a_{\mathsf{c}}^+ {a'_{\mathsf{c}}}^- {a'_j}^-$ (assuming the same notations as above). Then if $i \neq j$, γ has no residual along the c cut elimination.
- In the contraction cut case (p. 118) let $\gamma = a_i^+ a_{\mathsf{c}}^+ {a'_{\mathsf{c}}}^- \gamma_0 {a'_{\mathsf{c}}}^+ a_{\mathsf{c}}^- a_j^-$, where γ_0 is a path entirely contained in b', starting and ending in the !-node of b'. Then again γ has no residual in $\mathcal{R}'$ when $i \neq j$.

When γ has a subpath of one of the two form above with $i \neq j$ we say that γ **exchange the premises** of the cut c and that the cut c **breaks the path** γ.

There is a third case of a path having no residual: in the weakening cut case (p. 117) when γ is entirely contained in the box b'. This case is not of the same nature as γ is erased by the weakening cut together with the box b', whereas in the exchange of premises cases γ is disconnected by the cut elimination. For this reason the whole theory of paths is carried in the context of *strict reductions* (see p. 128).

Note that one shouldn't confuse between having no residuals and having only empty paths as residuals. In particular, except in the weakening case, an empty path always have at least one residual (which may be nonempty).

The definition of residuals is extended to many steps reductions in the natural way: if $\rho = \rho_0 c$ is a reduction of a proof net $\mathcal{R}$ beginning with the sequence of steps ρ_0 and ending by reducing a cut c in the ρ_0-retract $\mathcal{R}_0$, then the residuals of any set of paths Γ is $\mathsf{Res}_\rho(\Gamma) = \mathsf{Res}_{\mathsf{c}}(\mathsf{Res}_{\rho_0}(\Gamma))$ and we write $\gamma \rhd_\rho \gamma'$ for $\gamma' \in \mathsf{Res}_\rho(\gamma\}$.

Lemma 4.4.6. *Let $\mathcal{R}$ be a proof net, $\mathcal{R}'$ be obtained from $\mathcal{R}$ by a sequence of reduction ρ, $\mathcal{R}''$ be obtained from $\mathcal{R}'$ by a sequence of reduction σ. Suppose γ and γ'' are paths in $\mathcal{R}$ and $\mathcal{R}''$. Then $\gamma \rhd_{\rho\sigma} \gamma''$ iff there is path γ' in $\mathcal{R}'$ such that $\gamma \rhd_\rho \gamma' \rhd_\sigma$*
$gamma''$.

Proof. Suppose firstly that $\gamma \rhd_{\rho\sigma} \gamma''$. By definition this means that γ'' is minimal such that $\mathcal{L}_{\rho\sigma}\gamma'' \sqsupset \gamma$. But $\mathcal{L}_{\rho\sigma}\gamma'' = \mathcal{L}_\rho(\mathcal{L}_\sigma\gamma'')$ thus there is a minimal

subpath γ' of $\mathcal{L}_\sigma\gamma''$ such that $\mathcal{L}_\rho\gamma' \sqsupset \gamma$. In particular we have $\gamma \rhd_\rho \gamma'$. Suppose $\delta'' \sqsubset \gamma''$ is such that $\mathcal{L}_\sigma\delta'' \sqsupset \gamma'$. Then again by functoriality $\mathcal{L}_{\rho\sigma}\delta'' = \mathcal{L}_\rho(\mathcal{L}_\sigma\delta'') \sqsupset \mathcal{L}_\rho\gamma' \sqsupset \gamma$ so that by minimality of γ'' we must have $\delta'' = \gamma''$. This shows that $\gamma' \rhd_\sigma \gamma''$.

Conversely suppose $\gamma \rhd_\rho \gamma' \rhd_\sigma \gamma''$. We thus have:

- γ'' is minimal such that $\mathcal{L}_\sigma\gamma'' \sqsupset \gamma'$;
- γ' is minimal such that $\mathcal{L}_\rho\gamma' \sqsupset \gamma$.

By functoriality we get that $\mathcal{L}_{\rho\sigma}\gamma'' \sqsupset \gamma$. Let $\delta'' \sqsubset \gamma''$ be such that $\mathcal{L}_{\rho\sigma}\delta'' \sqsupset \gamma$.

If γ' and $\mathcal{L}_\sigma\delta''$ are completely disjoint then let γ''_0 and γ''_1 be subpaths of γ'' such that $\gamma'' = \gamma''_0\delta''\gamma''_1$. Then $\mathcal{L}_\sigma\gamma'' = \mathcal{L}_\sigma(\gamma''_0)\mathcal{L}_\sigma(\delta'')\mathcal{L}_\sigma(\gamma''_1)$, and the subpath γ' must lie in $\mathcal{L}_\sigma\gamma''_i$ for some i. This contradicts the minimality hypothesis for γ''.

Thus γ' and $\mathcal{L}_\sigma\delta''$ overlap, that is there are subpaths γ'_0, γ'_1 and γ'_2 of $\mathcal{L}_\sigma\gamma''$ such that (up to path reversal) $\mathcal{L}_\sigma\delta'' = \gamma'_0\gamma'_1$ and $\gamma' = \gamma'_1\gamma'_2$.

Then $\gamma \sqsubset \mathcal{L}_\rho(\gamma'_0\gamma'_1) = \mathcal{L}_\rho(\gamma'_0)\mathcal{L}_\rho(\gamma'_1)$ and $\gamma \sqsubset \mathcal{L}_\rho(\gamma'_1\gamma'_2) = \mathcal{L}_\rho(\gamma'_1)\mathcal{L}_\rho(\gamma'_2)$ which entails that $\gamma \sqsubset \mathcal{L}_\rho\gamma'_1$. By minimality of γ' we deduce that $\gamma' = \gamma'_1$ thus that $\gamma' \sqsubset \mathcal{L}_\sigma\delta''$. By minimality of γ'' if follows that $\delta'' = \gamma''$ □

4.4.5.4 Reductions of proof nets with paths

A path γ is said to *cross* a cut c if it has a subpath of the form $a^+_\mathsf{c}a'^-_\mathsf{c}$ where a_c and a'_c are the two distinct arrow premises of c.

If Γ is a set of paths in a proof net $\mathcal{R}$ we call a sequence ρ of reductions of $\mathcal{R}$ a **reduction of the proof net with paths** $(\mathcal{R}, \Gamma)$. The result of the reduction is the proof net with paths $(\mathcal{R}', \Gamma')$ where $\mathcal{R}' = \rho(\mathcal{R})$ is the reduced proof net, and $\Gamma' = \mathsf{Res}_\rho(\Gamma)$ is the set of residuals of Γ in $\mathcal{R}'$. If furthermore each cut reduced along the reduction ρ is crossed by at least one residual of a path in Γ we say that ρ is a reduction of Γ and call ρ a **path reduction**.

Theorem 4.4.7 (Lifting of local confluence)**.** *Let $\mathcal{R}$ be a proof-net with two distinct cuts c_1 and c_2, $\mathcal{R}_1$ and $\mathcal{R}_2$ be the proof-nets obtained by reducing respectively c_1 and c_2, ρ_2 and ρ_1 be the sequence of reductions starting from respectively $\mathcal{R}_1$ and $\mathcal{R}_2$ defined in the proof of local confluence, leading to the same proof-net $\mathcal{R}'$; ρ_2 thus reduces (residuals of) c_2 in $\mathcal{R}_1$, ρ_1 reduces (residuals of) c_1 in $\mathcal{R}_2$.*

If γ is any path in $\mathcal{R}'$ then its liftings along the reduction sequences $c_1\rho_2$ and $c_2\rho_1$ are the same path in $\mathcal{R}$:

$$\mathcal{L}_{c_1\rho_2}\gamma = \mathcal{L}_{c_2\rho_1}\gamma$$

Theorem 4.4.8 (Confluence of reduction of proof nets with paths)**.** *With the same hypothesis as in the lifting local confluence theorem if Γ is a set of paths in $\mathcal{R}$ leading respectively to $(\mathcal{R}', \Gamma')$ and $(\mathcal{R}', \Gamma'')$ by the reduction sequences $c_1\rho_2$ and $c_2\rho_1$ then $\Gamma' = \Gamma''$.*

As a consequence if $(\mathcal{R}, \Gamma)$ *is a proof net with paths,* ρ_1 *and* ρ_2 *any two sequences of reductions of* $(\mathcal{R}, \Gamma)$ *leading respectively to* $(\mathcal{R}_1, \Gamma_1)$ *and* $(\mathcal{R}_2, \Gamma_2)$*, then there are sequences of reductions* ρ_2' *and* ρ_1' *of respectively* (R_1, Γ_1) *and* (R_2, Γ_2) *leading to the same proof net with paths* (R', Γ')*.*

In the particular case where Γ is empty, the theorem just states the usual confluence property of proof net reduction.

Note that the theorem is dealing with reductions of proof nets with paths, not with path reductions. Indeed path reduction is not confluent, not even locally confluent: reducing one of the cut, say c_1, involved in a local confluence diagram may break all the elements of Γ so that Γ has no residual by c_1. Therefore the residual of the other cut, c_2, being crossed by no path cannot be reduced as a reduction of Γ, thus making impossible to close the local confluence diagram.

This failure may be overcomed by *saturating* Γ that is by adding all subpaths of paths in Γ; then it becomes impossible to break all paths in the saturated set because some subpaths crossing c_2 just don't cross c_1. Since the residual of a saturated set is saturated we do have confluence of path reductions on saturated sets as a consequence of confluence of reductions of proof nets with paths.

4.4.6 Strong normalisation for typed proof nets in MELL

We prove SN using the same method as Pagani-Tortora de Falco, only with a slightly simpler measure. This relies on the local confluence of numbered cut elimination.
We must prove local confluence of cut elimination before: from this we leave to the reader the task to check that it induces local confluence for numbered cut elimination.
We obtain confluence in the typed case for free. We then establish confluence in the general case, which is more demanding.
In the chapter on Rel, we then provide another proof of SN, based on the invariance of results of experiments: this is a first motivation to provide invariants of reduction. — Lionel

4.4.6.1 Reductions Steps

A *numbered proof net* is a proof net together with a strictly positive natural number, as well as a strictly natural number associated with each box. All these natural numbers are called *labels* of the numbered proof net. Numbered proof nets will mainly be a tool to prove properties of the normalization of proof nets. We define reduction steps on numbered proof nets, but the corresponding notion for proof nets can simply be obtained by forgetting labels.

- a: $n \mapsto n+1$
- m: $n \mapsto n+1$
- d: $n, m \mapsto n+m+1$

- c: $n, m \mapsto n, m, m$
- w: $n, m \mapsto n$
- p: $n, m, k \mapsto n, m, k$

Lemma 4.4.9 (Preservation of Correctness)**.** *If $\mathcal{R}$ is a proof net and $\mathcal{R} \to \mathcal{R}'$ then $\mathcal{R}'$ is a proof net.*

Proof. **TODO** □

4.4.6.2 Properties

The goal of this section is to prove the convergence of the reduction of proof nets.

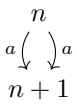
Lemma 4.4.10 (Numbered Congruence)**.** *If $\mathcal{R}$ is a proof net containing $\mathcal{R}_0$ as a sub proof net a depth 0, if $\mathcal{R}_0$ equipped with label m reduces to $\mathcal{R}'_0$ with label m' then $\mathcal{R}$ reduces to $\mathcal{R}'$ where $\mathcal{R}'$ is obtained from $\mathcal{R}$ by replacing $\mathcal{R}_0$ with $\mathcal{R}'_0$ and the label of $\mathcal{R}'$ is $n + m' - m$ (where n is the label of $\mathcal{R}$).*

Proof. **TODO** □

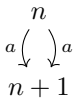
Proposition 4.4.11 (Local Confluence)**.** *The reduction of numbered proof nets is locally confluent.*

Proof.

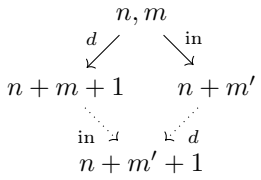
- a/a (shared *cut*)



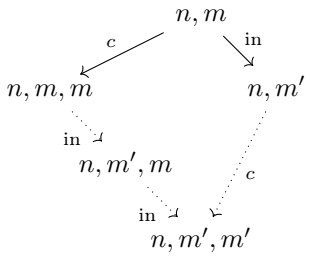
- a/a (shared *ax*)



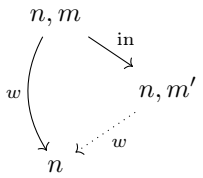
- d/in



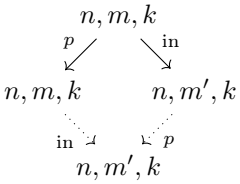
- c/in



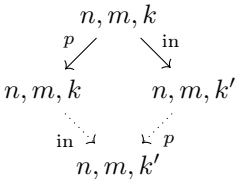
- w/in



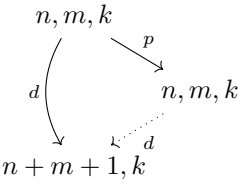
- p/in (left side)



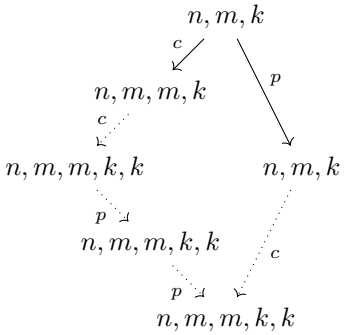
- p/in (right side)



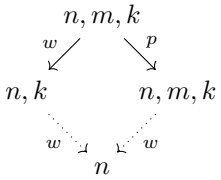
- d/p



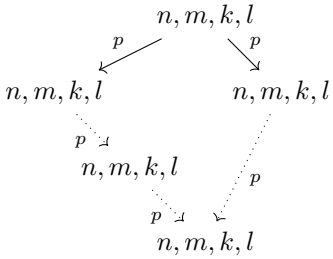
- c/p



- w/p



- p/p



- **TODO** p/p both on auxiliary doors of the same box

TODO □

Proposition 4.4.12 (Weak Normalization)**.** *The reduction of proof nets is weakly normalizing.*

Proof. We define a *size* associated with each cut of a proof net $\mathcal{R}$. It is a pair of natural numbers (s, t) where s is the size of the cut formula (*i.e.* the size of the types of the premises of the *cut* node) and t is the size of the ?-tree above the ? premise of the cut if any, and $t = 0$ otherwise. These pairs are ordered lexicographically. The *cut size* of the proof net $\mathcal{R}$ is the multiset of the sizes of its cuts. Thanks to the multiset ordering, the cut sizes are well ordered.

We now prove that it is always possible to reduce a cut in a proof net $\mathcal{R}$ in a way which makes its size strictly decrease. By Proposition B.3.4, this proves the weak normalization property.

A cut is of *exponential type* if the types of its premises are $!A$ and $?A^\perp$ for some A. Note the source of the premise with type $!A$ of a cut of exponential type must be an ax node or an ! node.

- If $\mathcal{R}$ contains an a redex for which the cut is not of exponential type, we reduce it. A cut disappears and the sizes of the other cuts are not modified.
- If $\mathcal{R}$ contains an m redex, we reduce it. If $A \otimes B$ and $A^\perp \parr B^\perp$ are the types of the premises of the cut, we replace a cut of size $(s_A + s_B + 1, 0)$ by two cuts of sizes $(s_A, \_)$ and $(s_B, \_)$ (and the sizes of the other cuts are not modified), thus the cut size of the proof net strictly decreases.
- If $\mathcal{R}$ has only cuts of exponential types, we consider the following relation on cuts: $c \prec c'$ if one of the following two properties holds:
  - The $!A$ premise of c has an ax node as source and there is a descent path from the $?A^\perp$ conclusion of this ax node to c'.
  - The $!A$ premise of c has an ! node with box $\mathcal{B}$ as source and there is a descent path from an auxiliary door of $\mathcal{B}$ to c'.

  We are going to show that $\prec$ is an acyclic relation on the cuts of $\mathcal{R}$. Let us consider a minimal cycle $c_0 \prec c_1 \prec \cdots \prec c_n$ with $n > 0$ and $c_n = c_0$, it induces a path in $\mathcal{R}$ (enriched with the edges from the main door of each box to its auxiliary doors): from each c_i we go to the ? premise of c_{i+1} by going to the ! premise of c_i reaching the main door of the box $\mathcal{B}_i$ (or an ax node) then we go to an auxiliary door of $\mathcal{B}_i$ (or to the ? conclusion of the ax node) and we follow the descent path until the ? premise of c_{i+1} (we cannot reach its ! premise since descent paths stop when going down on the premise of an ! node). In the case of a minimal cycle, the induced path is a simple path, and all the cuts under consideration must have the same depth since the depth always decreases along the $\prec$ relation. Moreover each $\parr$ node is crossed from one of its premises to its conclusion. By considering a switching graph which contains all the c_i's (they live in the same boxes) and which connects the $\parr$ nodes of the path with the premise contained in the path, we would obtain a cycle which contradicts the acyclicity of the proof net.

  Let us now consider the set $\mathcal{C}$ of all cuts which are maximal for the $\prec$ relation (it is finite and not empty since the set of cuts is finite and the relation $\prec$ is acyclic), and let c be a cut of $\mathcal{C}$ of maximal depth, we reduce c. The reduction of c does not modify the size of any other cut since:
  - If c is maximal for $\prec$, has a box $\mathcal{B}$ above its ! premise, then any cut in $\mathcal{B}$ which is maximal for $\prec$ is maximal in $\mathcal{R}$, so if there is a cut in $\mathcal{B}$ there is a maximal cut in $\mathcal{B}$ for $\prec$ with bigger depth than c (this contradicts the choice of c, thus the content of $\mathcal{B}$ is cut free).
  - The reduction of c does not modify the type of any other cut.

– The reduction of c can only modify the ?-trees of cuts c' such that $c \prec c'$ (and there is no such c' thanks to the choice of c).

If the reduction step is an a or w step, a cut disappears, thus the cut size strictly decreases. If the reduction step is a d step, a cut of size $(s+1, 1)$ is replaced by a cut of size $(s, \_)$, thus the cut size strictly decreases. If the reduction step is a c or p step, a cut of size (s, t) is replaced by 2 or 1 cut(s) of size(s) (s, t') with $t' < t$, thus the cut size strictly decreases.

□

We define some sub-reduction relations:

- The $\xrightarrow{\text{am}}$ reduction is the reduction of proof nets obtained by considering only $\xrightarrow{\text{a}}$ and $\xrightarrow{\text{m}}$ steps.
- The $\xrightarrow{\not w}$ reduction, also called **strict reduction**, is the reduction of proof nets restricted to non w steps.
- The $\xrightarrow{\not c}$ reduction is the reduction of proof nets restricted to non c steps.

Lemma 4.4.13 (Strong *am* Normalization)**.** *The* $\xrightarrow{\text{am}}$ *reduction of proof nets is strongly normalizing.*

Proof. We use Proposition B.3.4, since the number of nodes of proof nets is strictly decreasing along an a or m reduction step. □

Lemma 4.4.14 (Strong w Normalization)**.** *The* $\xrightarrow{\text{w}}$ *reduction of proof nets is strongly normalizing.*

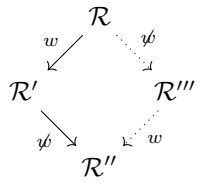
Proof. We use Proposition B.3.4, since the number of nodes of proof nets is strictly decreasing along a w reduction step. □

Lemma 4.4.15 (Sub-Commutation of *am* and non c)**.** *The reduction relations* $\xrightarrow{\text{am}}$ *and* $\xrightarrow{\not c}$ *sub-commute.*

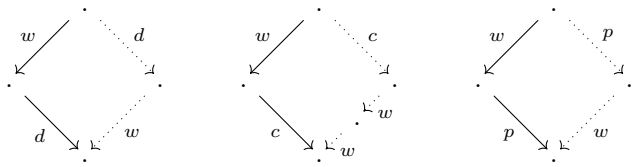
Proof. This easily comes by looking at the proof of Proposition 4.4.11. □

Lemma 4.4.16 (Quasi-Commutation of w over non w)**.** *The* $\xrightarrow{\text{w}}$ *reduction of proof nets quasi-commutes over the* $\xrightarrow{\not w}$ *reduction.*

Proof. Assume we have $\mathcal{R} \xrightarrow{\text{w}} \mathcal{R}' \xrightarrow{\not w} \mathcal{R}''$. If the $\xrightarrow{\text{w}}$ and the $\xrightarrow{\not w}$ steps do not overlap, we directly have commutation and by first applying the $\xrightarrow{\not w}$ step, one obtains $\mathcal{R} \xrightarrow{\not w} \mathcal{R}''' \xrightarrow{\text{w}} \mathcal{R}''$.



The only possible overlapping is when the $\xrightarrow{\not w}$ step acts on a box containing the $\xrightarrow{w}$ step, but then by looking at the _/in cases of the proof of Proposition 4.4.11, we can see we can close the diagrams in an appropriate way:



□

Lemma 4.4.17 (Weak non w Normalization)**.** *The $\xrightarrow{\not w}$ reduction of numbered proof nets is weakly normalizing.*

Proof. We can use the same proof as for Proposition 4.4.12, by using the following remarks.

We consider the *non w cut size* of a proof net to be the multiset of the sizes of the non w cuts of $\mathcal{R}$.

Reducing a cut of non exponential type makes the non w cut size strictly decrease.

If $c \prec c'$ (for the $\prec$ relation of the proof of Proposition 4.4.12) then c' cannot be a w cut. Thus if there are non w cuts in $\mathcal{R}$, the set $\mathcal{C}$ contains non w cuts. We now choose c to be of maximal depth among the non w elements of $\mathcal{C}$, and we reduce c. The only difference with the proof of Proposition 4.4.12 is that the box above c might contain some w cuts. We then see that the non w cut size strictly decreases. □

Lemma 4.4.18 (Increasing non w Reduction)**.** *The reduction $\xrightarrow{\not w}$ on numbered proof nets is μ-increasing where, for a numbered proof net $\mathcal{R}$, $\mu(\mathcal{R}) = l^2+p$ with:*

- *l is the sum of all the labels of $\mathcal{R}$,*
- *p is the sum of the depths of the boxes of $\mathcal{R}$.*

Proof. We analyse each non w step $\mathcal{R} \xrightarrow{\not w} \mathcal{R}'$, we note l' the sum of the labels of $\mathcal{R}'$ and p' the sum of the depths of the boxes of $\mathcal{R}'$.

- a: $\mu(\mathcal{R}') > \mu(\mathcal{R})$ ($l' = l + 1$ and $p' = p$).
- m: $\mu(\mathcal{R}') > \mu(\mathcal{R})$ ($l' = l + 1$ and $p' = p$).
- d: Let n be the label at the current depth in $\mathcal{R}$ and the same for n' in $\mathcal{R}'$, if m is the label of the box, we have $n' = n + m + 1$ and the other labels are not modified thus $l' = l + 1$. Let D be the depth of $\mathcal{R}$, the depth of the opened box is at most D. Let B be the number of boxes in $\mathcal{R}$, there are at most $B - 1$ boxes inside the opened box in $\mathcal{R}$. The opened box disappears, the depth of the boxes inside it decreases by 1, and the depth

of the other boxes is not modified. We thus have $p' \geq p - D - (B-1)$. Since all the labels are strictly positive numbers, we have $l > B \geq D$. We can deduce:

$$\mu(\mathcal{R}') = l'^2 + p' > (l+1)^2 + p - 2l = l^2 + 2l + 1 + p - 2l = \mu(\mathcal{R}) + 1$$

- c: The label of the duplicated box is duplicated (as well as for the labels of all the boxes included in it) and the other labels are not modified thus $l' > l$. New boxes are created (the duplicated one and the new boxes in the copy) and the depth of the other boxes is not modified thus $p' \geq p$ and $\mu(\mathcal{R}') > \mu(\mathcal{R})$.
- p: We have $l' = l$. The depth of the right box, as well as the depth of all the boxes included in it, increases by 1. The depth of all the other boxes is not modified thus $p' > p$ and $\mu(\mathcal{R}') > \mu(\mathcal{R})$.

□

Theorem 4.4.19 (Convergence)**.** *The reduction of proof nets is* convergent.

Proof. We first prove the strong normalization of the $\xrightarrow{\not w}$ reduction by means of Proposition B.3.7: we have Lemmas 4.4.18 and 4.4.17, and we can check in the proof of Proposition 4.4.11 that diagrams with $a \xrightarrow{\not w} b$ and $a \xrightarrow{\not w} c$ can be closed into $b \xrightarrow{\not w}{}^{*} d$ and $c \xrightarrow{\not w}{}^{*} d$ (that is there is no need for w steps in closing the diagram).

We now apply Proposition B.5.1 to $\xrightarrow{w}$ and $\xrightarrow{\not w}$, using Lemmas 4.4.16 and 4.4.14 to obtain strong normalization.

We conclude with confluence by Newman's Lemma (Proposition B.3.6) using Proposition 4.4.11. □

4.4.7 Generalized ? Nodes

We now consider a modified syntax for the exponential connectives in proof nets. The goal is to make more canonical the representation of ?-trees in proof nets. We want a syntax able to realize the fact that the differences between the following ?-trees do not matter:

$$\dfrac{\dfrac{?A \quad ?A}{?A} \quad \dfrac{?A \quad ?A}{?A}}{?A} \quad \text{vs} \quad \dfrac{\dfrac{?A \quad ?A}{?A} \quad \dfrac{?A \quad ?A}{?A}}{?A}$$

$$?A \quad \text{vs} \quad \dfrac{?A \quad \overline{?A}}{?A} \quad \text{vs} \quad \dfrac{\overline{?A} \quad ?A}{?A}$$

TODO

Among the different kinds of nodes we used for exponential proof nets, we replace d, c, w and p nodes by two new kinds of nodes:

- Nodes labelled p have exactly one premise and one conclusion. The label of the premise is the same as the label of the conclusion.
- Nodes labelled ? have an arbitrary number $n \geq 0$ of premises and one conclusion. The labels of the premises are the same formula A and the label of the conclusion is $?A$.

In a proof structure, we add the constraint that a p node must be above a p node or above a ? node. In particular it cannot be above a conclusion node.

It is not possible to represent arbitrary proofs of the sequent calculus MELL in this new syntax. We need the slight restriction that the principal connectives of the formulas introduced by (ax) rules is not ? or !. Note however there is an easy transformation of proofs ensuring this property:

$$\frac{}{\vdash !A, ?A^{\perp}}\,(ax) \qquad \mapsto \qquad \frac{\dfrac{\dfrac{}{\vdash A, A^{\perp}}\,(ax)}{\vdash A, ?A^{\perp}}\,(?)}{\vdash !A, ?A^{\perp}}\,(!)$$

This is an instance of the general notion of axioms expansion of proofs of MELL.

Instead of translating sequent calculus proofs, we will define a translation of the previous proof nets (with ax nodes not introducing formulas with principal connective ? or !) into the new syntax.

translation $(.)^{?}$ from proof nets to proof nets with ? nodes (just for information): replace maximal ?-trees by a ? node with chains of p nodes above it

correctness

reduction

translation $(.)^{cw}$ into proof nets: use degenerate binary trees (left comb trees)

Lemma 4.4.20 (Translation of Correctness)**.** *Let $\mathcal{S}$ be a proof structure with ? nodes, $\mathcal{S}$ is acyclic if and only if $\mathcal{S}^{cw}$ is acyclic.*

Proof. **TODO** □

Proposition 4.4.21 (Simulation)**.** *The translation $(.)^{cw}$ is an injective strict simulation which preserves normal forms from proof nets with ? nodes into proof nets.*

Proof. **TODO** □

Lemma 4.4.22 (Preservation of Correctness)**.** *Let $\mathcal{R}$ be a proof net with ? nodes which reduces into $\mathcal{R}'$, $\mathcal{R}'$ is a proof net.*

Proof. By Lemma 4.4.20, $\mathcal{R}^{cw}$ is acyclic. By Proposition 4.4.21, $\mathcal{R}^{cw} \to^{+} \mathcal{R}'^{cw}$, thus by Lemma 4.4.9 $\mathcal{R}'^{cw}$ is acyclic. By Lemma 4.4.20 again, $\mathcal{R}'$ is acyclic. □

Proposition 4.4.23 (Convergence)**.** *The reduction of proof nets with ? nodes is* convergent.

Proof. We have strong normalization by Propositions B.4.1 and 4.4.21 and Theorem 4.4.19.

Concerning confluence, by Proposition B.2.1 and Theorem 4.4.19, proof nets have the unique normal form property. By Propositions B.4.2 and 4.4.21, proof nets with ? nodes have the unique normal form property. By Propositions B.3.5, B.3.4, proof nets with ? nodes are confluent thanks to strong normalization. □

4.5 Translation of the Lambda-Calculus

4.5.1 The Lambda-Calculus inside Linear Logic

Given a denumerable set of λ-variables x, y, ..., the *terms* of the λ-calculus (or λ-*terms*) are:

$$t, u ::= x \mid \lambda x.t \mid t\, u$$

where λ is a binder for x in $\lambda x.t$ and terms are considered up to α-renaming of bound variables.

We assume given a denumerable set of ground types α, β, ... The simple types of the λ-calculus are:

$$\tau, \sigma ::= \alpha \mid \tau \to \sigma$$

Typing judgements are of the shape $\Gamma \vdash t : \tau$ where Γ is a finite partial function from λ-variables to simple types. The typing rules of the simply typed λ-calculus are:

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ (var) \qquad \frac{\Gamma, x : \tau \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \tau \to \sigma}\ (abs) \qquad \frac{\Gamma \vdash t : \tau \to \sigma \qquad \Gamma \vdash u : \tau}{\Gamma \vdash t\, u : \sigma}\ (app)$$

We assume given a bijection $(.)^\bullet$ from the ground types of the simply typed λ-calculus to the atoms of linear logic. We extend it to any simple type by:

$$(\tau \to \sigma)^\bullet = ?\tau^{\bullet\perp} \parr \sigma^\bullet$$

$$L, M ::= X \mid ?L^\perp \parr M$$

4.5.2 Directed Proof Nets

$$D, E ::= X \mid D \parr E \mid ?U$$
$$U, V ::= X^\perp \mid U \otimes V \mid !D$$

$L \subsetneq D$ and $L^\perp \subsetneq U$
with generalized ? nodes: appropriate definition of the orientation of edges
sequentialization: slight generalization of Theorem 4.5.1
mention cut-free correctness

4.5.3 The Translation

into directed proof nets using only sub-formulas of L (or dual) and only D conclusions

4.5.3.1 Definition

Pre-translation $(.)^\circ$

$$\dfrac{\dfrac{\dfrac{}{\vdash L^\perp, L}\,(ax)}{\vdash ?L^\perp, L}\,(?)}{\vdash ?\Gamma^\perp, ?L^\perp, L}\,(w)$$

$$\dfrac{\vdash ?\Gamma^\perp, ?L^\perp, M}{\vdash ?\Gamma^\perp, ?L^\perp \parr M}\,(\parr)$$

$$\dfrac{\dfrac{\vdash ?\Gamma^\perp, ?L^\perp \parr M \qquad \dfrac{\dfrac{\vdash ?\Gamma^\perp, L}{\vdash ?\Gamma^\perp, !L}\,(!) \qquad \dfrac{}{\vdash M^\perp, M}\,(ax)}{\vdash ?\Gamma^\perp, !L \otimes M^\perp, M}\,(\otimes)}{\vdash ?\Gamma^\perp, ?\Gamma^\perp, M}\,(cut)}{\vdash ?\Gamma^\perp, M}\,(c)$$

By looking at the proof of Lemma 4.4.15, one can see $\xrightarrow{\text{am}}$ is sub-confluent thus it satisfies the unique normal form property (Proposition B.2.1). Moreover $\xrightarrow{\text{am}}$ is strongly normalizing (Lemma 4.4.13), thus we can define the *multiplicative normal form* $\mathrm{NF}_{am}(\mathcal{R})$ of a proof net $\mathcal{R}$ as its unique $\xrightarrow{\text{am}}$ normal form.

We define the translation $t^\bullet$ of λ-term t by $t^\bullet = \mathrm{NF}_{am}(t^\circ)$.

4.5.3.2 Simulations

Substitution Lemma for $(.)^\circ$

$(.)^\circ$ is a strict simulation of β-reduction

translation $(.)^\bullet$ of a β-redex: $(\lambda y.t)\,u$

$$\dfrac{\vdash ?\Gamma^\perp, ?L^\perp, M \qquad \dfrac{\vdash ?\Gamma^\perp, L}{\vdash ?\Gamma^\perp, !L}\,(!)}{\vdash ?\Gamma^\perp, M}\,(cut)$$

cuts correspond to β-redexes through $(.)^\bullet$

$(.)^\bullet$ is an injective strict simulation of β-reduction which preserves normal forms

convergence of the simply typed λ-calculus

4.5.3.3 Image

We already mentioned that proof nets obtained from λ-term by means of the $(.)^\bullet$:

- only contain edges labelled with sub-formulas of formulas generated by the grammar L (or of their dual),
- and only contain exponential cuts.

One can remark as well that all conclusions are labelled with formulas of the shape L or $?L^\perp$.

Add variables as labels of ?_ formulas

require exactly one non ?_ formula or prove it is necessarily the case in directed LL

A proof net satisfying these three conditions is called a *λ-proof net*.

Theorem 4.5.1 (Sequentialization)**.** *Any λ-proof net is the image of a λ-term through the translation $(.)^\bullet$.*

Proof. **TODO** □

4.5.3.4 Kernel

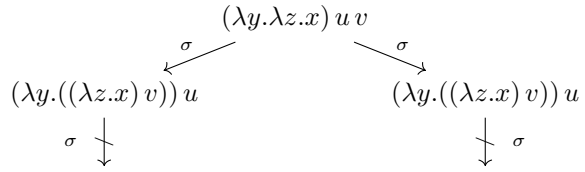
The *σ-reduction* is the congruence on λ-terms generated by:

$$\begin{array}{ll} ((\lambda y.t)\, u)\, v \to_\sigma (\lambda y.(t\, v))\, u & y \notin v \\ (\lambda y.\lambda x.t)\, u \to_\sigma \lambda x.((\lambda y.t)\, u) & x \notin u \end{array}$$

The *σ-equivalence* is the equivalence relation generated by the σ-reduction.

Lemma 4.5.2 (Strong Normalization)**.** *The σ-reduction is strongly normalizing.*

The σ-reduction is not locally confluent, as one can see with the following example:



with $y \notin v$ and $z \notin u$.

A λ-term is called a *canonical form* if it is of the shape:

$$\overrightarrow{\lambda z}.\overrightarrow{\beta(y,u)}.(x\ \overrightarrow{v})$$

where $\beta(y,u).t = (\lambda y.t)\,u$ and all the $\overrightarrow{u}$s and $\overrightarrow{v}$s are themselves canonical forms.

Note that β-normal forms are exactly canonical forms without β-redex.

Lemma 4.5.3 (σ-Normal Forms)**.** *A λ-term is a σ-normal form if and only if it is a canonical form.*

Proof. We prove, by induction on its size, that any λ-term t which is a σ-normal form is a canonical form. We can always write t in a unique way as $t = \overrightarrow{\lambda z}.(\_\ \overrightarrow{v})$ where $\_ = x$ or $\_ = \beta(y,u).t'$. In the first case, t is a canonical form (the $\overrightarrow{v}$s are themselves σ-normal forms thus canonical forms by induction hypothesis). In the second case, by induction hypothesis, t' is a canonical form (and u as well), moreover it does not start with a λ (otherwise we have a σ-redex in t). If the sequence $\overrightarrow{v}$ is not empty, we have a σ-redex in t as well. We can conclude that $t = \overrightarrow{\lambda z}.\beta(y,u).t'$ with u and t' in canonical form and t' not starting with a λ, which makes t a canonical form.

Conversely, there is no σ-redex in a canonical form. □

Theorem 4.5.4 (σ-Equivalence)**.** *Let t and t' be two λ-terms, $t^\bullet = t'^\bullet$ if and only if $t \simeq_\sigma t'$.*

Proof. We start with the second implication by considering the two equations defining the σ-reduction. **TODO**

Concerning the first implication, we can remark that it is enough to prove the result for two canonical forms t and t'. Indeed, assuming this particular case of the result, if u and u' are two arbitrary λ-terms such that $u^\bullet = u'^\bullet$, then by Lemmas 4.5.2 and 4.5.3, there exist two canonical forms $t \simeq_\sigma u$ and $t' \simeq_\sigma u'$ thus $t^\bullet = u^\bullet = u'^\bullet = t'^\bullet$ (using the other direction proved above). This entails $t \simeq_\sigma t'$ and thus $u \simeq_\sigma u'$.

We thus assume t and t' to be two canonical forms such that $t^\bullet = t'^\bullet$.
TODO □

direct/global translation of canonical forms into proof nets

4.5.4 Untyped Lambda-Calculus

The untyped λ-calculus can be seen as the result of quotienting the types of the simply typed λ-calculus by means of an equation $o = o \to o$. Any variable can then be seen as typed with type o and the typing rules become:

$$\frac{}{\Gamma, x : o \vdash x : o}\ (var) \qquad \frac{\Gamma, x : o \vdash t : o}{\Gamma \vdash \lambda x.t : o}\ (abs) \qquad \frac{\Gamma \vdash t : o \quad \Gamma \vdash u : o}{\Gamma \vdash t\,u : o}\ (app)$$

The information provided by these rules is mainly a super-set of the list of free variables of the term.

One can similarly quotient formulas of linear logic by means of the equation $o = !o \multimap o$, that is $o = ?o^\perp \parr o$. This entails that the set of the sub-formulas of

formulas generated from the atom o by the unique construction $?o^\perp \parr o$ and of their dual (up to the quotient) contains four elements: o, $\iota = o^\perp$, $!o$ and $?\iota$. It is then possible to translate λ-terms as proof net with edges labelled with these four formulas.

For example the λ-term $\lambda x.x\,x$ is translated as: `[figure]`

4.6 Further Reading

We suggest an incomplete list of related papers.

4.6.1 Historical Papers

- The original paper on linear logic which introduces proof nets [20]. The correctness criterion used there is the long trip criterion and the proof technique for sequentialization is based on the theory of empires.
- The definition of the acyclic-connected correctness criterion we use here [13].
- The definition of the σ-equivalence on λ-terms [41].

4.6.2 Sequentialization

- A sequentialization proof based on the acyclic-connected criterion and using empires [22].
- [9]
- [6]
- The sequentialization proof we used here [34].

4.6.3 Rewriting Properties

- [44]
- [9]
- [39]

4.6.4 Extensions of the Syntax

- Modules
- Units [8, 30]
- Quantifiers [22]
- Additive connectives [21, 31]

4.6.5 Relations with the Lambda-Calculus

- [41]
- [14]

4.6.6 Complexity

- [27]

Part II

Static models

Chapter 5

Rel

A plan for the future. We start with Rel, presented via experiments, and show that the interpretation is invariant under cut elimination.
We then introduce categorical models, as a way to capture relevant structure in a compositional way. We first restrict to the intuitionistic case, arguing that the interpretation of proofs as morphisms is more natural in this setting. We keep Rel as a running example: we must first explain that it also gives a model of ILL. We delay the introduction of *-autonomous structure until late in the chapter: this allows to recover an interpretation of classical sequents.
In later chapters on concrete models, we then rely on the categorical structure liberally. — Lionel

5.1 Relational interpretation of the sequent calculus: resource derivations

With any formula A of linear logic, we can associate a set $[A]$ of *tokens*, the definition is as follows.

$$[\mathbf{1}] = [\bot] = \{*\} \qquad [A \otimes B] = [A \parr B] = [A] \times [B]$$
$$[0] = [\top] = \emptyset \qquad [A \oplus B] = [A \,\&\, B] = \{1\} \times [A] \cup \{2\} \times [B]$$
$$[!A] = [?A] = \mathcal{M}_{\text{fin}}([A])$$

Let us define a *resource sequent* as a sequent

$$\vdash_{\mathsf{r}} a_1 : A_1, \ldots, a_k : A_k$$

where $A_1, \ldots, A_k$ are formulas and $a_i \in [A_i]$ for each $i \in \{1, \ldots, k\}$. Given a *resource context* $\Phi = (a_1 : A_1, \ldots, a_k : A_k)$, we use $\underline{\Phi}$ for the underlying context $(A_1, \ldots, A_k)$.

Then we introduce a deduction system for these sequents.

$$\frac{a \in [A]}{\vdash_{\mathsf{r}} a : A^{\perp}, a : A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A \quad \vdash_{\mathsf{r}} a : A^{\perp}, \Psi}{\vdash_{\mathsf{r}} \Phi, \Psi}$$

$$\frac{}{\vdash_{\mathsf{r}} * : 1} \qquad \frac{\vdash_{\mathsf{r}} \Phi}{\vdash_{\mathsf{r}} \Phi, * : \bot}$$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A \quad \vdash_{\mathsf{r}} \Psi, b : B}{\vdash_{\mathsf{r}} \Phi, \Psi, (a, b) : A \otimes B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A, b : B}{\vdash_{\mathsf{r}} \Phi, (a, b) : A \parr B}$$

No rule for 0 No rule for $\top$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, (1, a) : A \oplus B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, b : B}{\vdash_{\mathsf{r}} \Phi, (2, b) : A \oplus B}$$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, (1, a) : A \,\&\, B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, b : B}{\vdash_{\mathsf{r}} \Phi, (2, b) : A \,\&\, B}$$

$$\frac{\vdash_{\mathsf{r}} \Phi}{\vdash_{\mathsf{r}} \Phi, [] : ?A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, l : ?A, r : ?A}{\vdash_{\mathsf{r}} \Phi, l + r : ?A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, [a] : ?A}$$

$$\frac{\left(\vdash_{\mathsf{r}} m_1^j : ?A_1, \dots m_k^j : ?A_k, b^j : B\right)_{j=1}^{n}}{\vdash_{\mathsf{r}} \sum_{j=1}^{n} m_1^j : ?A_1, \dots, \sum_{j=1}^{n} m_k^j : ?A_k, [b^1, \dots, b^n] : !B}$$

A derivation θ in this system will be called a *resource derivation*.

Let π be a proof of $\vdash \Gamma$. We define $\mathcal{T}(\pi)$ as a set of resource derivations θ, each of them having a conclusion $\vdash_{\mathsf{r}} \Phi$ such that $\underline{\Phi} = \Gamma$. The set $\mathcal{T}(\pi)$ is defined by induction on π.

If π is an axiom

$$\frac{}{\vdash A^{\perp}, A}$$

then $\mathcal{T}(\pi)$ is the set of all resource axioms

$$\frac{}{\vdash_{\mathsf{r}} a : A^{\perp}, a : A}$$

for $a \in [A]$.

If π is a cut

$$\frac{\begin{array}{cc} \vdots\, \pi_1 & \vdots\, \pi_2 \\ \vdash \Gamma_1, A & \vdash A^{\perp}, \Gamma_2 \end{array}}{\vdash \Gamma_1, \Gamma_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\frac{\begin{array}{cc} \vdots\, \theta_1 & \vdots\, \theta_2 \\ \vdash_{\mathsf{r}} \Phi_1, a : A & \vdash_{\mathsf{r}} a : A^{\perp}, \Phi_2 \end{array}}{\vdash_{\mathsf{r}} \Phi_1, \Phi_2}$$

where $\theta_i \in \mathcal{T}(\pi_i)$ for $i = 1, 2$. The important constraint on θ_1 and θ_2 is that the corresponding tokens in A and $A^\perp$ are the same (namely a). Notice that this implies $\underline{\Phi_i} = \Gamma_i$ for $i = 1, 2$ and hence $\underline{\Phi_1, \Phi_2} = \Gamma_1, \Gamma_2$.

If π is the proof

$$\frac{}{\vdash 1}$$

then $\mathcal{T}(\pi)$ is the singleton consisting of the proof

$$\frac{}{\vdash_{\mathsf{r}} * : 1}$$

If π is the proof

$$\frac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma\end{array}}{\vdash \Gamma, \perp}$$

then $\mathcal{T}(\pi)$ is set et of all resource derivations

$$\frac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi\end{array}}{\vdash_{\mathsf{r}} \Phi, * : \perp}$$

such that $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\frac{\begin{array}{cc}\vdots\,\pi_1 & \vdots\,\pi_2\\ \vdash \Gamma_1, A_1 & \vdash \Gamma_2, A_2\end{array}}{\vdash \Gamma_1, \Gamma_2, A_1 \otimes A_2}$$

then $\mathcal{T}(\pi)$ is the set of all proofs

$$\frac{\begin{array}{cc}\vdots\,\theta_1 & \vdots\,\theta_2\\ \vdash_{\mathsf{r}} \Phi_1, a_1 : A_1 & \vdash_{\mathsf{r}} \Phi_2, a_2 : A_2\end{array}}{\vdash_{\mathsf{r}} \Phi_1, \Phi_2, (a_1, a_2) : A_1 \otimes A_2}$$

with $\theta_i \in \mathcal{T}(\pi_i)$ for $i = 1, 2$.

If π is the proof

$$\frac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma, A_1, A_2\end{array}}{\vdash \Gamma, A_1 \parr A_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi, a_1 : A_1, a_2 : A_2\end{array}}{\vdash_{\mathsf{r}} \Phi, (a_1, a_2) : A_1 \parr A_2}$$

with $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\dfrac{}{\vdash \Gamma, \top}$$

then $\mathcal{T}(\pi) = \emptyset$.

If $i \in \{1, 2\}$ and π is the proof

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma, A_i\end{array}}{\vdash \Gamma, A_1 \oplus A_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi, a : A_i\end{array}}{\vdash_{\mathsf{r}} \Phi, (i, a) : A_1 \oplus A_2}$$

If π is the proof

$$\dfrac{\begin{array}{cc}\vdots\,\pi_1 & \vdots\,\pi_2\\ \vdash \Gamma, A_1 & \vdash \Gamma, A_2\end{array}}{\vdash \Gamma, A_1 \mathbin{\&} A_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\,\theta\\ \vdash_{\mathsf{r}} \Phi, a : A_i\end{array}}{\vdash_{\mathsf{r}} \Phi, (i, a) : A_1 \mathbin{\&} A_2}$$

for $i \in \{1, 2\}$ and $\theta \in \mathcal{T}(\pi_i)$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma\end{array}}{\vdash \Gamma, ?A}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi\end{array}}{\vdash_{\mathsf{r}} \Phi, [] : ?A}$$

for $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\ \pi_1\\ \vdash \Gamma, ?A, ?A\end{array}}{\vdash \Gamma, ?A}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\ \theta_1\\ \vdash_{\mathsf{r}} \Phi, l : ?A, r : ?A\end{array}}{\vdash_{\mathsf{r}} \Phi, l + r : ?A}$$

for $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\ \pi_1\\ \vdash \Gamma, A\end{array}}{\vdash \Gamma, ?A}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\ \theta_1\\ \vdash_{\mathsf{r}} \Phi, a : A\end{array}}{\vdash_{\mathsf{r}} \Phi, [a] : ?A}$$

for $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\ \pi_1\\ \vdash ?A_1, \ldots, ?A_k, B\end{array}}{\vdash ?A_1, \ldots, ?A_k, !B}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{ccc}\vdots\ \theta_1 & & \vdots\ \theta_n\\ \vdash_{\mathsf{r}} m_1^1 : ?A_1, \ldots, m_k^1 : ?A_k, b^1 : B & \cdots & \vdash_{\mathsf{r}} m_1^n : ?A_1, \ldots, m_k^n : ?A_k, b^n : B\end{array}}{\vdash_{\mathsf{r}} \sum_{j=1}^n m_1^j : ?A_1, \ldots, \sum_{j=1}^n m_k^j : ?A_k, [b^1, \ldots, b^n] : !B}$$

for all $n \in \mathbf{N}$ and $\theta_1, \ldots, \theta_n \in \mathcal{T}(\pi_1)$.

The relational interpretation $[\pi]$ of a proof π of $\vdash A_1, \ldots, A_k$ is the set of all tuples $(a_1, \ldots, a_k) \in \prod_{i=1}^k [A_i]$ such that there is a resource derivation $\theta \in \mathcal{T}(\pi)$ of the resource sequent $\vdash_{\mathsf{r}} a_1 : A_1, \ldots, a_k : A_k$.

One can prove that if π reduces to π' by cut-elimination then $[\pi] = [\pi']$. We will present now the same relational semantics, but in a categorical way.

5.2 The relational model as a category

We introduce now the simplest (and perhaps most fundamental) $*$-autonomous category equipped with an exponential structure: the category of sets and relations.

Let **Rel** be the category whose objects are sets and where $\mathbf{Rel}(X,Y) = \mathcal{P}(X \times Y)$, identities being the diagonal relations and composition being defined as follows: if $R \in \mathbf{Rel}(X,Y)$ and $S \in \mathbf{Rel}(Y,Z)$ then

$$S\,R = \{(a,c) \in X \times Z \mid \exists b \in Y\ (a,b) \in R \text{ and } (b,c) \in S\}\,.$$

Let $x \subseteq X$, we set $R \cdot x = \{b \in Y \mid \exists a \in x\ (a,b) \in R\} \subseteq Y$ which is the direct image of x by R. We also define $R^{\perp} = \{(b,a) \in Y \times X \mid (a,b) \in Y\}$ which is the transpose of R (considering R as a matrix with coefficients in $\{0,1\}$). Given $x \subseteq X$ and $y \subseteq Y$, we have

$$(R \cdot x) \cap y = \mathsf{pr}_2(R \cap (x \times y)) \quad \text{and} \quad (R^{\perp} \cdot y) \cap x = \mathsf{pr}_1(R \cap (x \times y)) \tag{5.1}$$

where pr_1 and pr_2 are the two projections of the cartesian product in the category **Set** of sets and functions (the ordinary cartesian product "$\times$").

Lemma 5.2.1. *An isomorphism in* **Rel** *is a relation which is a bijection.*

We consider the symmetric monoidal structure on **Rel** given by the tensor product $X \otimes Y = X \times Y$ and the unit 1 an arbitrary singleton $\{*\}$. The neutrality, associativity and symmetry isomorphisms are defined as the obvious corresponding bijections (for instance, the symmetry isomorphism $\sigma_{X,Y} \in \mathbf{Rel}(X \otimes Y, Y \otimes X)$ is given by the bijection $(a,b) \mapsto (b,a)$). This symmetric monoidal category is closed, with linear function space given by $X \multimap Y = X \times Y$, the natural bijection between $\mathbf{Rel}(Z \otimes X, Y)$ and $\mathbf{Rel}(Z, X \multimap Y)$ being induced by the cartesian product associativity isomorphism. Last, one takes for $\perp$ an arbitrary singleton, and this turns **Rel** into a $*$-autonomous category. One denotes as $\star$ the unique element of 1 and $\perp$. Then, up to natural isomorphism, $X^{\perp} = X$.

This category is cartesian, with cartesian product $X_1 \,\&\, X_2$ of X_1 and X_2 defined as $\{1\} \times X_1 \cup \{2\} \times X_2$ with projections $\mathsf{pr}_i = \{((i,a),a) \mid a \in X_i\}$ (for $i = 1,2$), and terminal object $\top = \emptyset$. Given morphisms $R_i \in \mathbf{Rel}(Y, X_1, X_2)$, their cartesian pairing $\langle R_1, R_2\rangle \in \mathbf{Rel}(Y, X_1 \,\&\, X_2)$ is $\langle R_1, R_2\rangle = \{(b,(i,a)) \mid (b,a) \in R_i \text{ for } i = 1,2\}$.

Rel is also a Seely category (see Section 6.1), for a comonad $!\_$ defined as follows:

- $!X$ is the set of all finite multisets of elements of X;
- if $R \in \mathbf{Rel}(X,Y)$, then we set $!R = \{([a_1,\dots,a_n],[b_1,\dots,b_n]) \mid n \in \mathbf{N} \text{ and } \forall i\,(a_i,b_i) \in R\}$;
- $\mathsf{d}_X \in \mathbf{Rel}(!X, X)$ is $\mathsf{d}_X = \{([a],a) \mid a \in X\}$;

- $\mathsf{p}_X = \{(m_1 + \cdots + m_n, [m_1, \ldots, m_n]) \mid n \in \mathbf{N} \text{ and } m_1, \ldots, m_n \in !X\}$.

The monoidality isomorphism $\mathsf{m}^2_{X,Y} \in \mathbf{Rel}(!X \otimes !Y, !(X \,\&\, Y))$ is the bijection which maps $([a_1, \ldots, a_l], [b_1, \ldots, b_r])$ to $[(1, a_1), \ldots, (1, a_l), (2, b_1), \ldots, (2, b_r)]$. And m^0 is the obvious bijection from $1 = \{*\}$ to $!\top = \{[]\}$.

Then the associated structural morphisms $\mathsf{w}_X \in \mathbf{Rel}(!X, 1)$ and $\mathsf{c}_X \in \mathcal{L}(!X, !X \otimes !X)$ are

$$\begin{aligned} \mathsf{w}_X &= \{([], *)\} \\ \mathsf{c}_X &= \{(m, (m_1, m_2)) \mid m = m_1 + m_2\}. \end{aligned}$$

The induced lax monoidal structure ($\mu^0 \in \mathcal{L}(1, !1)$ and $\mu^2_{X,Y} \in \mathcal{L}(!X \otimes !Y, !(X \otimes Y))$) of $!\_$ is

$$\begin{aligned} \mu^0 &= \{(*, k[*]) \mid k \in \mathbf{N}\} \\ \mu^2_{X,Y} &= \{(([a_1, \ldots, a_n], [b_1, \ldots, b_n]), [(a_1, b_1), \ldots, (a_n, b_n)]) \\ &\qquad \mid a_1, \ldots, a_n \in X \text{ and } b_1, \ldots, b_n \in Y\}. \end{aligned}$$

If $R \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_k, Y)$ then the generalized promotion $R^! \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_k, !Y)$ of R is

$$\begin{aligned} R^! = \{(m^1_1 + \cdots + m^1_n, \ldots, m^k_1 + \cdots + m^k_n, [b_1, \ldots, b_n]) \mid \\ \forall i \in \{1, \ldots, n\}\ (m^1_i, \ldots, m^k_i, b_i) \in R\}. \end{aligned}$$

Applying the general interpretation of Section **??** and **??**, we can interpret any proof structure p such that $\vdash p : A_1, \ldots, A_k$ as an element $[p]$ of $\mathbf{Rel}(1, [A_1] \parr \cdots \parr [A_k]) \simeq \mathcal{P}([A_1] \times \cdots \times [A_k])$ where the interpretation of formulas is also defined in Section **??**; here: $[1] = [\bot] = \{*\}$, $[A \otimes B] = [A \parr B] = [A] \times [B]$ and $[!A] = [?A] = \mathcal{M}_{\mathrm{fin}}([A])$.

Fix references — Laurent

Fix reference — Laurent

Remark 5.2.2. This model is often presented as "degenerate", the main reason for this is that it makes no difference between the interpretation of a type (formula) and of its linear negation (thus identifying $\parr$ and $\otimes$, ? and !, & and $\oplus$ when additive connectives are taken into account). This however does not mean that the interpretation of proofs is degenerate. It is in some sense quite the contrary: as shown by De Carvalho **[DeCarvalho]**, Guerrieri and Tortora de Falco **[GuerrieriTortora]**, two cut-free proof-nets which have the same interpretation in **Rel** are "essentially" equal (that is equal up to the equivalence on proofs induced by Rétoré's reduction relation, including the fact that w is neutral for c and that c is associative and commutative).

However, one main weakeness of **Rel** is that types are interpreted as unstructured sets: the interpretation of a type A does not tell us anything about the specific subsets of $[A]$ which occur as interpretations of proof-nets p such that $\vdash p : A$. This was not the case of the original *coherence space*[1] model

[1] Of which many excellent presentations can be found in the litterature, startingof course with **[Girard]**.

discovered by Girard [**Girard**]. A coherence space is a structure $E = (|E|, \frown_E)$ where $|E|$ is a set (which can be assumed to be at most countable, it is called the *web* of E) and $\frown_E$ is a binary symmetric and reflexive relation on $|E|$ which express when two elements of $|E|$ can be put together to form a "piece of data" of E: these pieces of data are the *cliques* of E and one interprets LL in this model, a formula A is interpreted as a coherence space E and a proof-net p such that $\vdash p : A$ as a clique of E. This semantics has not the same "degeneracy" as **Rel** because the coherence space $E^\perp$ interpreting $A^\perp$ has the same web as E, but $a \frown_{E^\perp} a'$ if $a = a'$ or $\neg(a \frown_E a')$; this model provides us with non trivial information about proof interpretations. For instance, in coherence spaces, the only cliques of $1 \oplus 1$ are $\{(1, *)\}$, $\{(2, *)\}$ and $\emptyset$ which are the two usual boolean values and the undefined one.

Intuitively **Rel** is like the coherence space model, just forgetting the coherence relation and interpreting any type as the web of its interpretation in coherence spaces. Unfortunately the picture is not that simple because the web of the coherence space $!E$ interpreting $!A$ (when E is the coherence space interpretaing A) has the set of *finite cliques*[2] of E as web and not the set of finite multisets of elements of $|E|$. As shown in [**BucciarelliEhrhard**] this issue can be solved and one can design a *non uniform coherence space* model of LL which has the following properties which relate it strongly to the **Rel** model:

- any type A is interpreted by a non uniform coherence E such that $|E|$ is exactly the interpretation of A in **Rel**
- and the interpretation in this model of a proof-net p such that $\vdash p : A$ is a clique of E which, as a subset of $|E|$, coincides with the interpretation of p in **Rel**.

So the only job of this model is to sort out, among all subsets of the interpretation of A in **Rel**, some particularly well behaved ones among which the interpretation of proofs of A appear. We present now this model.

5.3 Enriching the relational model with a (non-uniform) coherence structure

A (non-uniform) coherence space[3] is a structure

$$E = (|E|, \frown_E, \smile_E)$$

where $|E|$ is a set (which can be assumed to be at most countable) and $\frown_E$ and $\smile_E$ are *disjoint* binary, symmetric and antireflexive relations on $|E|$ called *strict coherence* and *strict incoherence* respectively. The binary relation ν_E on

[2] Or finite *multi-cliques* which are multisets whose supports are cliques, as observed first by Van de Wiel and then by Lafont [**Lfont**]; indeed one obtains in that way a nice example of the concept of Lafont category of Section 6.1.5.

[3] We drop the "non-uniform" in the sequel.

$|E|$ which is the complementary set of $\frown_E \cup \smile_E$ is called *neutrality*, it is clearly symmetric, but usually, it is neither reflexive nor anti-reflexive[4].

We use the following notations: $\subset_E = \frown_E \cup \nu_E$ (large coherence) and $\asymp_E = \smile_E \cup \nu_E$ (large incoherence) which are symmetric relations on $|E|$. Notice that a coherence space can be specified by providing any pair of relations among the 7 following ones, satisfying the folowing conditions:

- $\frown_E$ and $\smile_E$ such that $\frown_E \cap \smile_E = \emptyset$;
- $\subset_E$ and $\frown_E$, two symmetric relations such that $\frown_E \subseteq \subset_E$ and then $\smile_E = |E|^2 \setminus \subset_E$;
- $\subset_E$ and ν_E with $\nu_E \subseteq \subset_E$ and then $\frown_E = \subset_E \setminus \nu_E$ and $\smile_E = |E|^2 \setminus \subset_E$;
- $\frown_E$ and ν_E with $\frown_E \cap \nu_E = \emptyset$ and then $\smile_E = |E|^2 \setminus (\frown_E \cup \nu_E)$
- and the duals of the 3 last pairs (replacing coherence with incoherence).

A clique of E is a subset u of $|E|$ such that $\forall a, a' \in u \; a \subset_E a'$ and we use $\mathsf{Cl}(E)$ for the set of cliques of E. The coherence space $E^\perp$ is defined by $|E^\perp| = |E|$, $\frown_{E^\perp} = \smile_E$ and $\smile_{E^\perp} = \frown_E$ so that obviously $E^{\perp\perp} = E$. Notice however that it is no more true that, given $u \in \mathsf{Cl}(E)$ and $u' \in \mathsf{Cl}(E^\perp)$, the set $u \cap u'$ has at most one element as in usual coherence spaces; all we can say *a priori* is that $\forall a, a' \in u \cap u' \; a \; \nu_E \; a'$.

Given coherence spaces E_1 and E_2, one first defines $E_1 \otimes E_2$ by $|E_1 \otimes E_2| = |E_1| \times |E_2|$, $(a_1, a_2) \subset_{E_1 \otimes E_2} (a'_1, a'_2)$ if $a_i \subset_{E_i} a'_i$ for $i = 1, 2$ and $(a_1, a_2) \; \nu_{E_1 \otimes E_2} \; (a'_1, a'_2)$ if $a_i \; \nu_{E_i} \; a'_i$ for $i = 1, 2$.

Then one sets $E \multimap F = (E \otimes F^\perp)^\perp$. In other words, $|E \multimap F| = |E| \times |F|$ and:

- $(a, b) \subset_{E \multimap F} (a', b')$ if $a \subset_E a' \Rightarrow b \subset_F b'$ and $a \frown_E a' \Rightarrow b \frown_F b'$,
- and $(a, b) \; \nu_{E \multimap F} \; (a', b')$ if $a \; \nu_E \; a'$ and $b \; \nu_F \; b'$,

Notice that $(a, b) \subset_{E \multimap F} (a', b')$ is equivalent to $b \; \nu_F \; b' \Rightarrow a \subset_E a'$ and $b \smile_F b' \Rightarrow a \smile_F a'$, a characterization of $\subset_{E \multimap F}$ which will be quite useful when dealing with Boudes' exponential.

The category **NCoh** has coherence spaces as objects, and $\mathbf{NCoh}(E, F) = \mathsf{Cl}(E \multimap F)$. Obviously, the diagonal $Id_E \subseteq |E|^2$ belongs to $\mathbf{NCoh}(E, E)$, it is the identity morphism (defined as in **Rel**). Also if $R \in \mathbf{NCoh}(E, F)$ and $S \in \mathbf{NCoh}(F, G)$, the relational composition $S\,R$ is easily seen to be in $\mathbf{NCoh}(E, G)$: this is the notion of composition we use to define the category **NCoh**.

This category is easily seen to be symmetric monoidal (with the operation $\otimes$ defined above on objects, its extension to morphisms being defined as in **Rel**, the structural isos of SMC being also defined as in **Rel**). The "neutral object" is $1 = (\{*\}, \emptyset, \emptyset)$ (in other words $* \; \nu_1 \; *$). This SMC **NCoh** is closed with $E \multimap F$ as object of morphisms from E to F (and linear evaluation morphism ev, as

[4]This uncoupling of equality and coherence on $|E|$ is the main feature of these *non-uniform* coherence spaces.

well as linear curryfication, defined as in **Rel**). It is also *-autonomous with dualizing object $\bot = 1$, the dual of E being $E^\bot$ (up to trivial iso).

Next, **NCoh** is easily seen to be cartesian, with terminal object $\top = (\emptyset, \emptyset, \emptyset)$ and cartesian product of E_1 and E_2 the coherence space $E_1 \,\&\, E_2$ defined by $|E_1 \,\&\, E_2| = \{1\} \times |E_1| \cup \{2\} \times |E_2|$ and

- $(i, a)\ \nu_{E_1 \& E_2}\ (j, b)$ if $i = j$ and $a\ \nu_{E_i}\ b$
- and $(i, a) \subset_{E_1 \& E_2} (j, b)$ if $i = j \Rightarrow a\ \nu_{E_i}\ b$.

The projection morphisms are defined as in **Rel** and so is the pairing of two morphisms in $\mathbf{NCoh}(F, E_i)$ for $i = 1, 2$.

Concerning the exponential, several definitions are possible, satisfying the requisit that $|!E| = \mathcal{M}_{\rm fin}(|E|) = !|E|$ (this latter exponential being taken that of **Rel**.

5.3.1 Boudes' exponential

The one we want to mention first is the free exponential of **NCoh** (it has been discovered after the second one actually, by Pierre Boudes [**Boudes**], for that reason we denote it as $!_{\mathsf{b}}E$). As already mentioned $!_{\mathsf{b}}E = \mathcal{M}_{\rm fin}(|E|)$. Given $m, m' \in \mathcal{M}_{\rm fin}(|E|)$:

- $m\ \nu_{!_{\mathsf{b}}E}\ m'$ if $m = [a_1, \ldots, a_n]$ and $m' = [a'_1, \ldots, a'_n]$ with $\forall i\ a_i\ \nu_E\ a'_i$ and $\forall i, j\ a_i \subset_E a'_j$
- and $m \smile_{!_{\mathsf{b}}E} m'$ if $\exists a \in m, a' \in m'\ a \smile_E a'$

where "$a \in m$" means $m(a) > 0$.

Given $R \in \mathbf{NCoh}(E, F)$, we prove that $!_{\mathsf{b}}R$, which is defined exactly as $!R$, satisfies $!_{\mathsf{b}}R \in \mathbf{NCoh}(!_{\mathsf{b}}E, !_{\mathsf{b}}F)$. So let $(m^i, p^i) \in !_{\mathsf{b}}R$ for $i = 1, 2$, that is: $m^i = [a^i_1, \ldots, a^i_{n^i}]$ and $p^i = [b^i_1, \ldots, b^i_{n^i}]$ with $(a^i_j, b^i_j) \in R$ for $i = 1, 2$ and $j = 1, \ldots, n^i$. Assume first that $p^1 \smile_{!_{\mathsf{b}}F} p^2$ so that we can find $j^i \in \{1, \ldots, n^i\}$ for $i = 1, 2$ such that $b^1_{j^1} \smile_F b^2_{j^2}$ so that $a^1_{j^1} \smile_E a^2_{j^2}$ because $(a^1_{j^1}, b^1_{j^1}) \subset_{E \multimap F} (a^2_{j^2}, b^2_{j^2})$ since $R \in \mathsf{Cl}(E \multimap F)$, this proves that $m^1 \smile_{!_{\mathsf{b}}E} m^2$. Assume next that $p^1\ \nu_{!_{\mathsf{b}}F}\ p^2$, we contend that $m^1 \asymp_{!_{\mathsf{b}}E} m^2$. We can assume that $n^1 = n^2 = n$, that $b^1_j\ \nu_F\ b^2_j$ for $j = 1, \ldots, n$, and we know that $b^1_{j^1} \subset_F b^2_{j^2}$ for all $j^1, j^2 \in \{1, \ldots, n\}$. Therefore we have $a^1_j \asymp_E a^2_j$ for $j = 1, \ldots, n$ (because $R \in \mathsf{Cl}(E \multimap F)$). If for some $j^1, j^2 \in \{1, \ldots, n\}$ we have $a^1_{j^1} \smile_E a^2_{j^2}$ then $m^1 \smile_{!_{\mathsf{b}}E} m^2$ and our contention holds so assume this is not the case, meaning that $\forall j^1, j^2 \in \{1, \ldots, n\}\ a^1_{j^1} \subset_E a^2_{j^2}$. In particular we have $\forall j \in \{1, \ldots, n\}\ a^1_j\ \nu_E\ a^2_j$ and hence $m^1\ \nu_{!_{\mathsf{b}}E}\ m^2$, proving our contention.

So we have proven that $!_{\mathsf{b}}\_$ is a functor $\mathbf{NCoh} \to \mathbf{NCoh}$, its action on morphisms being defined exactly as in **Rel**. Now we prove that the comonad structure of $!\_$ on **Rel** is actually a structure of comonad for $!_{\mathsf{b}}\_$ on **NCoh**. One has first to check that $\mathsf{d}_E = \mathsf{d}_{|E|} = \{([a], a) \mid a \in |E|\}$ belongs to $\mathbf{NCoh}(!E, E)$

which results from the straightforward observation that

$$\forall a, a' \in |E| \quad [a] \smile_{!_{\mathsf{b}}E} [a'] \Leftrightarrow a \smile_E a' \text{ and } [a] \nu_{!_{\mathsf{b}}E} [a'] \Leftrightarrow a \nu_E a' .$$

Next we have to check that $\mathsf{p}_E = \mathsf{p}_{|E|} = \{(m_1 + \cdots + m_n, [m_1, \ldots, m_n]) \mid m_1, \ldots, m_n \in \mathcal{M}_{\mathrm{fin}}(|E|)\}$ belongs to $\mathbf{NCoh}(!_{\mathsf{b}}E, !_{\mathsf{b}}!_{\mathsf{b}}E)$. So let $(m^i, M^i) \in \mathsf{p}_E$ for $i = 1, 2$ so that $M^i = [m^i_1, \ldots, m^i_{n^i}]$ and $m^i = \sum_{j=1}^{n^i} m^i_j$ for $i = 1, 2$. Assume first that $M^1 \smile_{!_{\mathsf{b}}!_{\mathsf{b}}E} M^2$. So let $j^i \in \{1, \ldots, n^i\}$ for $i = 1, 2$ be such that $m^1_{j^1} \smile_{!_{\mathsf{b}}E} m^2_{j^2}$. We can find $a^i \in m^i_{j^i}$ for $i = 1, 2$ such that $a^1 \smile_E a^2$. Since $a^i \in m^i$ for $i = 1, 2$, we have $m^1 \smile_{!_{\mathsf{b}}E} m^2$ as required. Assume now that $M^1 \nu_{!_{\mathsf{b}}!_{\mathsf{b}}E} M^2$ and let us prove that $m^1 \asymp_{!_{\mathsf{b}}E} m^2$. We have $n^1 = n^2 = n$ and, up to reindexing, we can assume that $m^1_j \nu_{!_{\mathsf{b}}E} m^2_j$ for $j = 1, \ldots, n$. So we can write $m^i_j = [a^i_{j,1}, \ldots, a^i_{j,k_j}]$ with $a^1_{j,l} \nu_E a^2_{j,l}$ for all $j = 1, \ldots, n$ and $l = 1, \ldots, k_j$ (k_j does not depend on i). If for all (j^1, l^1) and (j^2, l^2) we have $a^1_{j^1,l^1} \frown_E a^2_{j^2,l^2}$ and hence we have $m^1 \nu_{!_{\mathsf{b}}E} m^2$ and therefore $m^1 \asymp_{!_{\mathsf{b}}E} m^2$ as contended (because $m^i = [a^i_{j,l} \mid j \in \{1, \ldots n\}$ and $l \in \{1, \ldots, k_j\}]$). So assume that for some (j^1, l^1) and (j^2, l^2) we have $a^1_{j^1,l^1} \smile_E a^2_{j^2,l^2}$. It follows that $m^1_{j^1} \smile_{!_{\mathsf{b}}E} m^2_{j^2}$ and hence $M^1 \smile_{!_{\mathsf{b}}!_{\mathsf{b}}E} M^2$, contradicting our assumption.

To end the proof that **NCoh** is a model of LL, it suffices to prove that the Seely isomorphisms of Section 5.2 are actually morphisms of **NCoh**. This is obvious for m^0, so we are left with proving that given coherence spaces E_1 and E_2, the relation $\mathsf{m}^2_{|E_1|,|E_2|}$ (that we denote as $\mathsf{m}^2_{E_1,E_2}$) belongs to $\mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \,\&\, E_2))$. Given $m = [a_1, \ldots, a_n] \in |!_{\mathsf{b}}E_i| = \mathcal{M}_{\mathrm{fin}}(|E_i|)$, let $i \cdot m = [(i, a_1), \ldots, (i, a_n)] \in |!_{\mathsf{b}}(E_1 \,\&\, E_2)|$. Remember that

$$\mathsf{m}^2_{E_1,E_2} = \{((m_1, m_2), 1 \cdot m_1 + 2 \cdot m_2) \mid m_j \in |!_{\mathsf{b}}E_j| \text{ for } j = 1, 2\} .$$

Let $((m^i_1, m^i_2), m^i) \in \mathsf{m}^2_{E_1,E_2}$ for $i = 1, 2$ (so that $m^i = 1 \cdot m^i_1 + 2 \cdot m^i_2$). Assume first that $m^1 \smile_{!_{\mathsf{b}}(E_1 \& E_2)} m^2$. Due to the definition of $E_1 \,\&\, E_2$, there must be $a^i \in m^i_j$ for $i = 1, 2$ such that $a^1 \smile_{E_j} a^2$, for $j = 1$ or for $j = 2$; wlog. assume that $j = 1$. Then we have $m^1_1 \smile_{!_{\mathsf{b}}E_1} m^2_1$ and hence $(m^1_1, m^1_2) \smile_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m^2_1, m^2_2)$ as required. Assume next that $m^1 \nu_{!_{\mathsf{b}}(E_1 \& E_2)} m^2$ so that $m^1_j \nu_{!_{\mathsf{b}}E_j} m^2_j$ for $j = 1, 2$, as easily checked. It follows that $(m^1_1, m^1_2) \nu_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m^2_1, m^2_2)$ which ends the proof that $\mathsf{m}^2_{E_1,E_2} \in \mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \,\&\, E_2))$.

We also need to prove that $(\mathsf{m}^2_{|E_1|,|E_1|})^{-1} \in \mathbf{NCoh}(!_{\mathsf{b}}(E_1 \,\&\, E_2), !_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2)$ where $(\mathsf{m}^2_{|E_1|,|E_1|})^{-1}$ is of course $\{(1 \cdot m_1 + 2 \cdot m_2, (m_1, m_2)) \mid m_j \in |!_{\mathsf{b}}E_j| \text{ for } j = 1, 2\}$. So, with the same notations as above, assume that $(m^1_1, m^1_2) \smile_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m^2_1, m^2_2)$. Wlog. we can assume that $m^1_1 \smile_{!_{\mathsf{b}}E_1} m^2_1$ which clearly implies $m^1 \smile_{!_{\mathsf{b}}(E_1 \& E_2)} m^2$. Assume next that $(m^1_1, m^1_2) \nu_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m^2_1, m^2_2)$, that is $m^1_j \nu_{!_{\mathsf{b}}U_j} m^2_j$ for $j = 1, 2$. It follows clearly that $m^1 \nu_{!_{\mathsf{b}}(E_1 \& E_2)} m^2$.

Let us say that a coherence space E is a *Boudes' space* if $a \nu_E a \Rightarrow a = a'$. Observe that Boudes' condition is preserved by all coherence space constructions introduced so far. One benefit of this condition is that a clique and an anti-clique of a Boudes' space have at most one element in common (a basic feature of usual coherence spaces). Moreover, the following is a straightforward observation.

Proposition 5.3.1. *If E is a Boudes' space then so are $E^\perp$ and $!E$. If E_1 and E_2 are Boudes' spaces then so are $E_1 \otimes E_2$ and $E_1 \,\&\, E_2$. In other words, the full subcategory* **NCohB** *of* **NCoh** *whose objects are Boudes' spaces, equipped with Boudes' exponential $!_{\mathsf{b}}\_$, is a model of LL.*

It must also be noticed that Boudes' exponential is the free one, that is (**NCoh**, $!_{\mathsf{b}}\_$) (and probably also (**NCohB**, $!_{\mathsf{b}}\_$)) is a Lafont model of LL; it has been discovered when looking for such a free exponential in **NCoh**).

5.3.2 Another exponential

However the first exponential we discovered in **NCoh** seems quite different from Boudes' exponential and in some sense less intuitive. For instance it does not preserve the property of being a Boudes' space which looks like a rather natural one. We describe it shortly for the curious reader, denoting it as $!_{\mathsf{be}}\_$ for "Bucciarelli-Ehrhard exponential".

We set of course $|!_{\mathsf{be}}E| = \mathcal{M}_{\mathrm{fin}}(|E|)$. Given $m = [a_1, \ldots, a_k]$ and $m' = [a_{k+1}, \ldots, a_n]$ in $|\mathcal{M}_{\mathrm{fin}}(|E|)|$:

- $m \frown_{!_{\mathsf{be}}E} m'$ if $\forall i, j\ i \neq j \Rightarrow a_i \frown_E a_j$ (we say that $m + m'$ is a *multiclique*)
- $m \frown\!\!\!\frown_{!_{\mathsf{be}}E} m'$ if $m \frown_{!_{\mathsf{be}}E} m'$ and, moreover, $\exists i\ \forall j \neq i\ a_i \frown_E a_j$ (we say that $m + m'$ is a *star-shaped* multiclique and i is one of its centers).

It is important to observe that there is no condition on the coherence of a_i with itself, for a given i, only coherence conditions relative to points with distinct indices.

Notice that these definitions do not depend on the chosen enumerations of the multisets m and m', only on the multisets themselves. Observe also that $m\ \nu_{!_{\mathsf{be}}E}\ m'$ simply means that $m + m'$ is a multiclique which is not star-shaped, a condition much more liberal than Boudes'. For instance it does not imply at all that m and m' have the same number of elements as Boudes' does.

It is worthwile to check that the definition above, though admittedly a bit strange, leads to a perfectly regular exponential compatible with that of **Rel**. Let $R \in \mathbf{NCoh}(E, F)$ and let $(m, p), (m', p') \in !R$ (hence $m = [a_1, \ldots, a_k]$ and $m' = [a_{k+1}, \ldots, a_n]$, $p = [b_1, \ldots, b_k]$ and $p' = [b_{k+1}, \ldots, b_n]$, with $(a_i, b_i) \in R$ for each $i = 1, \ldots, n$). Assume first that $m \frown\!\!\!\frown_{!_{\mathsf{be}}E} m'$, that is, $m + m'$ is a multiclique (that is $i \neq j \Rightarrow a_i \frown_E a_j$). Since $\forall i, j\ (a_i, b_i) \frown_{E \multimap F} (a_j, b_j)$, it follows that $p + p'$ is a multiclique. For the same reason, if $m + m'$ is star-shaped with i as center, so is $p + p'$ and hence $m \frown_{!_{\mathsf{be}}E} m' \Rightarrow p \frown_{!_{\mathsf{be}}E} p'$. So $!_{\mathsf{be}}\_$ is a functor $\mathbf{NCoh} \to \mathbf{NCoh}$.

The fact the $\mathsf{d}_E = \mathsf{d}_{|E|} \in \mathbf{NCoh}(!_{\mathsf{be}}E, E)$ results again from the easy observation that

$$\forall a, a' \in |E| \quad [a] \frown_{!_{\mathsf{be}}E} [a'] \Leftrightarrow a \frown_E a' \text{ and } [a] \frown\!\!\!\frown_{!_{\mathsf{be}}E} [a'] \Leftrightarrow a \frown_E a'\,.$$

We prove now that $\mathsf{p}_E = \mathsf{p}_{|E|} \in \mathbf{NCoh}(!_{\mathsf{be}}E, !_{\mathsf{be}}!_{\mathsf{be}}E)$. So let $(m, M), (m', M') \in \mathsf{p}_E$ so that $M = [m_1, \ldots, m_k]$ with $m = \sum_{i=1}^n m_i$ and $M' = [m_{k+1}, \ldots, m_n]$

with $m' = \sum_{i=k+1}^{n} m_i$. Assume first that $m \supset_{!_{\mathsf{be}}E} m'$, that is, $m + m'$ is a multiclique. We must prove that $M + M' = [m_i \mid i = 1, \ldots, n]$ is a multiclique, that is $\forall i < j \; m_i \supset_{!_{\mathsf{be}}E} m_j$. But if $1 \leq i < j \leq n$ then $m_i + m_j$ is obviously a multiclique since $m_i + m_j \leq m + m'$ and $m + m' = \sum_{l=1}^{n} m_l$ is assumed to be a multiclique. Assume moreover that $m + m'$ is star-shaped. Remember that $m + m' = m_1 + \cdots + m_n$. We can write $m + m' = [a_1, \ldots, a_N]$ and wlog. we can assume that $\forall i > 1 \; a_1 \frown_E a_i$. Then a_1 appears in one of the m_j's, let us say in m_1 for the sake of readbility. Then for $j > 1$, $m_1 + m_j$ is a star-shaped multiclique and hence $m_1 \frown_{!_{\mathsf{be}}E} m_j$, showing that $M + M'$ itself is a star-shaped multclique. It follows that $M \frown_{!_{\mathsf{be}}!_{\mathsf{be}}E} M'$ as expected. This shows that $(!_{\mathsf{be}}\_, \mathsf{d}, \mathsf{p})$ is a comonad on **NCoh**, it remains to exhibit its Seely structure which of course must be that of $!\_$ in **Rel**.

We check that $\mathsf{m}^2_{|E_1|,|E_2|} \in \mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \,\&\, E_2))$ (and a similar statement for the inverse of this iso in **Rel**). Let $((m_1, m_2), m), ((m'_1, m'_2), m') \in \mathsf{m}^2_{|E_1|,|E_2|}$ and assume first that

$$(m_1, m_2) \supset_{!_{\mathsf{be}}E_1 \otimes !_{\mathsf{be}}E_2} (m'_1, m'_2)\,,$$

meaning that $m_i + m'_i$ is a multiclique in E_i, for $i = 1, 2$. Let us write $m_i = [a^i_1, \ldots, a^i_{k^i}]$, $m'_i = [a^i_{k^i+1}, \ldots, a^i_{n^i}]$ for $i = 1, 2$. Remember that $m = 1 \cdot m_1 + 2 \cdot m_2$ and similarly for m'. Then

$$\begin{aligned} m + m' &= 1 \cdot (m_1 + m'_1) + 2 \cdot (m_2 + m'_2) \\ &= [(1, a^1_1), \ldots, (1, a^1_{n^1}), (2, a^2_1), \ldots, (2, a^2_{n^2})]\,. \end{aligned}$$

For $l \in \{1, n^1 + n^2\}$ let us set $b_l = (1, a^1_l)$ if $1 \leq l \leq n^1$ and $b_l = (2, a^2_{l-n^1})$ if $n^1 + 1 \leq l \leq n^2$ so that $m + m' = [b_1, \ldots, b_{n^1+n^2}]$. It is clear that $m + m'$ a multiclique because the $m_i + m'_i$'s are and by definition of $E_1 \,\&\, E_2$. Assume moreover that $(m_1, m_2) \frown_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m'_1, m'_2)$. Wlog. assume that $m_1 \frown_{!_{\mathsf{be}}E_1} m'_1$, that is, $m_1 + m'_1$ is star-shaped with center, let's say, $l \in \{1, \ldots, k^1\}$. Then we have $b_l \frown_{E_1 \& E_2} b_{l'}$ for $l' \in \{1, \ldots, n^1\}$ because $m_1 + m'_1$ is star-shaped, and $b_l \frown_{E_1 \& E_2} b_{l'}$ for $l' \in \{n^1 + 1, \ldots, n^1 + n^2\}$ by definition of $E_1 \,\&\, E_2$.

To check that the inverse of $\mathsf{m}^2_{|E_1|,|E_2|}$ is a clique, we use the same notations and we assume first that $m \supset_{!_{\mathsf{be}}(E_1 \& E_2)} m'$, that is $m + m'$ is a clique in $E_1 \,\&\, E_2$ which implies that $m_i + m'_i$ is a multiclique in E_i for $i = 1, 2$ and hence $(m_1, m_2) \supset_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m'_1, m'_2)$. Assume moreover that $m + m'$ is star-shaped with $l \in \{1, \ldots, n^1 + n^2\}$ as center. By symmetry we can assume that $l \in \{1, \ldots, k^1\}$. Then $m_1 + m'_1$ is star-shaped and hence $m_1 \frown_{!_{\mathsf{be}}E_1} m'_1$, which implies that $(m_1, m_2) \frown_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m'_1, m'_2)$, ending the proof that **NCoh**, equipped with the $!_{\mathsf{be}}\_$ exponential, is a model of LL.

5.3.2.1 Example

We develop a simple example to illustrate the difference between the two exponentials. We have $|!_{\mathsf{b}}1| = |!_{\mathsf{be}}1| = \mathbf{N}$ and:

- Not surprisingly $n \; \nu_{!_{\mathsf{b}}1} \; n'$ iff $n = n'$ and $n \frown_{!_{\mathsf{b}}1} n'$ iff $n \neq n'$.

- Much more surprising is $!_{\mathsf{be}}1$: $n \ \nu_{!_{\mathsf{be}}1} \ n'$ iff $n + n' \neq 1$ and $n \frown_{!_{\mathsf{b}}1} n'$ iff $n + n' = 1$.

Indeed in **NCoh** the object 1 is characterized by $|1| = \{*\}$ with $* \ \nu_1 \ *$ and $\frown_1$ is empty. Therefore any $[*, \ldots, *]$ is a multiclique, but the only star-shaped multiclique if $[*]$. Of course $!_{\mathsf{be}}1$ is far from being a Boudes' space.

It follows that if $n, n' \in \mathbf{N}$:

- $\{((n,*),(1,*)),((n',*),(2,*))\} \in \mathsf{Cl}((!_{\mathsf{b}}1 \multimap 1) \multimap 1 \oplus 1)$ as soon as $n \neq n'$, meaning that the $!_{\mathsf{b}}\_$ based model is able to separate programs which differ by the number of use of their arguments in a very general way
- and $\{((n,*),(1,*)),((n',*),(2,*))\} \in \mathsf{Cl}((!_{\mathsf{be}}1 \multimap 1) \multimap 1 \oplus 1)$ iff $n+n' = 1$ meaning that the $!_{\mathsf{be}}\_$ based model seems to have a much more limited separation power: it can only separate a constant function from a function which uses its argument exactly once!

Another interesting observation is that $[(1,*),(2,*)] \smile_{!_{\mathsf{b}}(1\oplus 1)} [(1,*),(2,*)]$ (this is also true in $!_{\mathsf{be}}(1 \oplus 1)$) and therefore

$$\{([(1,*),(2,*)],(1,*)),([(1,*),(2,*)],(2,*))\} \in \mathbf{NCoh}(!_{\mathsf{b}}(1 \oplus 1), 1 \oplus 1)$$

5.4 Lamarche's strict coherence spaces

Another intersting model of LL based on the relational model has been introduced by François Lamarche in [**Lamarche-strict-coh**], and actually also arise as instances of the models generated by Index LL. We think useful to present them here because they feature at the same time a non trivial notion of coherence — in the sense that in $1 \oplus 1$, the two elements of the web do not form a clique — and that they seem closer to the Scott semantics of the λ-calculus in cpo's (which are not neecssarily lattices) that to the stable samantics.

Another interesting feature of this model is that it can be considered as a refinement of **Rel** by binary logical relations.

Definition 5.4.1. A strict coherence space is a pair $E = (|E|, \frown_E)$ where $|E|$ (the web of E) is a set and $\frown_E$ is a binary and symmetric relation on $|E|$.

Definition 5.4.2. Given a SCS E, we define an SCS $\mathcal{P}(E)$ by $|\mathcal{P}(E)| = \mathcal{P}(|E|)$ and $x_1 \frown_{\mathcal{P}(E)} x_2$ if $\forall (a_1, a_2) \in x_1 \times x_2 \ a_1 \frown_E a_2$. An element $x \in \mathcal{P}(|E|)$ such that $x \frown_{\mathcal{P}(E)} x$ is a *clique* of E, we use $\mathsf{Cl}(E)$ for the set of cliques of E.

Lemma 5.4.3. *The set $\mathsf{Cl}(E)$, ordered by inclusion, is a cpo whose least element is $\emptyset$.*

This is obvious. Observe that, as in NUCS, we do not always have $\{a\} \in \mathsf{Cl}(E)$ when $a \in |E|$.

Definition 5.4.4. The SCS $E \multimap F$ is defined by $|E \multimap F| = |E| \times |F|$ and $(a,b) \frown_{E \multimap F} (a',b')$ if $a \frown_E a' \Rightarrow b \frown_F b'$. The category **Scs** has the SCS as objects, and $\mathbf{Scs}(E,F) = \mathsf{Cl}(E \multimap F)$, composition and identities being defined as in **Rel**.

Indeed it is is clear that if $s \in \mathbf{Scs}(E,F)$ and $t \in \mathbf{Scs}(F,G)$ then $t\,s \in \mathbf{Scs}(E,G)$ and that $\mathsf{Id}_E = \{(a,a) \mid a \in |E|\}$ belongs to $\mathbf{Scs}(E,E)$.

Lemma 5.4.5. *Let* $(s_i \in |\mathcal{P}(E \multimap F)|)_{i=1,2}$*, one has* $s_1 \frown_{\mathcal{P}(E\multimap F)} s_2$ *iff* $\forall x_1, x_2 \in |\mathcal{P}(E)|\ x_1 \frown_{\mathcal{P}(E)} x_2 \Rightarrow s_1 \cdot x_1 \frown_F s_2 \cdot x_2$.

This proposition shows that the relation $\frown_{\mathcal{P}(E)}$ can be understood as a logical relation. Accordingly, a morphism $E \to F$ is an $s \in |\mathcal{P}(E \multimap F)|$ such that $s \frown_{\mathcal{P}(E\multimap F)} s$.

This category is an SMCC, we define $E \otimes F$ by $|E \otimes F| = |E| \times |F|$ and $(a_1,b_1) \frown_{E\otimes F} (a_2,b_2)$ if $a_1 \frown_E a_2$ and $b_1 \frown_F b_2$. Given $s \in \mathbf{Scs}(E_1,E_2)$ and $t \in \mathbf{Scs}(F_1,F_2)$ then $s \otimes t$ is defined as in **Rel** and it is easy to check that $s \otimes t \in \mathbf{Scs}(E_1 \otimes F_1, E_2 \otimes F_2)$. It is also trivial to check that the natural isomorphisms $\lambda, \rho, \alpha, \sigma$ are morphisms in **Scs**, for instance

$$\alpha_{|E_1|,|E_2|,|E_3|} \in \mathbf{Scs}((E_1 \otimes E_2) \otimes E_3, E_1 \otimes (E_2 \otimes E_3))$$

and in that way we have equipped **Scs** with a structure of SMC.

This SMC is closed, with $(E \multimap F, \mathsf{ev})$ as internal hom, the evaluation morphism $\mathsf{ev} \in \mathbf{Scs}((E \multimap F) \otimes E, F)$ is defined as in **Rel**, and similarly for the Curry transpose $\mathsf{cur}s \in \mathbf{Scs}(G, E \multimap F)$ when $s \in \mathbf{Scs}(G \otimes E, F)$. The unit of the tensor product is $1 = (\{*\}, \frown_1)$ with $* \frown_1 *$.

The category **Scs** is $*$-autonomous with dualizing object $\perp$ with $|\perp| = \{*\}$ and $* \smile_\perp *$. It is easy to check that $E \multimap \perp$ is isomorphic to the SCS $E^\perp$ with $|E^\perp| = |E|$ and $a \frown_{E^\perp} a'$ if $a \smile_E a'$.

Chapter 6

Categorical models of linear logic

We refer to Appendix C for the basic definitions and results of category theory that we rely upon.

6.1 Seely categories

We define now the basic notion of categorical model of Linear Logic. A Seely category is a tuple

$$(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \bot, !, \mathsf{d}, \mathsf{p}, \mathsf{m}^0, \mathsf{m}^2)$$

where $\mathcal{L}$ is a category and the additional components will be explained now.

6.1.1 The multiplicative structure

First we require that $1 \in \mathsf{Obj}(\mathcal{L})$, $\otimes : \mathcal{L}^2 \to \mathcal{L}$ is a functor and λ, ρ, α and σ are natural transformations which turn $\mathcal{L}$ into a symmetric monoidal category, see Section **??**.

Next we require this SMC to be closed. This means that for any two objects X and Y of $\mathcal{L}$, there is a pair $(X \multimap Y, \mathsf{ev})$ where $X \multimap Y$ is an object of $\mathcal{L}$ and $\mathsf{ev} \in \mathcal{L}((X \multimap Y) \otimes X, Y)$, and this pair has the following universal property: for any object Z of $\mathcal{L}$ and any $f \in \mathcal{L}(Z \otimes X, Y)$ there is exactly one morphism $\mathsf{cur} f \in \mathcal{L}(Z, X \multimap Y)$ such that the following diagram commutes

$$\begin{array}{ccc} Z \otimes X & \xrightarrow{\mathsf{cur} f \otimes X} & (X \multimap Y) \otimes X \\ & \searrow^{f} & \downarrow{\scriptstyle \mathsf{ev}} \\ & & Y \end{array}$$

In other terms $(X \multimap Y, \mathsf{ev})$ is the terminal object in the category $\mathcal{E}(X, Y)$ whose objects are the pairs (Z, f) where $Z \in \mathsf{Obj}(\mathcal{L})$ and $f \in \mathcal{L}(Z \otimes X, Y)$

and an element of $\mathcal{E}((Z,f),(Z',f'))$ is a $g \in \mathcal{L}(Z,Z')$ such that the following diagram commutes

$$\begin{array}{ccc} Z \otimes X & \xrightarrow{g \otimes X} & Z' \otimes X \\ & \searrow^{f} & \downarrow{\scriptstyle f'} \\ & & Y \end{array}$$

and composition in $\mathcal{E}(X,Y)$ is defined as in $\mathcal{L}$.

Another way of expressing the same property is to say that, for any object X of $\mathcal{L}$, the functor $\_ \otimes X : \mathcal{L} \to \mathcal{L}$ has a right adjoint $X \multimap \_ : \mathcal{L} \to \mathcal{L}$.

Last, this universal property is equivalent to the satisfaction of the following three equations:

- if $f \in \mathcal{L}(Z \otimes X, Y)$, one has $\mathsf{ev}\,(\mathsf{cur} f \otimes X) = f$;
- if moreover $g \in \mathcal{L}(Z', Z)$, one has $(\mathsf{cur} f)\, g = \mathsf{cur}(f\,(g \otimes X))$
- and last $\mathsf{cur}\,\mathsf{ev} = \mathsf{Id}_{X \multimap Y}$.

It is then possible to turn $\_ \multimap \_$ into a functor $\mathcal{L}^{\mathsf{op}} \times \mathcal{L} \to \mathcal{L}$. Of course this functor maps (X,Y) to $X \multimap Y$. Let now $f \in \mathcal{L}(X',X)$ and $g \in \mathcal{L}(Y,Y')$. We define $f \multimap g$ as the unique element of $\mathcal{L}(X \multimap Y, X' \multimap Y')$ such that the following diagram commutes

$$\begin{array}{ccccc} (X \multimap Y) \otimes X' & & \xrightarrow{(f \multimap g) \otimes X'} & & (X' \multimap Y') \otimes X' \\ {\scriptstyle (X \multimap Y) \otimes f}\downarrow & & & & \downarrow{\scriptstyle \mathsf{ev}} \\ (X \multimap Y) \otimes X & \xrightarrow{\mathsf{ev}} & Y & \xrightarrow{\quad g \quad} & Y' \end{array}$$

This means that

$$f \multimap g = \mathsf{cur}(g\ \mathsf{ev}\,((X \multimap Y) \otimes f))\,.$$

Functoriality results from the universal property. For instance, if $f' \in \mathcal{L}(X'',X')$ and $g' \in \mathcal{L}(Y',Y'')$, both elements h of $\{(f\,f') \multimap (g'\,g), (f' \multimap g')\,(f \multimap g)\}$ make the following diagram commutative

$$\begin{array}{ccccc} (X \multimap Y) \otimes X'' & & \xrightarrow{h \otimes X''} & & (X'' \multimap Y'') \otimes X'' \\ {\scriptstyle (X \multimap Y) \otimes (f'\,f)}\downarrow & & & & \downarrow{\scriptstyle \mathsf{ev}} \\ (X \multimap Y) \otimes X & \xrightarrow{\mathsf{ev}} & Y & \xrightarrow{\quad g'\,g \quad} & Y' \end{array}$$

by functoriality of $\otimes$ and hence they are equal.

Theorem 6.1.1. *The functor $\otimes$ commutes with all colimits existing in $\mathcal{L}$ and for each object X of $\mathcal{L}$ the functor $X \multimap \_$ commutes with all existing limits.*

Indeed for each object X the functor $\_ \otimes X$ is left adjoint to the functor $X \multimap \_$.

6.1.1.1 $*$-autonomy

Let Z be an object of $\mathcal{L}$. For any object X, we have a morphism

$$\eta_X^{(Z)} \in \mathcal{L}(X, (X \multimap Z) \multimap Z)$$

defined as $\mathsf{cur}(\varphi)$ where φ is the following composition of morphisms

$$X \otimes (X \multimap Z) \xrightarrow{\sigma} (X \multimap Z) \otimes X \xrightarrow{\mathsf{ev}} Z$$

This morphism defines a natural transformation from the identity functor to the functor $(\_ \multimap Z) \multimap Z$.

Lemma 6.1.2. *The following composition of morphisms*

$$X \multimap Z \xrightarrow{\eta^{(Z)}_{X \multimap Z}} ((X \multimap Z) \multimap Z) \multimap Z \xrightarrow{\eta^{(Z)}_X \multimap Z} X \multimap Z$$

coincides with the identity at $X \multimap Z$.

Proof. We have

$$\begin{aligned}
(\eta_X^{(Z)} \multimap Z)\, \eta^{(Z)}_{X \multimap Z} &= \mathsf{cur}(\mathsf{ev}\,(\mathsf{Id} \otimes \eta_X^{(Z)}))\, \eta^{(Z)}_{X \multimap Z} \\
&= \mathsf{cur}(\mathsf{ev}(\eta^{(Z)}_{X \multimap Z} \otimes \eta_X^{(Z)})) \\
&= \mathsf{cur}(\mathsf{ev}\,(\eta^{(Z)}_{X \multimap Z} \otimes \mathsf{Id})\,(\mathsf{Id} \otimes \eta_X^{(Z)})) \\
&= \mathsf{cur}(\mathsf{ev}\ \sigma\,(\mathsf{Id} \otimes \eta_X^{(Z)})) \\
&= \mathsf{cur}(\mathsf{ev}\,(\eta_X^{(Z)} \otimes \mathsf{Id})\,\sigma) \\
&= \mathsf{cur}(\mathsf{ev}\ \sigma\,\sigma) = \mathsf{Id}\ .
\end{aligned}$$

□

Definition 6.1.3. An object Z of $\mathcal{L}$ is a *dualizing object* if the morphism $\eta_X^{(Z)}$ is an iso for each object X of $\mathcal{L}$. A $*$-autonomous category is a tuple $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \bot)$ where $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma)$ is an SMCC and $\bot \in \mathsf{Obj}(\mathcal{L})$ is a dualizing object.

In the definition of a Seely category, the structure $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \bot)$ is assumed to be a $*$-autonomous category. We use then the notation $\_^\bot$ for the functor $\_ \multimap \bot : \mathcal{L}^{\mathsf{op}} \to \mathcal{L}$. We set $\eta_X = \eta_X^{(\bot)}$ so that for each object X of $\mathcal{L}$, $\eta_X \in \mathcal{L}(X, X^{\bot\bot})$ is an iso which is natural in X.

Lemma 6.1.4. *Let X be an object of $\mathcal{L}$, we have $\eta_X^\bot\, \eta_{X^\bot} = \mathsf{Id}_{X^\bot}$.*

This is just a special case of Lemma 6.1.2.

Remark 6.1.5. We could call *intuitionistic* Seely category a structure defined exactly as a Seely category, without the data of a dualizing object $\bot$.

6.1.2 The additive structure

A Seely category is also assumed to be cartesian, and more precisely to be equipped with a choice of terminal object $\top$ and cartesian product $(X_1 \mathbin{\&} X_2, \mathsf{pr}_1, \mathsf{pr}_2)$. In most known models countable cartesian products are also available and we denote them as

$$(\mathop{\&}_{i\in I} X_i, (\mathsf{pr}_i)_{i\in I})\,.$$

Remember that this means that, for any family $(f_i)_{i\in I}$ of morphisms $f_i \in \mathcal{L}(Y, X_i)$ there is exactly one morphism

$$f \in \mathcal{L}(Y, \mathop{\&}_{i\in I} X_i)$$

such that $\mathsf{pr}_i\ f = f_i$ for each $i \in I$. We set $f = \langle f_i\rangle_{i\in I}$.

Theorem 6.1.6. *If a $*$-autonomous category $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \bot)$ is (countably) cartesian, it is also (countably) cocartesian.*

Of course a similar statement holds for all kinds of limits and colimits: if $\mathcal{L}$ has equalizers, it also has coequalizers etc. The proof is straightforawrd. Let $(X_i)_{i\in I}$ be a family of objects (finite, or countable in the case where $\mathcal{L}$ is countably cartesian). Then

$$(\mathop{\oplus}_{i\in I} X_i, (\mathsf{in}_i)_{i\in I}) = ((\mathop{\&}_{i\in I} X_i^\bot)^\bot, (\mathsf{pr}_i^\bot\ \eta_{X_i})_{i\in I})$$

is the coproduct of the X_i's; notice indeed that $\mathsf{in}_j = \mathsf{pr}_j^\bot\ \eta_{X_j} \in \mathcal{L}(X_j, \oplus_{i\in I} X_i)$.

To check this fact, let $(f_i)_{i\in I}$ be a family of morphisms with $f_i \in \mathcal{L}(X_i, Y)$. Then we have $f_i^\bot \in \mathcal{L}(Y^\bot, Y_i^\bot)$ and hence $\langle f_i^\bot\rangle_{i\in I} \in \mathcal{L}(Y^\bot, \&_{i\in I} X_i^\bot)$. Therefore we set

$$[f_i]_{i\in I} = {\eta_Y}^{-1}\,\langle f_i^\bot\rangle_{i\in I}^\bot \in \mathcal{L}(\mathop{\oplus}_{i\in I} X_i, Y)\,.$$

Given $j \in I$, we have

$$\begin{aligned}
[f_i]_{i\in I}\ \mathsf{in}_j &= {\eta_Y}^{-1}\,\langle f_i^\bot\rangle_{i\in I}^\bot\ \mathsf{pr}_j^\bot\ \eta_{X_j} \\
&= {\eta_Y}^{-1}\,(\mathsf{pr}_j\,\langle f_i^\bot\rangle_{i\in I})^\bot\,\eta_{X_j} \\
&= {\eta_Y}^{-1}\,{f_j}^{\bot\bot}\,\eta_{X_j} \\
&= f_j
\end{aligned}$$

by naturality of η. Notice that for all $i \in I$ we have

$$\begin{aligned}
\mathsf{in}_i^\bot &= \eta_{X_i}^\bot\ \mathsf{pr}_i^{\bot\bot} \\
&= {\eta_{X_i^\bot}}^{-1}\ \mathsf{pr}_i^{\bot\bot} \quad \text{by Lemma 6.1.4} \\
&= \mathsf{pr}_i\ (\eta_{\&_{i\in I} X_i^\bot})^{-1} \quad \text{by naturality of } \eta^{-1}
\end{aligned}$$

Last let $f \in \mathcal{L}(\oplus_{i\in I} X_i, Y)$ be such that $f\ \mathsf{in}_i = f_i$ for each $i \in I$. Let $j \in I$, we have

$$\begin{aligned}\mathsf{pr}_j\ (\eta_{\&_{i\in I}\, X_i^\perp})^{-1}\, f^\perp &= \mathsf{in}_j^\perp\ f^\perp \\ &= (f\ \mathsf{in}_j)^\perp \\ &= f_j^\perp\,.\end{aligned}$$

It follows that $(\eta_{\&_{i\in I}\, X_i^\perp})^{-1}\, f^\perp = \langle f_j^\perp\rangle_{j\in I}$. Hence $f^\perp = \eta_{\&_{i\in I}\, X_i^\perp}\, \langle f_j^\perp\rangle_{j\in I}$, therefore $f^{\perp\perp} = \langle f_j^\perp\rangle_{j\in I}^\perp\, \eta^\perp_{\&_{i\in I}\, X_i^\perp}$. By Lemma 6.1.4 we have $f^{\perp\perp}\, \eta_{(\&_{i\in I}\, X_i^\perp)^\perp} = \langle f_j^\perp\rangle_{j\in I}^\perp$. By naturality of η it follows that $\eta_Y\, f = \langle f_j^\perp\rangle_{j\in I}^\perp$ and hence $f = \eta_Y{}^{-1}\, \langle f_j^\perp\rangle_{j\in I}^\perp = [f_i]_{i\in I}^\perp$. Therefore $(\oplus_{i\in I} X_i, (\mathsf{in}_i)_{i\in I})$ is the coproduct of the X_i's in $\mathcal{L}$.

6.1.3 The exponential structure

Let $\mathcal{L}$ be a cartesian monoidal category with the same notations as above. An *exponential structure* on $\mathcal{L}$ is a tuple

$$(!\_, \mathsf{d}_X, \mathsf{p}_X, \mathsf{m}^0, \mathsf{m}^2_{X,Y})$$

where $(!\_, \mathsf{d}_X, \mathsf{p}_X)$ is a comonad on $\mathcal{L}$, meaning that $!\_ : \mathcal{L} \to \mathcal{L}$ is a functor and $\mathsf{d}_X \in \mathcal{L}(!X, X)$ and $\mathsf{p}_X \in \mathcal{L}(!X, !!X)$ are natural transformations which satisfy

$$\begin{array}{ccc} !X & \xrightarrow{\mathsf{p}_X} & !!X \\ & {\scriptstyle X}\searrow & \downarrow{\scriptstyle \mathsf{d}_{!X}} \\ & & !X \end{array} \qquad \begin{array}{ccc} !X & \xrightarrow{\mathsf{p}_X} & !!X \\ & {\scriptstyle X}\searrow & \downarrow{\scriptstyle !\mathsf{d}_X} \\ & & !X \end{array} \qquad \begin{array}{ccc} !X & \xrightarrow{\mathsf{p}_X} & !!X \\ {\scriptstyle \mathsf{p}_X}\downarrow & & \downarrow{\scriptstyle !\mathsf{p}_X} \\ !!X & \xrightarrow{\mathsf{p}_{!X}} & !!!X \end{array}$$

The two last morphisms are the *Seely isomorphisms*: $\mathsf{m}^0 \in \mathcal{L}(1, !\top)$ is an isomorphism and $\mathsf{m}^2_{X_1,X_2} \in \mathcal{L}(!X_1 \otimes !X_2, !(X_1 \,\&\, X_2))$ is a natural isomorphism. Technically, these isomorphisms equip the comonad $!\_$ with a *symmetric monoidality structure* from the SMC $(\mathcal{L}, \&, \top)$ to the SMC $(\mathcal{L}, \otimes, 1)$. More explicitly this means that the following diagrams commute. The first one expresses compatibility with the associators of the two SMC's:

$$\begin{array}{ccc} (!X_1 \otimes !X_2) \otimes !X_3 & \xrightarrow{\alpha_{!X_1,!X_2,!X_3}} & !X_1 \otimes (!X_2 \otimes !X_3) \\ {\scriptstyle \mathsf{m}^2_{X_1,X_2}\otimes !X_3}\downarrow & & \downarrow{\scriptstyle !X_1 \otimes \mathsf{m}^2_{X_2,X_3}} \\ !(X_1 \,\&\, X_2) \otimes !X_3 & & !X_1 \otimes !(X_2 \,\&\, X_3) \\ {\scriptstyle \mathsf{m}^2_{X_1\&X_2,X_3}}\downarrow & & \downarrow{\scriptstyle \mathsf{m}^2_{X_1,X_2\&X_3}} \\ !((X_1 \,\&\, X_2) \,\&\, X_3) & \xrightarrow{\langle \mathsf{pr}_1\ \mathsf{pr}_1, \langle \mathsf{pr}_2\ \mathsf{pr}_1, \mathsf{pr}_2\rangle\rangle} & !(X_1 \,\&\, (X_2 \,\&\, X_3)) \end{array}$$

The second one deals with the commutators:

$$\begin{array}{ccc} !X_1 \otimes !X_2 & \xrightarrow{\sigma_{!X_1,!X_2}} & !X_2 \otimes !X_1 \\ {\scriptstyle \mathsf{m}^2_{X_1,X_2}}\downarrow & & \downarrow{\scriptstyle \mathsf{m}^2_{X_2,X_1}} \\ !(X_1 \,\&\, X_2) & \xrightarrow{!\langle \mathsf{pr}_2, \mathsf{pr}_1\rangle} & !(X_2 \,\&\, X_1) \end{array}$$

And the last one deals with the neutrality isos:

$$\begin{array}{ccc} !X \otimes 1 & \xrightarrow{\rho_X} & !X \\ {\scriptstyle !X\otimes \mathsf{m}^0}\downarrow & & \downarrow{\scriptstyle !\langle X, \mathsf{t}_X\rangle} \\ !X \otimes !\top & \xrightarrow{\mathsf{m}^2_{X,\top}} & !(X \,\&\, \top) \end{array} \qquad \begin{array}{ccc} 1 \otimes !X & \xrightarrow{\lambda_X} & !X \\ {\scriptstyle \mathsf{m}^0\otimes !X}\downarrow & & \downarrow{\scriptstyle !\langle \mathsf{t}_X, X\rangle} \\ !\top \otimes !X & \xrightarrow{\mathsf{m}^2_{\top,X}} & !(\top \,\&\, X) \end{array}$$

The compatibility of this symmetric monoidal structure with the comonad structure of $!\_$ is expressed by the following diagram

$$\begin{array}{ccc} !X_1 \otimes !X_2 & \xrightarrow{\mathsf{m}^2_{X_1,X_2}} & !(X_1 \,\&\, X_2) \\ & & \downarrow{\scriptstyle \mathsf{p}_{X_1\&X_2}} \\ {\scriptstyle \mathsf{p}_{X_1}\otimes \mathsf{p}_{X_2}}\downarrow & & !!(X_1 \,\&\, X_2) \\ & & \downarrow{\scriptstyle !\langle !\mathsf{pr}_1, !\mathsf{pr}_2\rangle} \\ !!X_1 \otimes !!X_2 & \xrightarrow{\mathsf{m}^2_{!X_1,!X_2}} & !(!X_1 \,\&\, !X_2) \end{array} \tag{6.1}$$

Definition 6.1.7. A *Seely category* is a cartesian $*$-autonomous category equipped with an exponential structure.

6.1.4 Derived structures in a Seely category

6.1.4.1 Promotion and the lax tensorial monoidality of the exponential

Given $f \in \mathcal{L}(!X, Y)$, we can define $f^! \in \mathcal{L}(!X, !Y)$ by $f^! = !f\ \mathsf{p}_X$. This is the *unary promotion* of f.

Given more generally $f \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, Y)$ we want now to define an n-ary promotion $f^! \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, !Y)$ in order to interpret the rule $(!_R)$.

For this we define $\mu^0 \in \mathcal{L}(1, !1)$ and $\mu^2_{X_1,X_2} \in \mathcal{L}(!X_1 \otimes !X_2, !(X_1 \otimes X_2))$. The first of these morphisms is defined as the following composition of morphisms in $\mathcal{L}$

$$1 \xrightarrow{\mathsf{m}^0} !\top \xrightarrow{\mathsf{p}_\top} !!\top \xrightarrow{!(\mathsf{m}^0)^{-1}} !1 \ .$$

and the second one is defined as follows:

$$!X_1 \otimes !X_2 \xrightarrow{\mathsf{p}_{X_1}\otimes \mathsf{p}_{X_2}} !!X_1 \otimes !!X_2 \xrightarrow{\mathsf{m}^2_{!X_1,!X_2}} !(!X_1 \,\&\, !X_2) \xrightarrow{!(\mathsf{d}_{X_1}\&\mathsf{d}_{X_2})} !(X_1 \,\&\, X_2)$$

It results straightforwardly from the definition that $\mu^2_{X_1,X_2}$ is natural in X_1 and X_2. These two morphisms equip the functor ! with a lax[1] symmetric monoidal structure, from the monoidal category $(\mathcal{L}, 1, \otimes)$ to itself. This means that the following diagrams commute

[1] "lax" means that the associated natural transformations are not isos in general.

$$\begin{array}{ccccc} 1 \otimes !X & \xrightarrow{\mu^0 \otimes !X} & !1 \otimes !X & \xrightarrow{\mu^2_{1,X}} & !(1 \otimes X) \\ & \searrow^{\lambda_{!X}} & & & \downarrow{\scriptstyle !\lambda_X} \\ & & & & !X \end{array}$$

$$\begin{array}{ccccc} (!X_1 \otimes !X_2) \otimes !X_3 & \xrightarrow{\mu^2_{X_1,X_2} \otimes !X_3} & !(X_1 \otimes X_2) \otimes !X_3 & \xrightarrow{\mu^2_{X_1 \otimes X_2, X_3}} & !((X_1 \otimes X_2) \otimes X_3) \\ \downarrow{\scriptstyle \alpha_{!X_1,!X_2,!X_3}} & & & & \downarrow{\scriptstyle !\alpha_{X_1,X_2,X_3}} \\ !X_1 \otimes (!X_2 \otimes !X_3) & \xrightarrow{!X_1 \otimes \mu^2_{X_2,X_3}} & !X_1 \otimes !(X_2 \otimes X_3) & \xrightarrow{\mu^2_{X_1, X_2 \otimes X_3}} & !(X_1 \otimes (X_2 \otimes X_3)) \end{array}$$

$$\begin{array}{ccc} !X_1 \otimes !X_2 & \xrightarrow{\mu^2_{X_1,X_2}} & !(X_1 \otimes X_2) \\ \downarrow{\scriptstyle \sigma_{!X_1,!X_2}} & & \downarrow{\scriptstyle !\sigma_{X_1,X_2}} \\ !X_2 \otimes !X_1 & \xrightarrow{\mu^2_{X_2,X_1}} & !(X_2 \otimes X_1) \end{array}$$

Exercise 6.1.8. Prove that the three diagrams above commute.

The compatibility of this lax monoidality with the comonad structure of ! is expressed as follows.

Proposition 6.1.9. *Let X_1 and X_2 be objects of $\mathcal{L}$, the following diagrams commutes*

$$\begin{array}{ccc} 1 & \xrightarrow{\mu^0} & !1 \\ & \searrow & \downarrow{\scriptstyle \mathsf{d}_1} \\ & & 1 \end{array} \qquad \begin{array}{ccc} 1 & \xrightarrow{\mu^0} & !1 \\ {\scriptstyle \mu^0}\downarrow & & \downarrow{\scriptstyle \mathsf{p}_1} \\ !1 & \xrightarrow{!\mu^0} & !!1 \end{array}$$

and

$$\begin{array}{ccccc} !X_1 \otimes !X_2 & & \xrightarrow{\mu^2_{X_1,X_2}} & & !(X_1 \otimes X_2) \\ & {\scriptstyle \mathsf{d}_{X_1} \otimes \mathsf{d}_{X_2}} \searrow & & \swarrow {\scriptstyle \mathsf{d}_{X_1 \otimes X_2}} & \\ & & X_1 \otimes X_2 & & \end{array}$$

$$\begin{array}{ccccc} !X_1 \otimes !X_2 & & \xrightarrow{\qquad\qquad\mu^2_{X_1,X_2}\qquad\qquad} & & !(X_1 \otimes X_2) \\ {\scriptstyle \mathsf{p}_{X_1} \otimes \mathsf{p}_{X_2}}\downarrow & & & & \downarrow{\scriptstyle \mathsf{p}_{X_1 \otimes X_2}} \\ !!X_1 \otimes !!X_2 & \xrightarrow{\mu^2_{!X_1,!X_2}} & !(!X_1 \otimes !X_2) & \xrightarrow{!\mu^2_{X_1,X_2}} & !!(X_1 \otimes X_2) \end{array}$$

Proof sketch. Direct application of the definitions of μ^0 and μ^2, and of our assumptions on the Seely morphisms. □

Exercise 6.1.10. Prove Proposition 6.1.9.

If we consider the isomorphisms α as identities (that is, if we identify the objects $(X \otimes Y) \otimes Z$ and $X \otimes (Y \otimes Z)$), then it makes sense to write an n-ary tensor as $X_1 \otimes \cdots \otimes X_n$, without parentheses. This is of course an abuse of

notation which can be suitably corrected by inserting parentheses and explicit isomorphisms. The property above of μ^2 means precisely that, independently of these choices of representations of n-ary tensors, we can define canonical morphism

$$\mu^n \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_1, !(X_1 \otimes \cdots \otimes X_n))$$

by combining freely occurrences of μ^2, the order in which we use them does not matter thanks to the coherence diagrams commutations satisfied by the monoidality isomorphisms associated with $\otimes$ (we can actually even insert 1's and permute factors).

Remark 6.1.11. These considerations can be made formal using the monoidal trees of Appendix C.5. For instance, given a monoidal tree τ of degree $n \in \mathbf{N}$, we can define $\mu^\tau \in \mathcal{L}(\mathsf{T}^\otimes_\tau(!X_1, \ldots, !X_n), !\mathsf{T}^\otimes_\tau(X_1, \ldots, X_n))$ and then replace all our lousy n-ary tensors $X_1 \otimes \cdots \otimes X_n$ by well-defined applications of the $\mathsf{T}^\otimes_\tau(\_)$ operator and parameterize all the generalized constructions we introduce by monoidal trees τ. The lax monoidality diagrams make it then possible to prove typically that if σ is another monoidal tree of degree n the following diagram commutes

$$\begin{array}{ccc} \mathsf{T}^\otimes_\sigma(!X_1, \ldots, !X_n) & \xrightarrow{\mu^\sigma} & !\mathsf{T}^\otimes_\sigma(X_1, \ldots, X_n) \\ \downarrow{\scriptstyle \varphi^\otimes_{\sigma,\tau}(!X_1,\ldots,!X_n)} & & \downarrow{\scriptstyle !\varphi^\otimes_{\sigma,\tau}(X_1,\ldots,X_n)} \\ \mathsf{T}^\otimes_\tau(!X_1, \ldots, !X_n) & \xrightarrow{\mu^\tau} & !\mathsf{T}^\otimes_\tau(X_1, \ldots, X_n) \end{array}$$

We prefer keep using n-ary tensor in order to make the presentation more readable, but the reader should be convinced that it can be made fully formal tnaks to these monoidal trees.

Thanks to these morphisms, we can generalize promotion as follows.

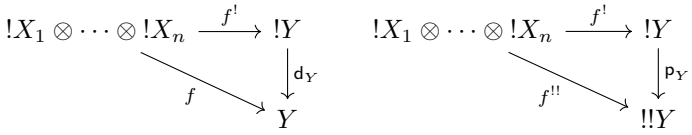
Let $f \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, Y)$, we define $f^! \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, !Y)$ as the following composition of morphisms in $\mathcal{L}$

$$\begin{array}{c} !X_1 \otimes \cdots \otimes !X_n \\ \downarrow{\scriptstyle \mathsf{p}_{X_1} \otimes \cdots \otimes \mathsf{p}_{X_n}} \\ !!X_1 \otimes \cdots \otimes !!X_n \\ \downarrow{\scriptstyle \mu^n_{!X_1,\ldots,!X_n}} \\ !(!X_1 \otimes \cdots \otimes !X_n) \\ \downarrow{\scriptstyle !f} \\ !Y \end{array}$$

We simply denote this morphism again as $f^!$ because the only case where this choice can introduce an ambiguity is $n = 1$, and in that case, both notions coincide. Observe that this definition also makes sense when $n = 0$, in which

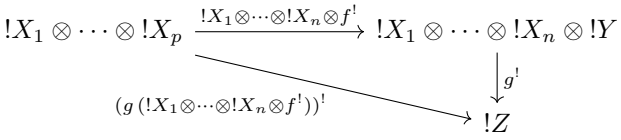
case we have $f \in \mathcal{L}(1, Y)$ and $f^! \in \mathcal{L}(1, !Y)$. Again, if we were using monoidal trees to make the presentation formally correct, this promotion operation should be parameterized by a monoidal tree τ.

Let $f \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, Y)$. The two following diagrams commute.



Exercise 6.1.12. Prove these commutations.

Exercise 6.1.13. Let $g \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n \otimes !Y, Z)$ and $f \in \mathcal{L}(!X_{n+1} \otimes \ldots \otimes !X_p, Y)$. Prove that the following diagram commutes



6.1.4.2 The structural morphisms

We define now morphisms corresponding to the structural rules of Linear Logic, weakening and contraction. Let X be an object of $\mathcal{L}$. We define $\mathsf{w}_X \in \mathcal{L}(!X, 1)$ as the following composition of morphisms in $\mathcal{L}$:

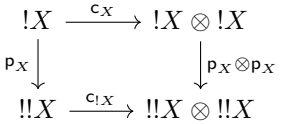
$$!X \xrightarrow{!\mathsf{t}_X} !\top \xrightarrow{(\mathsf{m}^0)^{-1}} 1$$

where t_X is the unique element of $\mathcal{L}(X, \top)$ (since $\top$ is the terminal object of $\mathcal{L}$). Similarly, we define $\mathsf{c}_X \in \mathcal{L}(!X, !X \otimes !X)$ as the following composition of morphisms

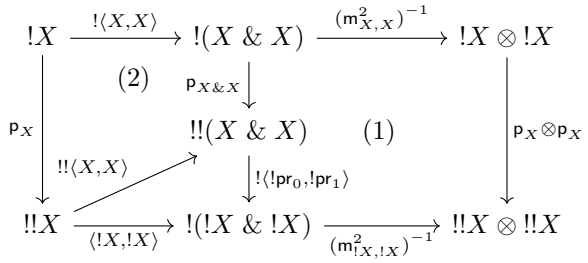
$$!X \xrightarrow{!\langle X,X\rangle} !(X \,\&\, X) \xrightarrow{(\mathsf{m}^2_{X,X})^{-1}} !X \otimes !X$$

Then one proves easily (exercise) that $(!X, \mathsf{w}_X, \mathsf{c}_X)$ is a symmetric comonoid in the SMC $(\mathcal{L}, 1, \otimes)$, in the sense of Section C.5.1.

Proposition 6.1.14. *For any object X of $\mathcal{L}$, the following diagram commutes*

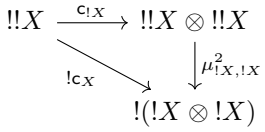


Proof. In the following diagram



the square (2) commutes by naturality of p, the pentagon (1) is an instance of Eq. (6.1) and the commutation of the remainig triangle results from the fonctoriality of ! and of the pasic properties of the cartesian product. □

Proposition 6.1.15. *For any object X of $\mathcal{L}$, the following diagram commutes*



Exercise 6.1.16. Prove Proposition 6.1.15.

6.1.4.3 The Kleisli category

We have defined in Section C.4 the Kleisli category of a general comonad, we apply it now to the comonad "!" and get the category $\mathcal{L}_!$. This means that the objects of this category are those of $\mathcal{L}$, that the identity at X is d_X and that composition of $f \in \mathcal{L}_!(X,Y)$ and $g \in \mathcal{L}_!(Y,Z)$ is $g \circ f = g\,!f\,\mathsf{p}_X$. Notice that there is a functor $\mathsf{Der} : \mathcal{L} \to \mathcal{L}_!$ which acts as the identity on objects and maps $f \in \mathcal{L}(X,Y)$ to $f\,\mathsf{d}_X$. This functor is faithful but not full and satisfies the following property.

Lemma 6.1.17. *If $f \in \mathcal{L}_!(X,Y)$ and $g \in \mathcal{C}(Y,Z)$ then $\mathsf{Der}(g) \circ f = g\,f$.*

Lemma 6.1.18. *The category $\mathcal{L}_!$ is cartesian.*

Proof. Given a finite family of objects $(X_i)_{i\in I}$, $(\&_{i\in I}\, X_i, (\mathsf{Pr}_i)_{i\in I})$ is the product of the X_i's in $\mathcal{L}_!$, if we set $\mathsf{Pr}_i = \mathsf{Der}(\mathsf{pr}_i)$ for each $i \in I$. Let indeed $(f_i \in \mathcal{L}_!(X,X_i))_{i\in I}$, that is $(f_i \in \mathcal{L}(!X,X_i))_{i\in I}$, then we have $\langle f_i\rangle_{i\in I} \in \mathcal{L}_!(X,\&_{i\in I}\,X_i)$ and $\mathsf{Pr}_i \circ \langle f_j\rangle_{j\in I} = \mathsf{pr}_i\,\langle f_j\rangle_{j\in I} = f_i$ for each $i \in I$ and if $g \in \mathcal{L}_!(X,\&_{i\in I}\,X_i)$ we have $g = \langle f_j\rangle_{j\in I}$ by application of the universal property of the product in $\mathcal{L}$. □

The associated cartesian product functor $\&_! : \mathcal{L}_!^2 \to \mathcal{L}_!$ acts as the $\mathcal{L}$ cartesian product of $\mathcal{L}$ on objects, and given $(f_i \in \mathcal{L}_!(X_i,Y_i))_{i=1,2}$, then $f_1 \,\&_!\, f_2 \in \mathcal{L}_!(X_1 \,\&\, X_2, Y_1 \,\&\, Y_2)$ is the following composition of morphisms

$$!(X_1 \,\&\, X_2) \xrightarrow{\langle !\mathsf{pr}_1, !\mathsf{pr}_2\rangle} !X_1 \,\&\, !X_2 \xrightarrow{f_1 \& f_2} Y_1 \,\&\, Y_2$$

Theorem 6.1.19. *The category $\mathcal{L}_!$ is cartesian closed.*

Proof. Given objects X, Y of $\mathcal{L}$ we set

$$(X \Rightarrow Y) = (!X \multimap Y)$$

and define $\mathsf{Ev} \in \mathcal{L}_!((X \Rightarrow Y) \,\&\, X, Y)$ as the following composition of morphisms in $\mathcal{L}$:

$$\begin{array}{c} !((!X \multimap Y) \,\&\, X) \\ \downarrow{\scriptstyle (\mathsf{m}^2_{!X\multimap Y, X})^{-1}} \\ !(!X \multimap Y) \otimes !X \\ \downarrow{\scriptstyle \mathsf{d}_{!X\multimap Y} \otimes \mathsf{Id}} \\ (!X \multimap Y) \otimes !X \\ \downarrow{\scriptstyle \mathsf{ev}} \\ Y \end{array}$$

The pair $(X \Rightarrow Y, \mathsf{Ev})$ is the internal hom of X, Y in $\mathcal{L}_!$. Let indeed $f \in \mathcal{L}_!(Z \,\&\, X, Y)$, we have $f\,\mathsf{m}^2_{Z,X} \in \mathcal{L}(!Z \otimes !X, Y)$ and hence

$$\mathsf{Cur}(f) = \mathsf{cur}(f\,\mathsf{m}^2_{Z,X}) \in \mathcal{L}_!(Z, X \Rightarrow Y)\,.$$

We check first that

$$\mathsf{Ev} \circ (\mathsf{Cur}(f) \,\&_!\, \mathsf{d}_X) = f \in \mathcal{L}_!(Z \,\&\, X, Y)\,.$$

We have

$$\begin{aligned} &\mathsf{Ev} \circ (\mathsf{Cur}(f) \,\&_!\, \mathsf{d}_X) \\ &\quad = \mathsf{ev}\,(\mathsf{d}_{!X\multimap Y} \otimes !X)\,(\mathsf{m}^2_{!X\multimap Y, X})^{-1}\,(\mathsf{Cur}(f) \,\&_!\, \mathsf{d}_X)^! \\ &\quad = \mathsf{ev}\,(\mathsf{d}_{!X\multimap Y} \otimes !X)\,(\mathsf{m}^2_{!X\multimap Y, X})^{-1} \\ &\qquad\quad !(\mathsf{Cur}(f) \,\&\, \mathsf{d}_X)\,!\langle !\mathsf{pr}_1, !\mathsf{pr}_2\rangle\ \mathsf{p}_{Z\&X} \quad \text{by naturality of } (\mathsf{m}^2)^{-1} \\ &\quad = \mathsf{ev}\,(\mathsf{d}_{!X\multimap Y} \otimes !X)\,(!\mathsf{Cur}(f) \otimes !\mathsf{d}_X)\,(\mathsf{m}^2_{!Z,!X})^{-1}\,!\langle !\mathsf{pr}_1, !\mathsf{pr}_2\rangle\ \mathsf{p}_{Z\&X} \\ &\quad = \mathsf{ev}\,(\mathsf{d}_{!X\multimap Y} \otimes !X)\,(!\mathsf{Cur}(f) \otimes !\mathsf{d}_X)\,(\mathsf{p}_Z \otimes \mathsf{p}_X)\,\mathsf{m}^2_{Z,X}{}^{-1} \quad \text{by Eq. (6.1)}. \end{aligned}$$

Observe that $\mathsf{d}_{!X\multimap Y}\,!\mathsf{Cur}(f) = \mathsf{Cur}(f)\,\mathsf{d}_{!Z}$ by naturality of d and that $!\mathsf{d}_X\ \mathsf{p}_X =$

$\mathsf{Id}_{!X}$ by the comonad equations and hence

$$\begin{aligned}
&\mathsf{Ev} \circ (\mathsf{Cur}(f) \mathbin{\&_!} \mathsf{d}_X) \\
&\quad = \mathsf{ev}\,((\mathsf{d}_{!X \multimap Y}\, !\mathsf{Cur}(f)\ \mathsf{p}_Z) \otimes !X)\, {\mathsf{m}^2_{Z,X}}^{-1} \\
&\quad = \mathsf{ev}\,((\mathsf{Cur}(f)\, \mathsf{d}_{!Z}\ \mathsf{p}_Z) \otimes !X)\, {\mathsf{m}^2_{Z,X}}^{-1} \quad \text{by naturality of } \mathsf{d} \\
&\quad = \mathsf{ev}\,(\mathsf{Cur}(f) \otimes !X)\, {\mathsf{m}^2_{Z,X}}^{-1} \quad \text{by a comonad equation} \\
&\quad = \mathsf{ev}\,(\mathsf{cur}(f\, \mathsf{m}^2_{Z,X}) \otimes !X)\, {\mathsf{m}^2_{Z,X}}^{-1} \\
&\quad = f\, \mathsf{m}^2_{Z,X}\, {\mathsf{m}^2_{Z,X}}^{-1} \quad \text{by monoidal closedness} \\
&\quad = f\,.
\end{aligned}$$

Next, given moreover $g \in \mathcal{L}_!(U, Z)$ we check that

$$\mathsf{Cur}(f) \circ g = \mathsf{Cur}(f \circ (g \mathbin{\&_!} \mathsf{d}_X))\,.$$

We have

$$\begin{aligned}
\mathsf{Cur}(f) \circ g &= \mathsf{cur}(f\, \mathsf{m}^2_{Z,X})\, !g\, \mathsf{p}_U \\
&= \mathsf{cur}(f\, \mathsf{m}^2_{Z,X}\, ((!g\ \mathsf{p}_U) \otimes !X)) \\
&= \mathsf{cur}(f\, !(g \mathbin{\&} X)\, \mathsf{m}^2_{!U,X}\, (\mathsf{p}_U \otimes !X))
\end{aligned}$$

and on the other hand

$$\begin{aligned}
\mathsf{Cur}(f \circ (g \mathbin{\&_!} \mathsf{d}_X)) &= \mathsf{cur}(f\, !(g \mathbin{\&_!} \mathsf{d}_X)\ \mathsf{p}_{U \& X}\ \mathsf{m}^2_{U,X}) \\
&= \mathsf{cur}(f\, !(g \mathbin{\&} \mathsf{d}_X)\, !\langle !\mathsf{pr}_1, !\mathsf{pr}_1 \rangle\ \mathsf{p}_{U \& X}\ \mathsf{m}^2_{U,X}) \\
&= \mathsf{cur}(f\, !(g \mathbin{\&} \mathsf{d}_X)\, \mathsf{m}^2_{!U,!X}(\mathsf{p}_U \otimes \mathsf{p}_X)) \quad \text{by Eq. (6.1).} \\
&= \mathsf{cur}(f\, !(g \mathbin{\&} X)\, \mathsf{m}^2_{!U,X}(\mathsf{p}_U \otimes (\mathsf{d}_X\ \mathsf{p}_X))) \\
&= \mathsf{Cur}(f) \circ g\,.
\end{aligned}$$

Last we must check that

$$\mathsf{Cur}(\mathsf{Ev}) = \mathsf{d}_{!X \multimap Y}$$

where $\mathsf{Ev} \in \mathcal{L}_!(!X \multimap Y \mathbin{\&} X, Y)$. We have

$$\begin{aligned}
\mathsf{Cur}(\mathsf{Ev}) &= \mathsf{Cur}(\mathsf{ev}\,(\mathsf{d}_{!X \multimap Y} \otimes !X)\, (\mathsf{m}^2_{!X \multimap Y, X})^{-1}) \\
&= \mathsf{cur}(\mathsf{ev}\,(\mathsf{d}_{!X \multimap Y} \otimes !X)\, (\mathsf{m}^2_{!X \multimap Y, X})^{-1}\, \mathsf{m}^2_{!X \multimap Y, X}) \\
&= \mathsf{cur}(\mathsf{ev})\, \mathsf{d}_{!X \multimap Y} \\
&= \mathsf{d}_{!X \multimap Y}
\end{aligned}$$

□

6.1.4.4 The Eilenberg-Moore category

The Eilenberg-Moore category $\mathcal{L}^!$ of ! has as objects the coalgebras of ! which are pairs $P = (\underline{P}, \mathsf{h}_P)$ where $\underline{P}$ is an object of $\mathcal{L}$ and $\mathsf{h}_P \in \mathcal{L}(\underline{P}, !\underline{P})$ satisfies the two following commutations

$$\begin{array}{ccc} \underline{P} & \xrightarrow{\mathsf{h}_P} & !\underline{P} \\ & \searrow^{\mathsf{Id}} & \downarrow{\scriptstyle \mathsf{d}_{\underline{P}}} \\ & & \underline{P} \end{array} \qquad \begin{array}{ccc} \underline{P} & \xrightarrow{\mathsf{h}_P} & !\underline{P} \\ {\scriptstyle \mathsf{h}_P}\downarrow & & \downarrow{\scriptstyle \mathsf{p}_{\underline{P}}} \\ !\underline{P} & \xrightarrow{!\mathsf{h}_P} & !!\underline{P} \end{array}$$

Given objects P and Q of that category, $\mathcal{L}^!(P, Q)$ is the set of all $f \in \mathcal{L}(\underline{P}, \underline{Q})$ such that the following diagram commutes

$$\begin{array}{ccc} \underline{P} & \xrightarrow{f} & \underline{Q} \\ {\scriptstyle \mathsf{h}_P}\downarrow & & \downarrow{\scriptstyle \mathsf{h}_Q} \\ !\underline{P} & \xrightarrow{!f} & !\underline{Q} \end{array}$$

If X is an object of $\mathcal{L}$, then $\mathsf{E}(X) = (!X, \mathsf{p}_X)$ is clearly an object of $\mathcal{L}^!$. If $f \in \mathcal{L}_!(X, Y) = \mathcal{L}(!X, Y)$ then we set

$$\mathsf{E}(f) = f^! = !f \ \mathsf{p}_X \in \mathcal{L}(!X, !Y)\,.$$

Intuitively, one should understand the Kleisli category $\mathcal{L}_!$ as the category of *free* coalgebras of !, this is made explicit by the following theorem which shows that $\mathcal{L}_!$ can be seen as a full subcategory of $\mathcal{L}^!$.

Theorem 6.1.20. *The operation* E *is a full and faithful functor* $\mathcal{L}_! \to \mathcal{L}^!$.

Proof. We need first to prove that, given $f \in \mathcal{L}(!X, Y)$, one has $f^! \in \mathcal{L}^!(\mathsf{E}(X), \mathsf{E}(Y))$. This follows from

$$\begin{aligned} \mathsf{d}_Y \ f^! &= \mathsf{d}_Y \ !f \ \mathsf{p}_X \\ &= f \ \mathsf{d}_X \ \mathsf{p}_X \quad \text{by naturality of } \mathsf{d} \\ &= f \end{aligned}$$

and

$$\begin{aligned} \mathsf{p}_Y \ f^! &= !!f \ \mathsf{p}_{!X} \ \mathsf{p}_X \quad \text{by naturality of } \mathsf{p} \\ &= !!f !\mathsf{p}_X \ \mathsf{p}_X \\ &= !(f^!) \ \mathsf{p}_X \ . \end{aligned}$$

Next we have $\mathsf{E}(\mathsf{d}_X) = !\mathsf{d}_X \ \mathsf{p}_X = \mathsf{Id}_{!X}$ and, given $f \in \mathcal{L}(!X, Y)$ and $g \in \mathcal{L}(!Y, Z)$,

we have

$$\begin{aligned}\mathsf{E}(g \circ f) &= !(g\,!f\,\mathsf{p}_X)\ \mathsf{p}_X \\ &= !g\,!!f\,!\mathsf{p}_X\ \mathsf{p}_X \\ &= !g\,!!f\ \mathsf{p}_{!X}\ \mathsf{p}_X \\ &= !g\ \mathsf{p}_Y\ !f\ \mathsf{p}_X \quad \text{by naturality of } \mathsf{p} \\ &= \mathsf{E}(g)\,\mathsf{E}(f)\,.\end{aligned}$$

Next observe that $\mathsf{d}_Y\,\mathsf{E}(f) = f$ (for $f \in \mathcal{L}(!X, Y)$) so that the functor E is faithful. Last let $f \in \mathcal{L}^!(\mathsf{E}(X), \mathsf{E}(Y))$ so that $\mathsf{d}_Y\,f \in \mathcal{L}(!X, Y)$, we have

$$\begin{aligned}\mathsf{E}(\mathsf{d}_Y\,f) &= !(\mathsf{d}_Y\,f)\ \mathsf{p}_X \\ &= !\mathsf{d}_Y\,!f\ \mathsf{p}_X \\ &= !\mathsf{d}_Y\ \mathsf{p}_Y\ f \quad \text{since } f \in \mathcal{L}^!(\mathsf{E}(X), \mathsf{E}(Y)) \\ &= f\end{aligned}$$

which proves that the functor E is full. □

Theorem 6.1.21. *The category $\mathcal{L}^!$ is cocartesian.*

Proof. Let $(P_i)_{i\in I}$ be a finite family of objects of $\mathcal{L}$ and let $X = \oplus_{i\in I} \underline{P_i}$. For each $j \in J$ we have $!\mathsf{in}_j\,\mathsf{h}_{P_j} \in \mathcal{L}(\underline{P_j}, !X)$ and hence we can set

$$h = \left[!\mathsf{in}_j\,\mathsf{h}_{P_j}\right]_{j\in J} \in \mathcal{L}(X, !X)\,.$$

We prove that this is a !-coalgebra structure on X. First we have

$$\begin{aligned}\mathsf{d}_X\,h &= \left[\mathsf{d}_X\,!\mathsf{in}_j\,\mathsf{h}_{P_j}\right]_{j\in J} \\ &= \left[\mathsf{in}_j\ \mathsf{d}_{\underline{P_j}}\,\mathsf{h}_{P_j}\right]_{j\in J} \quad \text{by naturality of } \mathsf{d} \\ &= \left[\mathsf{in}_j\right]_{j\in J} \quad \text{since each } P_j \text{ is a coalgebra} \\ &= \mathsf{Id}_X\ .\end{aligned}$$

Next we have

$$\begin{aligned}\mathsf{p}_X\ h &= \left[\mathsf{p}_X\ !\mathsf{in}_j\,\mathsf{h}_{P_j}\right]_{j\in J} \\ &= \left[!!\mathsf{in}_j\ \mathsf{p}_{\underline{P_j}}\ \mathsf{h}_{P_j}\right]_{j\in J} \quad \text{by naturality of } \mathsf{p} \\ &= \left[!!\mathsf{in}_j\,!\mathsf{h}_{P_j}\,\mathsf{h}_{P_j}\right]_{j\in J} \quad \text{since each } P_j \text{ is a coalgebra}\end{aligned}$$

and

$$\begin{aligned}!h\,h &= \left[!h\,!\mathsf{in}_j\,\mathsf{h}_{P_j}\right]_{j\in J} \\ &= \left[!(h\ \mathsf{in}_j)\,\mathsf{h}_{P_j}\right]_{j\in J} \\ &= \left[!(!\mathsf{in}_j\,\mathsf{h}_{P_j})\,\mathsf{h}_{P_j}\right]_{j\in J} \quad \text{by definition of } h\end{aligned}$$

so that $\mathsf{p}_X\ h = !h\,h$ which shows that (X,h) is a !-coalgebra.

Observe that $\mathsf{in}_i \in \mathcal{L}^!(P_i,(X,h))$ for all $i \in I$ since $h\ \mathsf{in}_i = !\mathsf{in}_i\,\mathsf{h}_{P_i}$ by definition of h.

We prove that (X,h), together with the injections in_i, is the coproduct of the P_i's in $\mathcal{L}^!$ so let $(f_i \in \mathcal{L}^!(P_i,P))_{i\in I}$. We prove that $f = [f_i]_{i\in I} \in \mathcal{L}(X,\underline{P})$ is in $\mathcal{L}^!((X,h),P)$: we have

$$\begin{aligned}
\mathsf{h}_P\,f &= [\mathsf{h}_P\,f_i]_{i\in I}\\
&= [!f_i\,\mathsf{h}_{P_i}]_{i\in I} \quad \text{since } f_i \in \mathcal{L}^!(P_i,P)\\
&= [!f\,!\mathsf{in}_i\,\mathsf{h}_{P_i}]_{i\in I} \quad \text{by definition of } f\\
&= !f\ [!\mathsf{in}_i\,\mathsf{h}_{P_i}]_{i\in I}\\
&= !f\,h
\end{aligned}$$

so that f is the unique element of $\mathcal{L}^!((X,h),P)$ such that $f\ \mathsf{in}_i = f_i$ for all $i \in I$, which shows that $(X,h) = \oplus_{i\in I} P_i$ (with injections $(\mathsf{in}_i)_{i\in I}$). □

Lemma 6.1.22. *The object* 1 *of* $\mathcal{L}$ *has a* !*-coalgebra structure which turns it into the terminal object of* $\mathcal{L}^!$.

Proof. We set $\mathsf{h}_1 = !(\mathsf{m}^0)^{-1}\ \mathsf{p}_\top\ \mathsf{m}^0$ which is typed as follows

$$1 \xrightarrow{\mathsf{m}^0} !\top \xrightarrow{\mathsf{p}_\top} !!\top \xrightarrow{!(\mathsf{m}^0)^{-1}} !1\,.$$

We have

$$\begin{aligned}
\mathsf{d}_1\,\mathsf{h}_1 &= (\mathsf{m}^0)^{-1}\,\mathsf{d}_{!\top}\ \mathsf{p}_\top\ \mathsf{m}^0\\
&= (\mathsf{m}^0)^{-1}\,\mathsf{m}^0 = \mathsf{Id}
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{p}_1\ \mathsf{h}_1 &= !!(\mathsf{m}^0)^{-1}\ \mathsf{p}_{!\top}\ \mathsf{p}_\top\ \mathsf{m}^0\\
&= !!(\mathsf{m}^0)^{-1}\,!\mathsf{p}_\top\ \mathsf{p}_\top\ \mathsf{m}^0\\
&= !!(\mathsf{m}^0)^{-1}\,!\mathsf{p}_\top\,!\mathsf{m}^0\,!(\mathsf{m}^0)^{-1}\ \mathsf{p}_\top\ \mathsf{m}^0\\
&= !\mathsf{h}_1\,\mathsf{h}_1
\end{aligned}$$

which shows that $(1,\mathsf{h}_1)$ is an !-coalgebra.

Let P be an arbitrary coalgebra, and let $\mathsf{w}_P = (\mathsf{m}^0)^{-1}\,!\mathsf{t}_{\underline{P}}\,\mathsf{h}_P \in \mathcal{L}(\underline{P},1)$, typed as follows

$$\underline{P} \xrightarrow{\mathsf{h}_P} !\underline{P} \xrightarrow{!\mathsf{t}_{\underline{P}}} !\top \xrightarrow{(\mathsf{m}^0)^{-1}} 1$$

We check that $\mathsf{w}_P \in \mathcal{L}^!(P,1)$: we have

$$\begin{aligned}
\mathsf{h}_1\,\mathsf{w}_P &= !(\mathsf{m}^0)^{-1}\ \mathsf{p}_\top\ \mathsf{m}^0\,(\mathsf{m}^0)^{-1}\,!\mathsf{t}_{\underline{P}}\,\mathsf{h}_P \\
&= !(\mathsf{m}^0)^{-1}\ \mathsf{p}_\top\ !\mathsf{t}_{\underline{P}}\,\mathsf{h}_P \\
&= !(\mathsf{m}^0)^{-1}\,!!\mathsf{t}_{\underline{P}}\ \mathsf{p}_{\underline{P}}\ \mathsf{h}_P \quad \text{by naturality of } \mathsf{p} \\
&= !(\mathsf{m}^0)^{-1}\,!!\mathsf{t}_{\underline{P}}\,!\mathsf{h}_P\,\mathsf{h}_P \quad \text{because } P \text{ is a coalgebra} \\
&= !\mathsf{w}_P\,\mathsf{h}_P\,.
\end{aligned}$$

Let now $f \in \mathcal{L}^!(P,1)$, which means that $f \in \mathcal{L}(\underline{P},1)$ and $!f\,\mathsf{h}_P = \mathsf{h}_1\,f = !(\mathsf{m}^0)^{-1}\ \mathsf{p}_\top\ \mathsf{m}^0\,f$, that is $!(\mathsf{m}^0\,f)\,\mathsf{h}_P = \mathsf{p}_\top\ \mathsf{m}^0\,f$. We have

$$\begin{aligned}
\mathsf{m}^0\,f &= !\mathsf{d}_\top\ \mathsf{p}_\top\ \mathsf{m}^0\,f \quad \text{since } !\mathsf{d}_\top\ \mathsf{p}_\top = \mathsf{Id}_{!\top} \\
&= !\mathsf{d}_\top\,!(\mathsf{m}^0\,f)\,\mathsf{h}_P \\
&= !(\mathsf{d}_\top\,\mathsf{m}^0\,f)\,\mathsf{h}_P \\
&= !\mathsf{t}_{\underline{P}}\,\mathsf{h}_P \quad \text{since } \top \text{ is the terminal object of } \mathcal{L}
\end{aligned}$$

and hence $f = \mathsf{w}_P$, which shows that 1 is the terminal object of $\mathcal{L}^!$. □

Let P_1, P_2 be objects of $\mathcal{L}^!$. We define $h \in \mathcal{L}(\underline{P_1} \otimes \underline{P_2}, !(\underline{P_1} \otimes \underline{P_2}))$ as the following composition of morphisms

$$\underline{P_1} \otimes \underline{P_2} \xrightarrow{\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}} !\underline{P_1} \otimes !\underline{P_2} \xrightarrow{\mu^2_{\underline{P_1},\underline{P_2}}} !(\underline{P_1} \otimes \underline{P_2})$$

Lemma 6.1.23. *Equipped with the morphism h, the object $\underline{P_1} \otimes \underline{P_2}$ is an $!$-coalgebra that we denote as $P_1 \otimes P_2$.*

Proof. We have

$$\begin{aligned}
\mathsf{d}_{\underline{P_1}\otimes\underline{P_2}}\,h &= \mathsf{d}_{\underline{P_1}\otimes\underline{P_2}}\,\mu^2_{\underline{P_1},\underline{P_2}}\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \\
&= (\mathsf{d}_{\underline{P_1}} \otimes \mathsf{d}_{\underline{P_2}})\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \\
&= \mathsf{Id}
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{p}_{\underline{P_1}\otimes\underline{P_2}}\,h &= \mathsf{p}_{\underline{P_1}\otimes\underline{P_2}}\ \mu^2_{\underline{P_1},\underline{P_2}}\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \\
&= !\mu^2_{\underline{P_1},\underline{P_2}}\,\mu^2_{!\underline{P_1},!\underline{P_2}}\,(\mathsf{p}_{\underline{P_1}} \otimes \mathsf{p}_{\underline{P_2}})\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \quad \text{by Proposition 6.1.9} \\
&= !\mu^2_{\underline{P_1},\underline{P_2}}\,\mu^2_{!\underline{P_1},!\underline{P_2}}\,(!\mathsf{h}_{P_1} \otimes !\mathsf{h}_{P_2})\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \\
&\qquad\qquad \text{since } P_1 \text{ and } P_2 \text{ are coalgebras} \\
&= !\mu^2_{\underline{P_1},\underline{P_2}}\,!(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2})\,\mu^2_{!\underline{P_1},!\underline{P_2}}\,(\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2}) \quad \text{by naturality of } \mu^2 \\
&= !h\,f\,.
\end{aligned}$$

□

Lemma 6.1.24. *Let $(f_i \in \mathcal{L}^!(P_i,Q_i))_{i=1,2}$. Then $f_1 \otimes f_2 \in \mathcal{L}^!(P_1 \otimes P_2, Q_1 \otimes Q_2)$.*

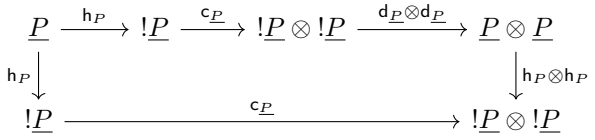
Proof. By definition of coalgebra morphisms and naturality of μ^2. □

For any object P of $\mathcal{L}^!$, one defines a generalized contraction morphism $\mathsf{c}_P \in \mathcal{L}(\underline{P}, \underline{P} \otimes \underline{P})$ as

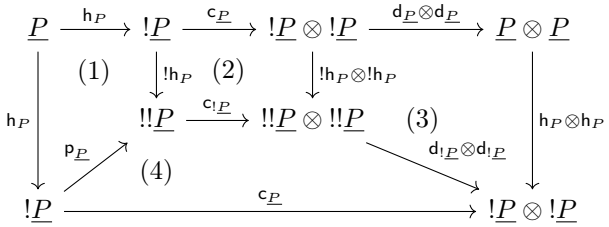
$$\underline{P} \xrightarrow{\mathsf{h}_P} !\underline{P} \xrightarrow{\mathsf{c}_{\underline{P}}} !\underline{P} \otimes !\underline{P} \xrightarrow{\mathsf{d}_{\underline{P}} \otimes \mathsf{d}_{\underline{P}}} \underline{P} \otimes \underline{P}$$

Our goal now is to prove that this morphism of $\mathcal{L}$ is actually a coalgebra morphisms from P to the coalgebra $P \otimes P$ we juste defined, and for this we follow [**Mellies**]. The main observation is that, in spite of the fact that $\mathsf{h}_P\, \mathsf{d}_{\underline{P}}$ has no reason to be equal to $\mathsf{Id}_{!\underline{P}}$, we have the following property.

Lemma 6.1.25. *For any object P of $\mathcal{L}^!$, the following diagram commutes.*

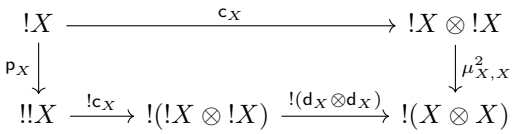


Proof. One proceeds by diagram chasing in

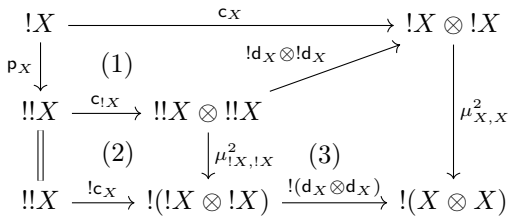


where (1) commutes because P is a coalgebra, (2) commutes by naturality of c_X in X, (3) commutes by naturality of d_X in X and (4) commutes by Proposition 6.1.14 and by the fact that $\mathsf{d}_{!\underline{P}}\ \mathsf{p}_{\underline{P}} = \mathsf{Id}$. □

Lemma 6.1.26. *For any object X of $\mathcal{L}$, the following diagram commute*



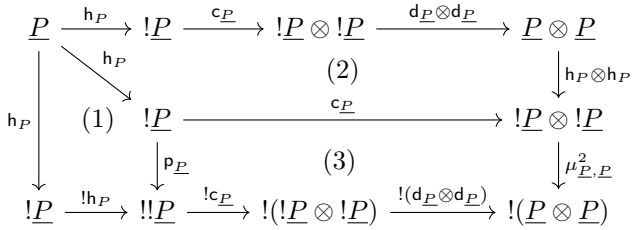
Proof. Diagram chasing in



where (1) commutes by Proposition 6.1.14 and by the fact that $!\mathsf{d}_{\underline{P}}\,\mathsf{p}_{\underline{P}} = \mathsf{Id}$, (2) commutes by Proposition 6.1.15 and (3) commutes by naturality of μ^2. □

Theorem 6.1.27. *For any object P of $\mathcal{L}^!$, one has $\mathsf{c}_P \in \mathcal{L}^!(P, P \otimes P)$.*

Proof. Diagram chasing in



where (1) commutes because P is a coalgebra, (2) commutes by Lemma 6.1.25 and (3) commutes by Lemma 6.1.26. □

We define $\mathsf{pr}_i^{(!)} \in \mathcal{L}(\underline{P_1} \otimes \underline{P_2}, \underline{P_i})$ for $i = 1, 2$ as the following compositions of morphisms

$$\underline{P_1} \otimes P_2 \xrightarrow{\underline{P_1} \otimes \mathsf{w}_{P_2}} \underline{P_1} \otimes 1 \xrightarrow{\rho} \underline{P_1}$$

$$\underline{P_1} \otimes P_2 \xrightarrow{\mathsf{w}_{P_1} \otimes \underline{P_2}} 1 \otimes \underline{P_2} \xrightarrow{\lambda} \underline{P_1}$$

Lemma 6.1.28. *The morphism $\mathsf{pr}_i^{(!)}$ of $\mathcal{L}$ belongs to $\mathcal{L}^!(P_1 \otimes P_2, P_i)$.*

The proof is straightforward and uses the fact that morphisms w_{P_i} are coalgebra morphisms.

Theorem 6.1.29. *Given objects P_1 and P_2 of $\mathcal{L}^!$, the triple $(P_1 \otimes P_2, \mathsf{pr}_1^{(!)}, \mathsf{pr}_2^{(!)})$ is the cartesian product of P_1 and P_2 in $\mathcal{L}^!$.*

Proof. Let $(f_i \in \mathcal{L}^!(Q, P_i))_{i=1,2}$, then we define $\langle f_1, f_2 \rangle^{(!)} \in \mathcal{L}(\underline{Q}, \underline{P_1} \otimes \underline{P_2})$ as

$$\underline{Q} \xrightarrow{\mathsf{c}_Q} \underline{Q} \otimes \underline{Q} \xrightarrow{f_1 \otimes f_2} \underline{P_1} \otimes \underline{P_2}$$

By Theorem 6.1.27 and Lemma 6.1.24, we have $\langle f_1, f_2 \rangle^{(!)} \in \mathcal{L}^!(Q, P_1 \otimes P_2)$. The fact that $\mathsf{pr}_i^{(!)} \langle f_1, f_2 \rangle^{(!)} = f_i$ results from the naturality of w_P in P and from the fact that $(\underline{Q}, \mathsf{c}_Q, \mathsf{w}_Q)$ is a comonoid. Let $g \in \mathcal{L}^!(R, Q)$, then

$$\begin{aligned}
\langle f_1, f_2 \rangle^{(!)}\, g &= (f_1 \otimes f_2)\, \mathsf{c}_Q\, g \\
&= (f_1 \otimes f_2)\, (\mathsf{d}_{\underline{Q}} \otimes \mathsf{d}_{\underline{Q}})\, \mathsf{c}_{\underline{Q}}\, \mathsf{h}_Q\, g \\
&= (f_1 \otimes f_2)\, (\mathsf{d}_{\underline{Q}} \otimes \mathsf{d}_{\underline{Q}})\, \mathsf{c}_{\underline{Q}}\, !g\, \mathsf{h}_R \\
&= (f_1 \otimes f_2)\, (\mathsf{d}_{\underline{Q}} \otimes \mathsf{d}_{\underline{Q}})\, (!g \otimes !g)\, \mathsf{c}_{\underline{R}}\, \mathsf{h}_R \\
&= ((f_1\, g) \otimes (f_2\, g))\, (\mathsf{d}_R \otimes \mathsf{d}_R)\, \mathsf{c}_{\underline{R}}\, \mathsf{h}_R \\
&= \langle f_1\, g, f_2\, g \rangle^{(!)}\,.
\end{aligned}$$

Last, with $Q = P_1 \otimes P_2$, we have

$$\begin{aligned}\langle \mathsf{pr}_1^{(!)}, \mathsf{pr}_2^{(!)} \rangle^{(!)} &= (\mathsf{pr}_1^{(!)} \otimes \mathsf{pr}_2^{(!)})\, \mathsf{c}_{P_1 \otimes P_2} \\ &= (\rho \otimes \lambda)\, (\underline{P_1} \otimes \mathsf{w}_{P_2} \otimes \mathsf{w}_{P_1} \otimes \underline{P_2})\, (\mathsf{d}_{\underline{P_1} \otimes \underline{P_2}} \otimes \mathsf{d}_{\underline{P_1} \otimes \underline{P_2}})\, \mathsf{c}_{\underline{P_1} \otimes \underline{P_2}}\, \mathsf{h}_{P_1 \otimes P_2} \\ &= (\rho \otimes \lambda)\, (\underline{P_1} \otimes \mathsf{w}_{P_2} \otimes \mathsf{w}_{P_1} \otimes \underline{P_2})\, (\mathsf{d}_{\underline{P_1} \otimes \underline{P_2}} \otimes \mathsf{d}_{\underline{P_1} \otimes \underline{P_2}})\, \mathsf{c}_{\underline{P_1} \otimes \underline{P_2}}\, \mu^2_{\underline{P_1}, \underline{P_2}}\, (\mathsf{h}_{P_1} \otimes \mathsf{h}_{P_2})\end{aligned}$$

□

6.1.4.5 The models of free comodules: Girard's construction

6.1.5 Free exponentials and Lafont categories

6.1.5.1 The Melliès-Tabareau-Tasson formula

6.1.6 Interpreting the sequent calculus in a Seely category

Remember that a sequent is an expression $\vdash \Gamma$ where $\Gamma = A_1, \ldots, A_n$ is a finite sequence of formulas of Linear Logic.

We assume to be given a categorical model $\mathcal{L}$ of Linear Logic.

First, with each formula A of Linear Logic we can assign an object $[A]$ of $\mathcal{L}$ by an obvious induction:

$$\begin{array}{cc} [1] = 1 & [A_1 \otimes A_2] = [A_1] \otimes [A_2] \\ [\bot] = \bot & [A_1 \parr A_2] = [A_1] \parr [A_2] \\ [0] = 0 & [A_1 \oplus A_2] = [A_1] \oplus [A_2] \\ [\top] = \top & [A_1 \,\&\, A_2] = [A_1] \,\&\, [A_2] \\ [!A] = ![A] & [?A] = ?[A] \end{array}$$

The semantics a sequence $\Gamma = (A_1, \ldots, A_n)$ is parameterized by a monoidal tree τ of degree n: we set $[\Gamma]^\tau = \mathsf{T}^{\parr}_\tau([A_1], \ldots, [A_n])$.

Given a proof π of the sequent Γ, we define for each well-formed tree τ of degree n a morphism

$$[\pi]^\tau \in \mathcal{L}(1, [\Gamma]^\tau)$$

in such a way that, for any other well-formed tree of degree n, one has

$$[\pi]^{\tau'} = \varphi^{\parr}_{\tau,\tau'}\, [\pi]^\tau$$

Chapter 7

Coh

Uniform coherent spaces were defined by Girard in [23] as a variant of Scott domains giving a denotational semantics to system F. However their main interest was the analogy they bare with linear algebra that led Girard to discover firstly their linear structure from which he could then derive the definition of linear logic.

In this chapter we will stick to the historical terminology and call uniform coherent spaces just *coherent spaces*. Most proofs will be sketched, when not left to the reader. We will also use the convenient language of category theory, the reader is refered to the chapter 6 for the basic definitions and properties.

7.1 Coherent spaces

7.1.1 The coherence relation

A *coherent space* E is a structure

$$E = (|E|, \frown_E)$$

where $|E|$ is a set (which can be assumed to be at most countable) and $\frown_E$ is a binary reflexive and symmetric relation on $|E|$ called *coherence*.

We use the following definitions and notations:

Strict coherence: $\smile_E = (\frown_E \cap \neq)$, that is $a \smile_E a'$ iff ($a \frown_E a'$ and $a \neq a'$);

Incoherence: $\asymp_E = \neg \smile_E$, that is $a \asymp_E a'$ iff ($a \not\frown_E a'$ or $a = a'$);

Strict incoherence: $\frown_E = \neg \frown_E$, that is $a \frown_E a'$ iff $a \not\frown_E a'$.

Note that any of these four relations caracterises the three others.

An easy consequence of these definitions, that will be used in the sequel is that the intersection of $\frown_E$ and $\asymp_E$ is equality:

$$\forall a, a' \in |E|,\ a = a' \text{ iff } a \frown_E a' \text{ and } a \asymp_E a'$$

A *clique* of E is a set of pairwise coherent points of $|E|$; we denote by $\mathrm{Cl}(E)$ the set of cliques:

$$\mathrm{Cl}(E) = \{u \subset |E|,\ \forall a, a' \in u,\ a \frown_E a'\}$$

We note $u \sqsubset E$ when u is a clique of E, that is $u \sqsubset E$ iff $u \in \mathrm{Cl}(E)$. The following properties are immediately derived from the definition.

Proposition 7.1.1 (Elementary properties of cliques)**.** *Let E be a coherent space. We have:*

- $\emptyset \sqsubset E$ *so that* $\mathrm{Cl}(E)$ *is never empty (even when the web is empty).*
- *Singletons are cliques: for any $a \in |E|$, $\{a\} \sqsubset E$.*
- $\mathrm{Cl}(E)$ *is downward closed for inclusion: if $u \sqsubset E$ and $u' \subset u$ then $u' \sqsubset E$. For that reason we call u' a* subclique *of u.*
- $\mathrm{Cl}(E)$ *is closed by directed unions: if U is a directed family of cliques of E, then $\bigcup U$ is a clique. In particular any clique u is the directed union of its finite subcliques:*

$$u = \bigcup \{u_0 \in \mathrm{Cl}(E),\ u_0 \subset_{\mathrm{fin}} u\}\,.$$

These three properties imply in particular that the space $\mathrm{Cl}(E)$ ordered by inclusion is a Scott domain (see section 8.1). However it is a special case of Scott domain that satisfies **binary completeness**: say that a set U of cliques is *compatible* if the union of any two cliques in U is a clique; binary completeness states that if U is any compatible set of cliques then $\bigcup U$ is a clique. Note that if U is directed then it is compatible, but the converse is not necessarily true, thus binary completeness is indeed stronger than closure by directed union.

Proposition 7.1.2 (Dual of a coherent space)**.** *The space $E^\perp = (|E|, \asymp_E)$ is a coherent space. The cliques of $E^\perp$ are sets of pairwise incoherent points and are called the* anticliques *of E.*

The dual of the dual $(E^\perp)^\perp$ is denoted $E^{\perp\perp}$. By definition of incoherence we have:

$$E^{\perp\perp} = E\,.$$

Ref to *-autonomous category section — Laurent

The last property exhibits a canonical duality in coherent spaces: we will see that it is the very reason why the category of coherent spaces is $*$-autonomous.

7.1.2 Clique spaces

The notion of clique spaces is an alternative, and more modern way, for defining coherent spaces that is based on the fact that, given a coherent space E, cliques and anticliques of E intersect in at most one point:

$$\text{For all } u \sqsubset E, u' \sqsubset E^\perp,\ \mathrm{Card}(u \cap u') \leq 1$$

Indeed if $a, a' \in u \cap u'$ then $a \supset_E a'$ because u is a clique, and $a \asymp_E a'$ because u' is an anticlique, thus $a = a'$.

When u and u' are two subsets of a set X we say that u and u' are *orthogonal* and note $u \perp u'$ if their intersection has at most one element:

$$u \perp u' \text{ iff } \mathrm{Card}(u \cap u') \leq 1$$

Thus any clique and anticlique of a coherent space are orthogonal.

If U is a family of subsets of X we denote by $U^\perp$ the set of subsets of X othogonal to all elements of U:

$$U^\perp = \{u' \subset X, \forall u \in U, u \perp u'\}$$

Proposition 7.1.3. *Let E be a coherent space; a subset $u \subset |E|$ is a clique iff it is orthogonal to all anticliques:*

$$u \sqsubset E \text{ iff } \forall u' \sqsubset E^\perp, u \perp u'$$

Thus we have:

$$\mathrm{Cl}(E) = \mathrm{Cl}(E^\perp)^\perp$$

Proof. The only if part is immediate. Suppose u is orthogonal to any anticlique and let $a, a' \in u$. If $a \asymp_E a'$ then $\{a, a'\}$ is an anticlique of E, and since $u \perp \{a, a'\}$ we have $\mathrm{Card}(u \cap \{a, a'\}) \leq 1$, but since $a, a' \in u$, $u \cap \{a, a'\}$ cannot be void, thus is a singleton. In summary, if $a \asymp_E a'$ then $a = a'$, which by definition of $\asymp_E$ is equivalent to $a \supset_E a'$. □

Note that, since $E^{\perp\perp} = E$, we also have:

$$\mathrm{Cl}(E^\perp) = \mathrm{Cl}(E)^\perp$$

from which we deduce

$$\mathrm{Cl}(E)^{\perp\perp} = \mathrm{Cl}(E)$$

This property motivates the following definition: a *clique space* E is a structure

$$E = (|E|, C_E)$$

where $|E|$ is a set and C_E is a family of subsets of $|E|$ that is equal to its biorthogonal:

$$C_E{}^{\perp\perp} = C_E$$

Proposition 7.1.4. *Let $E = (|E|, C_E)$ be a clique space. Then we have:*

- $\emptyset \in C_E$.
- *For any $a \in |E|$, $\{a\} \in C_E$*
- *C_E is downward closed for inclusion.*
- *C_E is closed by directed union.*

Proof. Clearly $\emptyset \perp u'$ and $\{a\} \perp u'$ for any $u' \in C_E^\perp$ thus $\emptyset, \{a\} \in {C_E}^{\perp\perp} = C_E$.

Let $u \in C_E$, $v \subset u$ and $u' \in C_E^\perp$; since $v \subset u$, $\mathrm{Card}(v \cap u') \leq \mathrm{Card}(u \cap u') \leq 1$ thus $v \perp u'$. Therefore $v \in {C_E}^{\perp\perp} = C_E$.

Let U be a directed family of elements of C_E. Let $u' \in C_E^\perp$ and $a_1, a_2 \in \bigcup U \cap u'$. Since $a_i \in \bigcup U$ there are $u_1, u_2 \in U$ such that $a_1 \in u_1$ and $a_2 \in u_2$ but since U is directed there is $u \in U$ such that $a_1, a_2 \in u$. As $U \subset C_E$ we have $u \perp u'$ that is $\mathrm{Card}(u \cap u') \leq 1$. But $a_1, a_2 \in u \cap u'$, thus $a_1 = a_2$ and we have proved that $\mathrm{Card}(\bigcup U \cap u') \leq 1$, that is $\bigcup U \perp u'$. Thus $\bigcup U \in {C_E}^{\perp\perp} = C_E$. □

The following theorem expresses the fact that clique spaces are really an alternative definition for coherent spaces:

Theorem 7.1.5. *If E is a coherent space then $\mathsf{Clique}(E) = (|E|, \mathrm{Cl}(E))$ is a clique space. Conversely if E is a clique space, we set $a \frown_E a'$ iff $\{a, a'\} \in C_E$. Then $\mathsf{Coh}(E) = (|E|, \frown_E)$ is a coherent space.*

Furthermore the two operations Coh and Clique are inverse of each other: for any coherent space E, $\mathsf{Coh}(\mathsf{Clique}(E)) = E$ and for any clique space E, $\mathsf{Clique}(\mathsf{Coh}(E)) = E$.

Proof. The fact that $\mathsf{Clique}(E)$ is a clique space is immediate consequence of the proposition 7.1.3. Conversely suppose E is a clique space and define $\frown_E$ as above. Since it is obviously symmetric we just have to show that $\frown_E$ is reflexive, that is that $\{a\} \in C_E$ for any $a \in |E|$ which is proved in the proposition 7.1.4.

Suppose now that $E = (|E|, \frown)$ is a coherent space and define the relation $\frown_E$ by $a \frown_E a'$ iff $\{a, a'\} \in \mathrm{Cl}(E)$. We thus have $\mathsf{Coh}(\mathsf{Clique}(E)) = (|E|, \frown_E)$ and we have to show that $\frown_E = \frown$: let $a, a' \in |E|$, then $a \frown_E a'$ iff $\{a, a'\} \in \mathrm{Cl}(E)$ iff $a \frown a'$.

Conversely suppose $E = (|E|, C_E)$ is a clique space; define $\mathrm{Cl}(E)$ to be the set of cliques of the coherent space $\mathsf{Coh}(E) = (|E|, \frown_E)$ where $\frown_E$ is defined by $a \frown_E a'$ iff $\{a, a'\} \in C_E$. We have to show that $C_E = \mathrm{Cl}(\mathsf{Coh}(E))$.

Let $u \in C_E$ and $a, a' \in u$; then $\{a, a'\} \in C_E$ thus $a \frown_E a'$ which shows that u is a clique: $u \in \mathrm{Cl}(\mathsf{Coh}(E))$. Conversely let $u \in \mathrm{Cl}(\mathsf{Coh}(E))$, $u' \in C_E^\perp$ and $a_1, a_2 \in u \cap u'$. Since $a_1, a_2 \in u$ we have $a_1 \frown_E a_2$ thus $\{a_1, a_2\} \in C_E$. But $u' \in C_E^\perp$ thus $\{a_1, a_2\} \cap u'$ has at most one element, thus $a_1 = a_2$. Therefore $u \perp u'$ so that $u \in {C_E}^{\perp\perp} = C_E$. □

generalize to discrete spaces? — Olivier

Example 7.1.6 (The space of booleans). The coherent space of booleans B is defined by: $|B| = \{0, 1\}$ is the two-points set, and $a \frown_B b$ iff $a = b$. The space B has three cliques that we will denote by $\perp = \emptyset$, $\mathbf{F} = \{0\}$ and $\mathbf{V} = \{1\}$ to emphasize the fact that it is isomorphic to the flat domain of booleans.

7.2 The cartesian closed structure of coherent spaces

This section is here mostly for historical reasons: we will briefly define a first notion of morphism between coherent spaces, *stable functions*, that makes the category of coherent spaces a model of typed lambda-calculus. Most proofs are straigthforward and left to the reader.

We will end up with the origin of linear logic: the fact that stable functions can be decomposed through *linear functions* and the *exponential* space as expressed by the famous isomorphism:

$$X \to Y \simeq !X \multimap Y$$

7.2.1 Stable functions

Let X and Y be coherent spaces. A *stable function* from X to Y is a map $F : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ satisfying:

Continuity: F is monotone (for inclusion) and commutes with directed unions: if U is a directed family of cliques in X then:

$$F\left(\bigcup U\right) = \bigcup_{u \in U} F(u)$$

Note that since F is monotone, the family $(F(u))_{u\in U}$ is directed in $\mathrm{Cl}(Y)$ thus the right member is a clique of Y.

Stability: F commutes with compatible intersections: if u and u' are such that $u \cup u' \sqsubset X$ then:

$$F(u \cap u') = F(u) \cap F(u')$$

Continuity states that F is continuous in the Scott sense. Stability was first introduced by Berry [7] as a property of Scott-continuous functions expressing a kind of determinism of certain computable functions. Stability was discovered independently by Girard who uses it as we will see to endow the space of stable functions with the structure of coherent space.

One easily checks that the composition of two stable functions is stable and that the identity function: $\mathsf{Id}_X : \mathrm{Cl}(X) \to \mathrm{Cl}(X)$ is stable, thus that coherent spaces with stable functions form a category. The hom set of stable functions from X to Y will be denoted $\mathsf{Stable}(X, Y)$ and we will sometimes write $F : X \to Y$ for $F \in \mathsf{Stable}(X, Y)$. If F is bijective and its reciprocal $F^{-1} : \mathrm{Cl}(Y) \to \mathrm{Cl}(X)$ is stable we say that F is a stable isomorphism.

A map $F : \mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1) \to \mathrm{Cl}(X)$ is *bi-stable if* it is stable in each of its variables, that is if for each $x_i \sqsubset X_i$ the map $F_{x_i} = \lambda x_{\bar{\imath}}\, F(x_0, x_1) : \mathrm{Cl}(X_{\bar{\imath}}) \to \mathrm{Cl}(X)$ is stable (where $\bar{\imath} = 1 - i$). This definition extends naturally to functions of arity n.

Example 7.2.1 (Parallel-or). The function $\mathsf{Por} : \mathrm{Cl}(B) \times \mathrm{Cl}(B) \to \mathrm{Cl}(B)$ (where B is the boolean space defined in example 7.1.6) is the paradigmatic function that is continuous but not (bi-)stable. It is defined by:

$$\begin{aligned} \mathsf{Por}(x, \mathbf{V}) = \mathsf{Por}(\mathbf{V}, x) = \mathbf{V} \text{ for any } x \sqsubset B \\ \mathsf{Por}(\mathbf{F}, \mathbf{F}) = \mathbf{F} \\ \mathsf{Por}(\bot, \bot) = \mathsf{Por}(\bot, \mathbf{F}) = \mathsf{Por}(\mathbf{F}, \bot) = \bot \end{aligned}$$

It is not stable because $\mathsf{Por}(\bot, \mathbf{V}) = \mathsf{Por}(\mathbf{V}, \bot) = \mathbf{V}$ but the intersection of $(\bot, \mathbf{V})$ and $(\mathbf{V}, \bot)$ is $(\bot, \bot)$ and $\mathsf{Por}(\bot, \bot) = \bot \neq \mathbf{V}$.

Example 7.2.2 (The Gustave function). This famous example was proposed by Berry as a function that, although (tri-)stable, is not sequential: it cannot be implemented by a lambda-term (and more generally by any sequential program). It is the monotone function $G : \mathrm{Cl}(B) \times \mathrm{Cl}(B) \times \mathrm{Cl}(B) \to \mathrm{Cl}(B)$ defined by:

$$\begin{aligned} G(\mathbf{V},\mathbf{V},\mathbf{V}) = G(\mathbf{F},\mathbf{F},\mathbf{F}) &= \mathbf{F} \\ G(\mathbf{V},\mathbf{F},x) = G(x,\mathbf{V},\mathbf{F}) = G(\mathbf{F},x,\mathbf{V}) &= \mathbf{V} \text{ for any } x \sqsubset B \\ G(x,y,z) &= \bot \text{ for any other } (x,y,z) \in \mathrm{Cl}(B)^3 \end{aligned}$$

The function G checks whether it has two distinct arguments by testing that two consecutive arguments are $\mathbf{V}$, $\mathbf{F}$ respectively. It is a kind of ternary parallel-or but contrarily to the parallel-or it is stable.

7.2.2 Cartesian product

If X_0 and X_1 are coherent spaces we define $X_0 \,\&\, X_1 = (|X_0 \,\&\, X_1|, \frown_{X_0 \& X_1})$ by:

Notation $\{1\} \times \_ \cup \{2\} \times \_$ in Rel, it would probably be good to make a uniform choice — Olivier

Web: $|X_0 \,\&\, X_1| = \{0\} \times |X_0| \;\cup\; \{1\} \times |X_1|$.

Coherence: $(i,a) \frown_{X_0 \& X_1} (j,b)$ iff $\begin{cases} i \neq j & \text{or} \\ i = j \text{ and } a \frown_{X_i} b \end{cases}$

Any clique $x \sqsubset X_0 \,\&\, X_1$ has the form $x = \{0\} \times \mathsf{pr}_0(x) \cup \{1\} \times \mathsf{pr}_1(x)$ where $\mathsf{pr}_i(x) \sqsubset X_i$ is defined by $\mathsf{pr}_i(x) = \{a \in |X_i|,\ (i,a) \in x\}$. From this we get:

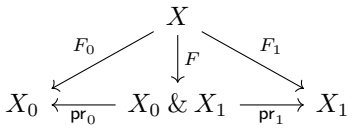
Lemma 7.2.3. *A map $F : \mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1) \to \mathrm{Cl}(X)$ is bi-stable iff the map $F \circ \varphi : X_0 \,\&\, X_1 \to X$ is stable where φ is the bijective function defined by:*

$$\begin{aligned} \varphi : \mathrm{Cl}(X_0 \,\&\, X_1) &\to \mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1) \\ x &\mapsto (\mathsf{pr}_0(x), \mathsf{pr}_1(x)) \end{aligned}$$

In view of this property we will identify $\mathrm{Cl}(X_0 \,\&\, X_1)$ with $\mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1)$ and write (x_0, x_1) the clique $\{0\} \times x_0 \;\cup\; \{1\} \times x_1$.

No surprise if we thus get:

Theorem 7.2.4. *The space $X_0 \,\&\, X_1$ is a cartesian product (see section **??**) in the category of coherent spaces (and stable functions): the maps $\mathsf{pr}_0 : X_0 \,\&\, X_1 \to X_0$ and $\mathsf{pr}_1 : X_0 \,\&\, X_1 \to X_1$ are stable and satisfy that for any stable $F_0 : X \to X_0$ and $F_1 : X \to X_1$ there is a unique stable $F : X \to X_0 \,\&\, X_1$ making the diagram commute:*



The morphism F is often denoted $\langle F_0, F_1 \rangle$. The pairing is given by $F(x) = \{0\} \times F_0(x) \;\cup\; \{1\} \times F_1(x) = (F_0(x), F_1(x))$ according to our notational convention.

7.2.3 The coherent space of stable functions

The fundamental lemma on stable functions is:

Lemma 7.2.5. *Let $F : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ be a monotone function (for inclusion). Then F is stable iff for any clique $x \sqsubset X$ and any point $b \in F(x)$ there is a* finite *subclique $x_0 \subset x$ such that:*

- *$b \in F(x_0)$ and*
- *for any $x' \subset x$, if $b \in F(x')$ then $x_0 \subset x'$.*

The existence of the finite subclique x_0 is consequence of continuity; the least property is consequence of stability. We let the reader check the details.

We denote by $\mathrm{Cl}_{\mathrm{fin}}(X)$ the set of finites cliques of X; when x_0 is a finite clique we write $x_0 \sqsubset_{\mathrm{fin}} X$.

Let $F : X \to Y$ is stable, we define the *trace* of F to be the set

$$\mathsf{Tr}(F) = \{(x_0, b) \in \mathrm{Cl}_{\mathrm{fin}}(X) \times |Y|,\ x_0 \text{ minimal such that } b \in F(x_0)\}$$

To make things clear again, by "minimal such that..." we mean that for any subclique $x' \subset x_0$, if $b \in F(x')$ then $x' = x_0$.

The lemma suggests the following definition of the coherent space $X \Rightarrow Y$ (designed so that $\mathsf{Tr}(F)$ is a clique of $X \Rightarrow Y$):

Web: $|X \Rightarrow Y| = \mathrm{Cl}_{\mathrm{fin}}(X) \times |Y|$.

Coherence: $(x_0, b) \frown_{X \Rightarrow Y} (x_0', b')$ iff $x_0 \cup x_0' \not\sqsubset X$ or $b \frown_Y b'$.

Note that it is more convenient here to first define strict coherence, from which we deduce the coherence relation by $(x_0, b) \subset\!\!\frown_{X \Rightarrow Y} (x_0', b')$ iff $(x_0, b) = (x_0', b')$ or $(x_0, b) \frown_{X \Rightarrow Y} (x_0', b')$.

If $f \sqsubset X \Rightarrow Y$ is a clique of the just defined coherent space we denote by $\mathsf{Fun}(f)$ the function from $\mathrm{Cl}(X)$ to $\mathrm{Cl}(Y)$ defined by:

$$\begin{aligned} \mathsf{Fun}(f) : \mathrm{Cl}(X) &\to \mathrm{Cl}(Y) \\ x &\mapsto \{b \in |Y|,\ \exists x_0 \subset x,\ (x_0, b) \in f\} \end{aligned}$$

Theorem 7.2.6. *If $F : X \to Y$ is stable then $\mathsf{Tr}(F) \sqsubset X \Rightarrow Y$ and we have $\mathsf{Fun}(\mathsf{Tr}(F)) = F$. Conversely if $f \sqsubset X \Rightarrow Y$ then $\mathsf{Fun}(f) : X \to Y$ is stable and we have $\mathsf{Tr}(\mathsf{Fun}(f)) = f$.*

Thus $X \Rightarrow Y$ may be viewed as the coherent space of (traces of) stable functions.

Theorem 7.2.7. *The category of coherent spaces and stable functions is cartesian closed.*

Proof. In view of the definition of cartesian closed categories (see section **??**) we just have to define the evaluation map and the curryfication operation and check equations **??**:

$$\begin{array}{rrcl} \mathsf{Ev}: & (X \Rightarrow Y) \,\&\, X & \to & Y \\ & (f,x) & \mapsto & \mathsf{Fun}(f)(x) \end{array}$$

$$\begin{array}{rrcl} \mathsf{Cur}: \mathsf{Stable}(Z \,\&\, X, Y) & \to & \mathsf{Stable}(Z, X \Rightarrow Y) & \\ F : Z \,\&\, X \to Y & \mapsto & \mathsf{Cur}(F) : Z & \to & X \Rightarrow Y \\ & & z & \mapsto & \mathsf{Tr}(\lambda x F(z,x)) \end{array}$$

in which, all maps (but Cur) being stable we used the notation $F : X \to Y$ for $F : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ and the identification $(z,x) = \{0\} \times z \cup \{1\} \times x$. We leave the verifications of the equations to the reader. □

7.3 The monoidal structure of coherent spaces

7.3.1 Linear functions

A *linear function* between coherent spaces X and Y is a stable function $F : X \to Y$ commuting with *all* compatible unions (as opposed to continuous functions that commute only with directed unions). More precisely $F : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ is linear if:

Linearity: For any family U of cliques in X such that $\bigcup U$ is a clique:

$$F\left(\bigcup U\right) = \bigcup_{u \in U} F(u)$$

Stability: F commutes with compatible intersections.

Remark 7.3.1. The linearity condition entails that F is monotone and continuous, thus that a linear function is stable. The converse is false because the linearity condition, commutation with *any* union, is much stronger than the continuity condition, commutation with directed unions. For example when applied to the empty union linearity entails that the empty clique is sent on the empty clique: if F is linear than $F(\emptyset) = \bigcup_{x \in \emptyset} F(x) = \emptyset$. This argument doesn't apply to continuity because a directed set is nonempty; typically a constant function such as $\lambda x\, \mathbf{V} : B \to B$ is stable but nonlinear because $T(\emptyset) = \mathbf{V}$.

Sending the empty clique on the empty clique is necessary but not sufficient for linearity, we give an example shortly.

Example 7.3.2 (Identity). The identity function $\mathsf{Id}_X : X \to X$ is linear.

Example 7.3.3 (Projections). The projections $\mathsf{pr}_i : X_0 \,\&\, X_1 \to X_i$ are linear.

Example 7.3.4. The evaluation map $\mathsf{Ev} : (X \Rightarrow Y) \,\&\, X \to Y$ is linear in its first argument: for each $x \sqsubset X$ the function $\lambda f\, \mathsf{Ev}(f,x) : (X \Rightarrow Y) \to Y$ is linear.

Example 7.3.5. The Gustave function, viewed as defined on $\mathrm{Cl}(B \,\&\, B \,\&\, B)$ is not linear although it sends the empty clique on the empty clique. Indeed $G(\mathbf{V}, \perp, \perp) = G(\perp, \mathbf{F}, \perp) = \perp$ but $(\mathbf{V}, \perp, \perp) \cup (\perp, \mathbf{F}, \perp) = (\mathbf{V}, \mathbf{F}, \perp)$ and $G(\mathbf{V}, \mathbf{F}, \perp) = \mathbf{V}$ whereas $\perp \cup \perp = \perp$.

The composition of linear functions is clearly a linear function, so that coherent spaces with linear functions form a subcategory of the category of coherent spaces and stable functions. We will denote $\mathsf{Lin}(X, Y)$ the hom set of linear functions from X to Y and write $F : X \to_\ell Y$ for $F \in \mathsf{Lin}(X, Y)$.

7.3.2 Tensor product

We define the coherent space $X_0 \otimes X_1$ by:

Web: $|X_0 \otimes X_1| = |X_0| \times |X_1|$.

Coherence: $(a_0, a_1) \frown_{X_0 \otimes X_1} (a'_0, a'_1)$ iff $a_0 \frown_{X_0} a'_0$ and $a_1 \frown_{X_1} a'_1$.

Proposition 7.3.6. *The tensor product enjoys the following properties, in which all mentionned isomorphisms are linear isomorphisms and the X_is are any coherent spaces:*

Associative: $X_0 \otimes (X_1 \otimes X_2) \simeq (X_0 \otimes X_1) \otimes X_2$.

Symmetric: $X_0 \otimes X_1 \simeq X_1 \otimes X_0$.

Unit: *The space* $\mathbf{1}$ *defined by* $|\mathbf{1}| = \{*\}$ *(singleton set) and* $* \frown_{\mathbf{1}} *$ *is the unit of the tensor:* $X \otimes \mathbf{1} \simeq \mathbf{1} \otimes X \simeq X$.

A stable function $F : X_0 \,\&\, X_1 \to X$ is *bilinear* if it is linear in each of its arguments, that is if for each $x_i \sqsubset X_i$, the maps $F_{x_i} = \lambda x_{\bar{\imath}}\, F(x_0, x_1) : \mathrm{Cl}(X_{\bar{\imath}}) \to \mathrm{Cl}(X)$ is linear.

As in vector spaces, bilinear maps factorize through the tensor space:

Theorem 7.3.7. *Let $\psi : X_0 \,\&\, X_1 \to X_0 \otimes X_1$ be the (bi-)stable function defined by $\psi(x_0, x_1) = x_0 \times x_1$. Then ψ is bilinear.*

Furthermore if $F : X_0 \,\&\, X_1 \to X$ is a bilinear function then there is a unique linear $\hat{F} : X_0 \otimes X_1 \to_\ell X$ such that $F = \hat{F} \circ \psi$. Diagrammatically:

$$\begin{array}{ccc} X_0 \,\&\, X_1 & \xrightarrow{F} & X \\ {\scriptstyle\psi}\downarrow & \nearrow_{\hat{F}} & \\ X_0 \otimes X_1 & & \end{array}$$

Remark 7.3.8 (Extension by linearity). If $F : X \otimes Y \to_\ell Z$ is linear then F is completely determined by its value on rectangle cliques, that is cliques of the form $x \times y$. Indeed any clique $t \sqsubset X \otimes Y$ may be decomposed into a union of rectangles, for example $t = \bigcup_{x \times y \subset t} x \times y$, so that by linearity $F(t) = \bigcup_{x \times y \subset t} F(x \times y)$.

We will often use this fact to define a linear function only on rectangles and say that we extend the definition by linearity.

7.3.3 The coherent space of linear functions

As with stable functions there is a fundamental lemma for linear functions:

Lemma 7.3.9. *Let $F : \mathrm{Cl}(X) \to \mathrm{Cl}(Y)$ be a monotone map. Then F is linear iff for any clique $x \sqsubset X$ and any point $b \in F(x)$, there is an $a \in x$ such that:*

- *$b \in F(\{a\})$ and*
- *for any $x' \subset x$, if $b \in F(x')$ then $a \in x'$.*

Compared to stable functions, linear functions satisfy the stronger property that the finite least subclique $x_0 \subset x$ such that $b \in F(x_0)$ is a singleton $\{a\}$ (in particular it is nonempty).

Proof. If F is linear than write $x = \bigcup_{a \in x} \{a\}$ and apply linearity to get a; note that F being linear $F(\emptyset) = \emptyset$ so that $\{a\}$ is minimal such that $b \in F\{a\}$. By stability we thus have $\{a\} \subset x'$ for any $x' \subset x$ such that $b \in F(x')$.

Conversely assume F satisfy the property. By the fundamental lemma for stable functions 7.2.5 we deduce that F is stable. To get linearity let $(x_i)_{i \in I}$ be a family of cliques such that $x = \bigcup x_i$ is a clique and $b \in F(x)$. By the property there is an $a \in x$ such that $b \in F(\{a\})$. Since $a \in x$ there is an x_i such that $a \in x_i \subset x$ thus $b \in F(x_i)$ and we have proved that $F(x) \subset \bigcup_{i \in I} F(x_i)$; the other inclusion is immediate by monotonicity of F. □

Let $F : X \to_\ell Y$ be a linear function, we define the *linear trace* of F as the set:

$$\mathsf{Tr}_\ell(F) = \{(a, b) \in |X| \times |Y|,\ b \in F(\{a\})$$

Although $\mathsf{Tr}_\ell(F)$ is a subset of $|X \otimes Y|$ it is not in general a clique of $X \otimes Y$ so we have to design a new coherent space for that. The space $X \multimap Y$ is defined by:

Web: $|X \multimap Y| = |X| \times |Y|$.

Coherence: $(a, b) \frown_{X \multimap Y} (a', b')$ iff $(\{a\}, b) \frown_{X \Rightarrow Y} (\{a'\}, b')$ iff $a \smile_X a'$ or $b \frown_Y b'$.

Just as the space for stable functions, the space $X \multimap Y$ is designed so that $\mathsf{Tr}_\ell(F)$ is a clique for any $F : X \to_\ell Y$. Similarly we define a converse of Tr_ℓ: for any clique $f \sqsubset X \multimap Y$ the (to be verified to be) linear function $\mathsf{Fun}_\ell(f)$ is:

$$\begin{aligned} \mathsf{Fun}_\ell(f) : X &\to_\ell Y \\ x &\mapsto \{b \in |Y|,\ \exists a \in x,\ (a, b) \in f\} \end{aligned}$$

Theorem 7.3.10. *If $F : X \to_\ell Y$ is linear then $\mathsf{Tr}_\ell(F) \sqsubset X \multimap Y$ and we have $\mathsf{Fun}_\ell(\mathsf{Tr}_\ell(F)) = F$. Conversely if $f \sqsubset X \multimap Y$ then $\mathsf{Fun}(f) : X \to_\ell Y$ is linear and we have $\mathsf{Tr}_\ell(\mathsf{Fun}_\ell(f)) = f$.*

As a consequence we get:

Theorem 7.3.11. *The category of coherent spaces and linear functions is monoidal symmetric closed.*

Proof. The properties of the tensor (associativity, symmetry, neutral) are immediate and depicted in the following. For the closure, as in the case of stable functions we just have to define the evaluation map and the curryfication operation (see section **??**):

$$\begin{array}{rrcl} \mathsf{Ev}_\ell : & (X \multimap Y) \otimes X & \to_\ell & Y \\ & f \times x & \mapsto & \mathsf{Fun}_\ell(f)(x) \end{array}$$

$$\begin{array}{rrcl} \mathsf{Cur}_\ell : & \mathsf{Lin}(Z \otimes X, Y) & \to & \mathsf{Lin}(Z, X \multimap Y) \\ & F : Z \otimes X \to_\ell Y & \mapsto & \mathsf{Cur}_\ell(F) : Z \to X \multimap Y \\ & & & \quad z \mapsto \mathsf{Tr}_\ell(\lambda x F(z \times x)) \end{array}$$

where Ev_ℓ is defined by extension by linearity (see remark 7.3.8 above). □

7.3.4 Duality

Recall that $X^\perp$ is the dual of X defined by $|X^\perp| = |X|$ and $\frown_{X^\perp} = \asymp_X$. We thus also have $\frown_{X^\perp} = \smile_X$ or equivalently $\smile_{X^\perp} = \frown_X$. Therefore we have:

$$\begin{array}{rl} (a,b) \frown_{X \multimap Y} (a',b') & \text{iff } a \smile_X a' \text{ or } b \frown_Y b' \\ & \text{iff } a \frown_{X^\perp} a' \text{ or } b \smile_{Y^\perp} b' \\ & \text{iff } (b,a) \frown_{Y^\perp \multimap X^\perp} (b',a') \end{array}$$

so that that the coherent spaces $X \multimap Y$ and $Y^\perp \multimap X^\perp$ are naturally isomorphic by the contraposition isomorphism:

$$\begin{array}{c} X \multimap Y \to_\ell Y^\perp \multimap X^\perp \\ f \mapsto f^\perp = \{(b,a) \in |Y| \times |X|,\ (a,b) \in f\} \end{array}$$

Theorem 7.3.12. *The category of coherent spaces and linear function is $*$-autonomous (see 6.1).*

In particular the space **1** is the dualizing object that we will also denote $\perp$ to emphasize this fact. We also have the linear isomorphism:

$$X^\perp \simeq X \multimap \perp$$

Note that the fact that the dualizing object is the same as the unit of the tensor is a peculiarity of coherent spaces.

7.3.5 Additive constructions

Applying duality to the cartesian product & gives rise to a new construction: the $\oplus$ thus defined by:

$$X_0 \oplus X_1 = (X_0^\perp \,\&\, X_1^\perp)^\perp$$

which can be explicited:

Web: $|X_0 \oplus X_1| = \{0\} \times X_0 \ \cup \ \{1\} \times X_1$.

Coherence: $(i,a) \frown_{X_0 \oplus X_1} (j,b)$ iff $\begin{cases} i = j & \text{and} \\ a \frown_{X_i} b \end{cases}$

The cliques of $X_0 \oplus X_1$ are therefore of the form $\{i\} \times x_i$ where $x_i \sqsubset X_i$ so that $\mathrm{Cl}(X_0) \oplus \mathrm{Cl}(X_1)$ may almost be viewed as the disjoint unions of $\mathrm{Cl}(X_0)$ and $\mathrm{Cl}(X_1)$. There is a slight problem with the empty set though, because it is a clique of both X_i but appears only once in $X_0 \oplus X_1$, we come back on this below.

Because $|X_0 \,\&\, X_1| = |X_0 \oplus X_1|$ is the disjoint union of the webs, Girard called them the *additive constructions*.

Theorem 7.3.13. *The space $X_0 \,\&\, X_1$ is a cartesian product in the category of coherent spaces and linear functions: for $i = 0, 1$ the maps $\mathsf{pr}_i : X_0 \,\&\, X_1 \to_\ell X_i$ are linear, an so is the pairing $\langle F_0, F_1 \rangle : X \to_\ell X_0 \& X_1$ when each $F_i : X \to_\ell X_i$ is linear.*

Dually the space $X_0 \oplus X_1$ is a direct sum in the category of coherent spaces and linear functions: the injections $\mathsf{inj}_i : X_i \to_\ell X_0 \oplus X_1$ are defined by $\mathsf{inj}_i(x_i) = \{i\} \times x_i$, and given linear functions $F_i : X_i \to_\ell X$ the copairing is defined by:

$$\begin{aligned} [F_0, F_1] : X_0 \oplus X_1 &\to_\ell X \\ \{i\} \times x_i &\mapsto F_i(x_i) \end{aligned}$$

Remark 7.3.14. The space $X_0 \oplus X_1$ is not a direct sum in the category of *stable* functions because the copairing is not defined on the empty set: if the F_is are nonlinear we may have $F_0(\emptyset) \neq F_1(\emptyset)$ and there is no natural way to define $[F_0, F_1](\emptyset)$ because $\emptyset$ is a clique in either space X_i. The problem doesn't occur when the F_is are linear because then $F_0(\emptyset) = F_1(\emptyset) = \emptyset$ so that we may safely define $[F_0, F_1](\emptyset) = \emptyset$. Girard mention this problem in [**proofntypes**] as the one that led him to the discovery of linear functions.

Remark 7.3.15. The spaces $X_0 \,\&\, X_1$ and $X_0 \oplus X_1$ are different (in general), contrarily to what happens in the category of finite dimension vector spaces or in the category **Rel** of sets and relations in which there is a single biproduct space that is at the same time a cartesian product and a direct sum.

To be complete we should add that both constructions have neutrals: the space $\top = \mathbf{0}$ is the space with empty web and trivial coherence relation. As for the $\mathbf{1}$ and the $\bot$ we use two different names for the same space to emphasize the different roles it plays, as unit of & or of $\oplus$.

Let us end up this section with an enumeration of some linear isomorphisms:

De Morgan: $\mathbf{0}^\bot = \top$, $\top^\bot = \mathbf{0}$, $(X \,\&\, Y)^\bot = X^\bot \oplus Y^\bot$, $(X \oplus Y)^\bot = X^\bot \,\&\, Y^\bot$; all these are actually equalities: same web, same coherence.

Neutrals: $X \,\&\, \top \simeq \top \,\&\, X \simeq X$, $X \oplus \mathbf{0} \simeq \mathbf{0} \oplus X \simeq X$.

Commutativity, associativity: $X \,\&\, Y \simeq Y \,\&\, X$, $X \,\&\, (Y \,\&\, Z) \simeq (X \,\&\, Y) \,\&\, Z$ and similarly with $\oplus$.

7.3.6 Multiplicative constructions

We already have two *multiplicative* spaces, thus named because their web is the cartesian product of the web of their components: $X \otimes Y$ and $X \multimap Y$. The dual of the tensor is the so-called *par* construction denoted $\parr$ defined by:

$$X \parr Y = (X^\perp \otimes Y^\perp)^\perp$$

which give rises to the explicit definition:

Web: $|X \parr Y| = |X| \times |Y|$.

Coherence: $(a,b) \frown_{X \parr Y} (a',b')$ iff $a \frown_X a'$ or $b \frown_Y b'$.

By construction we have:

$$X \multimap Y = X^\perp \parr Y$$

Remark 7.3.16. The $\parr$ construction is a tensor product (in the categorical sense) which has the $\perp$ space as unit. The fact that the $\parr$ is different from the $\otimes$ is a main difference between coherent spaces and finite dimensional vector spaces (or sets and relations): the latter form a compact closed category in which the dual of the tensor is the tensor whereas coherent spaces and linear map form a $*$-autonomous category that is not compact closed.

Here is a collection of linear isomorphisms involving the multiplicative constructions:

De Morgan: $\mathbf{1}^\perp = \perp$, $\perp^\perp = \mathbf{1}$, $(X \otimes Y)^\perp = X^\perp \parr Y^\perp$, $(X \parr Y)^\perp = X^\perp \otimes Y^\perp$, $X \multimap Y = X^\perp \parr Y$, $(X \multimap Y)^\perp = X \otimes Y^\perp$; just as in the additive case all these isomorphisms are actually equalities.

Neutrals: $X \otimes \mathbf{1} \simeq \mathbf{1} \otimes X \simeq X$, $X \parr \perp \simeq \perp \parr X \simeq X$.

Commutativity, associativity: $X \otimes Y \simeq Y \otimes Y$, $X \otimes (Y \otimes Z) = (X \otimes Y) \otimes Z$, and similarly with the $\parr$. These isomorphisms are the final touch for the symmetric monoidal structure of the category of coherent spaces and linear functions.

Distributivity: $X \otimes (Y \oplus Z) \simeq (X \otimes Y) \oplus (X \otimes Z)$, $X \otimes \mathbf{0} \simeq \mathbf{0}$ and the similar ones obtained by duality, expressing distributivity of the $\parr$ on the &.

Remark 7.3.17. The distributivity isomorphisms were another reason Girard invoked for the terminology additive/multiplicative.

7.4 Exponentials

Recall the definition of the space $X \Rightarrow Y$ in section 7.2.3: $(x_0,b) \frown_{X \Rightarrow Y} (x_0',b')$ iff $x_0 \cup x_0' \not\subset X$ or $b \frown_Y b'$ where $(x_0,b),(x_0',b') \in |X \Rightarrow Y| = \mathrm{Cl}_{\mathrm{fin}}(X) \times |Y|$. This suggests the following definition of the *exponential space* $!X$ (read *of course* X or *bang* X):

Web: $|!X| = \mathrm{Cl}_{\mathrm{fin}}(X)$.

Coherence: $x_0 \supset_{!X} x'_0$ iff $x_0 \cup x'_0 \sqsubset X$.

We can then rewrite the definition of $X \Rightarrow Y$ as:

Web: $|X \Rightarrow Y| = |!X| \times |Y|$.

Coherence: $(x_0, b) \frown_{X \Rightarrow Y} (x'_0, b')$ iff $x_0 \smile_{!X} x'_0$ or $b \frown_Y b'$.

which we recognize as the definition of the space $!X \multimap Y$, thus proving the founding isomophism of linear logic:

$$X \Rightarrow Y = !X \multimap Y$$

This equality on internal hom sets can also be depicted as an isomorphism between the sets of stables functions $\mathsf{Stable}(X, Y)$ and the set of linear functions $\mathsf{Lin}(!X, Y)$:

Theorem 7.4.1. *Let $!_X : X \to !X$ be the stable function defined by $!_X(x) = \{x_0 \subset_{\mathrm{fin}} x\}$, the set of finite subcliques of x.*

If $F : X \to Y$ is a stable function we define its linearisation *$F_\ell : !X \to_\ell Y$ by $F_\ell = \mathsf{Fun}_\ell(\mathsf{Tr}(F))$ (so that $\mathsf{Tr}_\ell(F_\ell) = \mathsf{Tr}(F)$); by definition F_ℓ is linear and we have: $F = F_\ell \circ !_X$.*

Conversely if $L : !X \to_\ell Y$ is linear then $L \circ !_X : X \to Y$ is stable and $(L \circ !_X)_\ell = L$.

Remark 7.4.2. The closure property $F_\ell = \mathsf{Fun}_\ell(\mathsf{Tr}_\ell(F_\ell))$ allows us to define functions by giving their trace, a convenience that we just used here and that we will reuse in the sequel.

If $F : X \to_\ell Y$ is a linear function we define $!F : !X \to_\ell !Y$ by:

$$\mathsf{Tr}_\ell(!F) = \bigcup_{n \geq 0} \{(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\}),\ (a_i, b_i) \in \mathsf{Tr}_\ell(F) \text{ for } i = 1, \ldots, n\}$$

(which has to be checked to be a clique in $!X \multimap !Y$).

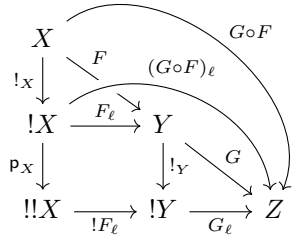
Theorem 7.4.3. *The $!$ operation is functorial on the category of coherent spaces and linear functions. Moreover it is a comonad the counit and comultiplication of which are $\mathsf{d} : ! \to_\ell \mathsf{Id}$ (dereliction) and $\mathsf{p} : ! \to_\ell !!$ (digging) defined by:*

$$\mathsf{Tr}_\ell(\mathsf{d}_X : !X \to_\ell X) = \{(\{a\}, a),\ a \in |X|\}$$
$$\mathsf{Tr}_\ell(\mathsf{p}_X : !X \to_\ell !!X) = \{(u, U),\ u = \bigcup U\}$$

The category of coherent spaces and stable functions is the co-Kleisli category of the linear functions by the exponential comonad.

Proof. For the functoriality and the comonad assertion one has to check a number of equations (see section 6.1.3 and the diagrams in the summary below) that we leave to the reader.

Let $F_\ell : !X \to_\ell Y$ and $G_\ell : !Y \to_\ell Z$ and put $F = F_\ell \circ !_X : X \to Y$, $G = G_\ell \circ !_Y : Y \to Z$; the co-Kleisli composition of F_ℓ and G_ℓ is $G_\ell \circ !F_\ell \circ !\,\mathsf{p}_X : !X \to_\ell Z$. To prove the co-Kleisli assertion we have to check that $(G \circ F)_\ell = G_\ell \circ !F_\ell \circ \mathsf{p}_X$, which in turn is consequence of the fact that $!F_\ell \circ \mathsf{p}_X = !_Y \circ F_\ell$:



□

Remark 7.4.4. Note that d_X is the linearisation of $\mathsf{Id}_X : X \to X$ viewed as a morphism in the category of stable functions, so that we have $\mathsf{Id}_X = \mathsf{d}_X \circ !_X$.

On the other hand $!_X : X \to !X$ is the co-Kleisli counterpart of $\mathsf{Id}_{!X} : !X \to_\ell !X$.

As usual the operation ! has a dual ? (read *why not*) defined by:

$$?X = (!X^\perp)^\perp$$

in which $!X^\perp$ should be read $!(X^\perp)$. Contrarily to the additive and multiplicative case, the web of $?X$ is not the web of $!X$:

Web: $|?X| = \mathrm{Cl}_{\mathrm{fin}}(X^\perp)$ is the set of finite anticliques of X.

Coherence: $x_0 \frown_{?X} x_0'$ iff there is $a \in x_0$ and $a' \in x_0'$ such that $a \frown_X a'$.

We end up this section with a review of the main properties of the exponentials.

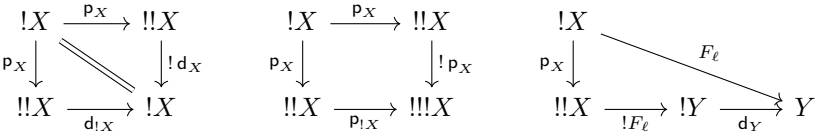
De Morgan: $(!X)^\perp = ?X^\perp$, $(?X)^\perp = !X^\perp$.

Weakening, contraction: There are two natural transformation giving $!X$ the structure of a commutative $\otimes$-comonoid:

$$\mathsf{w}_X : !X \to_\ell 1 \qquad \mathsf{Tr}_\ell(\mathsf{w}_X) = \{(\emptyset, *)\}$$
$$\mathsf{c}_X : !X \to_\ell !X \otimes !X \qquad \mathsf{Tr}_\ell(\mathsf{c}_X) = \{(x, (x_0, x_1)),\ x_0 \cup x_1 = x \sqsubset_{\mathrm{fin}} X\}$$

and the dual transformations obtained by orthogonality giving $?X$ the structure of a commutative $\parr$-monoid.

Comonad: We have the following commuting diagrams expressing the fact that the exponential is a comonad with associated natural transformations d and p:



The last diagram is obtained by the co-Kleisli composition of $F = F_\ell \circ !_X : X \to Y$ and $\mathsf{Id}_Y = \mathsf{d}_Y \circ !_Y : Y \to Y$. Of course dual diagrams exists expressing the fact that ? is a monad.

Exponential isomorphisms: These are the reasons of the terminology *exponential*, the ! and the ? send the additives on the multiplicatives:

$$!(X \,\&\, Y) \simeq !X \otimes !Y$$
$$!\top \simeq \mathbf{1}$$

$$?(X \oplus Y) \simeq ?X \parr ?Y$$
$$?0 \simeq \bot$$

7.5 Conclusion

As said above, coherent spaces were the first model of linear logic, actually Girard designed linear logic after coherent spaces. Among other nice features they form a $*$-autonomous category that is not compact closed, some say that is non degenerated because in compact closed categories the $\otimes$ is equal to its dual. Note however that the multiplicative neutrals $\bot$ and $\mathbf{1}$ (not to speak of the additive ones) are identical, and it is still a question to construct some natural model of linear logic in which $\bot \neq \mathbf{1}$ (related to the problem of finding an explicit construction for the free $*$-autonomous category).

Despite their nice and simple structure they present some peculiarities that makes them singular in the realm of models of linear logic. Here are two important ones:

- The ! comonad is *idempotent* by which we mean that there is a linear function $!X \otimes !X \to_\ell !X$ the linear trace of which is $\{((x_0, x_0), x_0) \sqsubset_{\text{fin}} X\}$ which is a left inverse of c_X. This is unfortunate if one wants to keep the intuition that linear logic is about resource consumption and in particular that the contraction morphism should keep track of the number of times a resource is used.

- The ! comonad is *uniform*, by which we mean that the web $!X$ is made of finite *cliques* as opposed to finite *sets* of points of X. This restricts a function $F : X \to Y$ to expect an argument in $\mathrm{Cl}(X)$ thus made of

coherent points, so to speak the argument has to answer uniformly to any request F has. This might also seem unfortunate if one wants to model non deterministic programs or probabilistic programs in which a same input can give incoherent answers when requested several times.

Indeed other comonads are possible: in coherent spaces one can choose finite *multicliques*, that is multisets of pairwise coherent points, as the web of $!X$; this was proposed by Lafont in the early ages of linear logic and makes the comonad free; it yields a somehow different co-Kleisli category in which functions can differ as soon as they use their arguments a different number of times, a fact that is not naturally expressible with stable functions, and that opens the way to *quantitative semantics*.

Another possible generalization is to remove the coherence restriction, that is taking finite multisets of points not necessarily pairwise coherent for the web of $!X$. This is typically what is done in the sets and relations category and its numerous derived *non uniform* models (see the non uniform coherent model in section 5.3), that allowed among other things the discovery of differential lambda-calculus.

Chapter 8

Scott

8.1 Scott semantics

. The relational model is infinitary[1] in the sense that even a most simple formula like $!(1 \oplus 1) \multimap 1 \oplus 1$ (the type of programs from booleans to booleans) is interpreted as an infinite set (because this is already true of $!(1 \oplus 1)$). This is due to the fact that the interpretation takes into account all repeated uses of the argument of the function, whereas only two values for this parameter are possible. The situation is quite different in Girard's original coherence spaces model because $!(1 \oplus 1)$ is interpreted as a coherence space which has the cliques of $1 \oplus 1$ as web, and there are only 3 such cliques. Therefore, in that model, $!(1 \oplus 1) \multimap 1 \oplus 1$ is interpreted as a coherence space whose web has 6 elements.

A natural attempt to turn **Rel** into a finitary model is therefore to try introduce an exponential $!X$ such that $!X = \mathcal{P}_{\mathrm{fin}}(X)$. This however does not seem possible; at least the most natural attempt, which consists in copying *mutatis mutandis* the definitions of Section 5.2, fails by lack of naturality of $\mathtt{d}$. A way to solve this problem consists in equipping the sets interpreting formulas with a further preorder structure and interpreting proofs as downwards-closed sets. The model we obtain in that way is actually a linear extension of the very first denotational model, discovered by Scott and Strachey **[ScottStrachey]**: the model of complete lattices and Scott-continuous functions.

We present this model now, trying to keep as tight as possible the connection with the relational model of Section 5.2. This connection will be made more explicit in Section 8.2.

A *preorder* is a pair $S = (|S|, \leq_S)$ where $|S|$ (the web of S) is a set (which can be assumed to be at most countable) and $\leq_S$ is a transitive and reflexive relation on $|S|$ (it is important not to assume this relation to be an order relation). An *initial segment* of S is a subset u of $|S|$ such that

$$\forall a \in u \forall a' \in |S| \; a' \leq_S a \Rightarrow a' \in u$$

[1] This is also true of its non-uniform coherent refinements.

that is, which is downwards closed and we use $\mathcal{I}(S)$ for the set of these initial segments. Observe that $(\mathcal{I}(S), \subseteq)$ is a complete lattice with lubs defined as unions and glbs as intersections. Its least element is $\emptyset$ and its largest element is $|S|$.

Given $u \subseteq |S|$ we set $\downarrow_S u = \{a' \in |S| \mid \exists a \in u \; a' \leq_S a\}$ which is the least element of $\mathcal{I}(S)$ which contains u. Observe that $\mathcal{I}(S)$ is prime-algebraic, the prime elements of $\mathcal{I}(S)$ being those of shape $\downarrow_S \{a\}$ where $a \in |S|$. It is crucial to observe that several different preorders S can generate the same $\mathcal{I}(S)$ up to iso.

Given a preorder S, the preorder $S^\perp$ is defined as $S^\perp = (|S|, \geq_S)$ (in other terms it is the opposit of S) so that obviously $S^{\perp\perp} = S$. Given preorders S_1 and S_2, the preorder $S_1 \otimes S_2$ is simply the product preorder: $|S_1 \otimes S_2| = |S_1| \times |S_2|$ and $(a_1, a_2) \leq_{S_1 \otimes S_2} (b_1, b_2)$ if $a_i \leq_{S_i} b_i$ for $i = 1, 2$. Then we set $S \multimap T = (S \otimes T^\perp)^\perp$ so that $|S \multimap T| = |S| \times |T|$ and $(a, b) \leq_{S \multimap T} (a', b')$ is $b' \leq_T b$ and $a \leq_S a'$.

In that way we define a category **ScottL** whose objects are the preorders and where $\mathbf{ScottL}(S, T) = \mathcal{I}(S \multimap T)$. The identity morphism is

$$\begin{aligned} \mathsf{Id}_S &= \{(a, a') \in |S| \times |S| \mid a' \leq a\} \\ &= \underset{S \multimap S}{\downarrow} \{(a, a) \mid a \in |S|\} \end{aligned}$$

and composition is defined as in **Rel** (ordinary composition of relations); it is easy to see that if $s \in \mathbf{ScottL}(S, T)$ and $t \in \mathbf{ScottL}(T, U)$ then indeed $t\, s \in \mathbf{ScottL}(S, U)$.

Let $s \in \mathbf{ScottL}(S, T)$ and $u \in \mathcal{I}(S)$, then we set $s \cdot u = \{b \in |T| \mid \exists a \in u \; (a, b) \in s\} \in \mathcal{I}(T)$. The following lemma is quite easy to prove.

Lemma 8.1.1. *The map* $\mathsf{fun}(s) : \mathcal{I}(S) \to \mathcal{I}(T)$ *defined by* $\mathsf{fun}(s)(u) = s \cdot u$ *commutes with arbitrary unions (we say that it is* linear*). Moreover any linear map* $f : \mathcal{I}(S) \to \mathcal{I}(T)$ *satisfies* $f = \mathsf{fun}(s)$ *for a unique* $s \in \mathbf{ScottL}(S, T)$ *given by* $s = \{(a, b) \in |S| \times |T| \mid b \in f(\downarrow_S \{a\})\}$. *In particular, if* $s, s' \in \mathbf{ScottL}(S, T)$ *satisfy* $\forall u \in \mathcal{I}(S) \; s \cdot u = s' \cdot u$, *then* $s = s'$.

Notice that an isomorphism in **ScottL** from S to T is not necessarily an isomorphism between the preorders. For instance 1 and $(\mathbf{N}, =)$ are isomorphic in **ScottL** (with iso $\{(*, n) \mid n \in \mathbf{N}\}$) but obviously not as preorders. So a bijection $\theta : |S| \to |T|$ such that $\forall a, a' \in |S| \; a \leq_S a' \Leftrightarrow \theta(a) \leq_T \theta(a')$ will be called a *strong isomorphism*. Such a strong isomorphism θ induces an isomorphism $\widehat{\theta} = \{(a, b) \mid b \leq_T \theta(a)\} \in \mathbf{ScottL}(S, T)$ whose inverse (in **ScottL**) is $\widehat{\theta^{-1}}$.

8.1.1 Multiplicative structure

The operation $\otimes$ defined above on preorders (together with the object $1 = (\{*\}, =)$) can be extended to morphisms (same definition as for the tensor of

morphisms in **Rel**), turning **ScottL** into a symmetric monoidal closed category[2], the object of linear morphisms from S to T being $S \multimap T$ equipped with a linear evaluation morphism $\mathsf{ev} \in \mathbf{ScottL}((S \multimap T) \otimes S, T)$ defined as

$$\mathsf{ev} = \{(((a,b),a'),b') \mid b' \leq_T b \text{ and } a \leq_S a'\}\,.$$

The map $\tau : \mathcal{I}(S_1) \times \mathcal{I}(S_2) \to \mathcal{I}(S_1 \otimes S_2)$ defined by $\tau(u_1,u_2) = u_1 \otimes u_2 = u_1 \times u_2$ is bilinear in the sense that, given u_2 it is linear in u_1 and conversely. It has the following universal property.

Lemma 8.1.2. *Given any bilinear $f : \mathcal{I}(S_1) \times \mathcal{I}(S_2) \to \mathcal{I}(T)$, there is exactly one $s \in \mathbf{ScottL}(S_1 \otimes S_2, T)$ such that*

$$\forall u_1 \in \mathcal{I}(S_1), u_2 \in \mathcal{I}(S_2) \quad f(u_1,u_2) = s \cdot (u_1 \otimes u_2)\,,$$

which is given by $s = \{((a_1,a_2),b) \mid b \in f(\downarrow_{S_1}\{a_1\}, \downarrow_{S_2}\{a_2\})\}$. In particular, if $s, s' \in \mathbf{ScottL}(S_1 \otimes S_2, T)$ satisfy $\forall u_1 \in \mathcal{I}(S_1), u_2 \in \mathcal{I}(S_2)$ $s \cdot (u_1 \otimes u_2) = s' \cdot (u_1 \otimes u_2)$, then $s = s'$.

This SMCC is actually *-autonomous with dualizing object $\bot = 1$ and dual of S isomorphic to $S^\bot$ (the opposit of S). Given $s \in \mathbf{ScottL}(S,T)$, $s^\bot \in \mathbf{ScottL}(T^\bot, S^\bot)$ is just the usual relational transpose of s, $s^\bot = \{(b,a) \mid (a,b) \in s\}$.

8.1.2 Additive structure

The category **ScottL** is cartesian with terminal object $\top = (\emptyset,\emptyset)$ and cartesian product of S_1 and S_2 the object $S_1 \,\&\, S_2$ such that $|S_1 \,\&\, S_2| = \{1\} \times |S_1| \cup \{2\} \times |S_2|$ and $(i,a) \leq_{S_1 \& S_2} (i',a')$ if $i = i'$ and $a \leq_{S_i} a'$. The projections are $\mathsf{pr}_i = \{((i,a),a') \mid a' \leq_{S_i} a\}$ for $i = 1,2$. Given $s_i \in \mathbf{ScottL}(T, S_i)$ for $i = 1,2$, the pairing $\langle s_1, s_2\rangle \in \mathbf{ScottL}(T, S_1 \,\&\, S_2)$ is defined as $\{(b,(i,a)) \mid i \in \{1,2\} \text{ and } (b,a) \in s_i\}$ exactly as in **Rel**. By *-autonomy **ScottL** is also co-cartesian, with initial object $0 = \top$ and coproduct $S_1 \oplus S_2 = S_1 \,\&\, S_2$. The associated injections and co-pairing are easily retreived from the projections and the pairing of &.

8.1.3 Exponential structure

Last we come to the exponential which was the main motivation for this model. We take $|!_{\mathsf{s}}S| = \mathcal{M}_{\mathrm{fin}}(|S|)$ with preorder defined by $m \leq_{!_{\mathsf{s}}S} m'$ if $\forall a \in m \exists a' \in m'$ $a \leq_S a'$. Notice that if we had taken $|!_{\mathsf{s}}S| = \mathcal{M}_{\mathrm{fin}}(|S|)$ with the same definition of the preorder relation, we would have obtained a lattice $\mathcal{I}(!_{\mathsf{s}}S)$ isomorphic to that associated with our multiset-based definition, that we prefer in view of Section 8.2. Given $s \in \mathbf{ScottL}(S,T)$, we define $!_{\mathsf{s}}s \subseteq |!_{\mathsf{s}}S \multimap !_{\mathsf{s}}T|$ as

$$!_{\mathsf{s}}s = \{(m,p) \in \mathcal{M}_{\mathrm{fin}}(|S|) \times \mathcal{M}_{\mathrm{fin}}(|T|) \mid \forall b \in p \exists a \in m\ (a,b) \in s\}$$

[2]The associated isomorphisms are strong.

then is is easy to prove that $!_{\mathsf{s}}s \in \mathbf{ScottL}(!_{\mathsf{s}}S, !_{\mathsf{s}}T)$. Given $u \in \mathcal{I}(S)$, let $u^! \in \mathcal{I}(!_{\mathsf{s}}S)$ and let $\pi : \mathcal{I}(S) \to \mathcal{I}(!_{\mathsf{s}}S)$ be defined by $\pi(u) = u^!$. It is easy to prove that π is Scott continuous. Moreover, it enjoys the following universal property.

Lemma 8.1.3. *Given any Scott continuous function*[3] $\mathcal{I}(S) \to \mathcal{I}(T)$*, there is exactly one* $s \in \mathbf{ScottL}(!_{\mathsf{s}}S, T)$ *such that* $\forall u \in \mathcal{I}(S)\ f(u) = s \cdot u^!$*. This morphism* s *is given by* $s = \{([a_1, \ldots, a_n], b) \mid b \in f(\downarrow_S \{a_1, \ldots, a_n\})\}$*. In particular, if* $s, s' \in \mathbf{ScottL}(!_{\mathsf{s}}S, T)$ *satisfy* $\forall u \in \mathcal{I}(S)\ s \cdot u^! = s' \cdot u^!$*, then* $s = s'$*.*

It is easy to check that $!_{\mathsf{s}}s \cdot u^! = (s \cdot u)^!$. A consequence of this equation and of Lemma 8.1.3 is that $!_{\mathsf{s}}\_$ is a functor. Its comonadic structure is given by $\mathsf{d}^{\mathsf{s}}_S = \{(m, a') \mid \exists a \in m\ a' \leq_S a\} \in \mathbf{ScottL}(!_{\mathsf{s}}S, S)$ which satisfies $\forall u \in \mathcal{I}(S)\ \mathsf{d}^{\mathsf{s}}_S \cdot u^! = u$. This equation, together with Lemma 8.1.3, allows to prove easily that d^{s} is natural. The comultiplication of the comonad is $\mathsf{p}^{\mathsf{s}}_S = \{(m, [m_1, \ldots, m_n]) \mid \forall i\ m_i \leq_{!_{\mathsf{s}}S} m\} \in \mathbf{ScottL}(!_{\mathsf{s}}S, !_{\mathsf{s}}!_{\mathsf{s}}S)$ which is easily seen to satisfy $\forall u \in \mathcal{I}(S)\ \mathsf{p}^{\mathsf{s}}_S \cdot u^! = u^{!!}$. Again, the naturality of p^{s} and the three required comonad commutative diagrams easily follow from that equation and from Lemma 8.1.3.

The Seely monoidal structure $(\mathsf{m}^{\mathsf{s},0}, \mathsf{m}^{\mathsf{s},2})$ is defined by $\mathsf{m}^{\mathsf{s},0} = \{(*, [])\} \in \mathbf{ScottL}(1, !_{\mathsf{s}}\top)$ and $\mathsf{m}^{\mathsf{s},2}_{S_1,S_2} = \{((m_1, m_2), 1 \cdot m'_1 + 2 \cdot m'_2) \mid m'_i \leq_{!_{\mathsf{s}}S} m_i \text{ for } i = 1, 2\} = \widehat{\theta} \in \mathbf{ScottL}(!_{\mathsf{s}}S_1 \otimes !_{\mathsf{s}}S_2, !_{\mathsf{s}}(S_1 \,\&\, S_2))$ where $\theta : |!_{\mathsf{s}}S_1 \otimes !_{\mathsf{s}}S_2| \to |!_{\mathsf{s}}(S_1 \,\&\, S_2)|$ is the strong iso defined by $\theta(m_1, m_2) = 1 \cdot m_1 + 2 \cdot m_2$.

The Kleisli category $\mathbf{ScottL}_{!_{\mathsf{s}}}$ has preorders as objects and can be described as follows (thanks to Lemma 8.1.3 and to the equations satsfied by d^{s} and p^{s}): a morphism from S to T is a Scott continuous function $\mathcal{I}(S) \to \mathcal{I}(T)$, composition is the ordinary composition of functions. It is cartesian closed with $\top$ as final object (indeed $\mathcal{I}(\top) = \{\emptyset\}$), $S_1 \,\&\, S_2$ as cartesian product of S_1 and S_2 (and indeed $\mathcal{I}(S_1 \,\&\, S_2) \simeq \mathcal{I}(S_1) \times \mathcal{I}(S_1)$) and $S \Rightarrow T = !_{\mathsf{s}}S \multimap T$ as object of morphisms from S to T (and indeed $\mathcal{I}(S \Rightarrow T)$ is isomorphic to the lattice of Scott continuous functions $\mathcal{I}(S) \to \mathcal{I}(T)$ ordered under the usual pointwise order) and evaluation function defined as usual. So we can identify $\mathbf{ScottL}_{!_{\mathsf{s}}}$ with the usual Scott model of (typed) lambda-calculus, PCF etc.

The main motivation for this construction was to build a "finitary" model on top of **Rel**. Let us explain in what sense this goal has been reached. Let us say that an object S of **ScottL** is finite if $\mathcal{I}(S)$ is a finite set.

Proposition 8.1.4. *The preorders* 1 *and* $\top$ *are finite. If* S *is finite then* $S^\perp$ *and* $!_{\mathsf{s}}S$ *are finite. If* S_1 *and* S_2 *are finite so are* $S_1 \otimes S_2$ *and* $S_1 \,\&\, S_2$*. In particular, if* S *and* T *are finite, so are* $S \multimap T$ *and* $S \Rightarrow T$*.*

Proof. Since $\mathcal{I}(S_1 \,\&\, S_2) \simeq \mathcal{I}(S_1) \times \mathcal{I}(S_2)$, the finiteness of the S_i's implies that of $S_1 \,\&\, S_2$. Next, we know by Lemma 8.1.1 that $\mathcal{I}(S \multimap T)$ is isomorphic to the space of linear functions $\mathcal{I}(S) \to \mathcal{I}(T)$, hence if S and T are finite so is $S \multimap T$. Taking $T = \bot$, we see that the finiteness of S implies that of $S^\perp$. Since $S_1 \otimes S_2 = (S_1 \multimap S_2^\perp)^\perp$, it follows that the finiteness of S_1 and S_2 implies that of $S_1 \otimes S_2$. Next, by Lemma 8.1.3, the finiteness of S and T implies that of

[3] Remember that this means that f is monotonic and commutes with directed unions.

$S \Rightarrow T$. Therefore, since $!_{\mathsf{s}}S$ is (strongly) isomorphic to $(S \Rightarrow \bot)^{\bot}$ it follows that the finiteness of S implies that of $!_{\mathsf{s}}S$. □

Notice that the original coherence space model of Girard [**Girard**] has a quite similar finiteness property (this is also true of our hypercoherence space model [**Ehrhard**]). One main feature of the Scott model is that it combines this finiteness with a strong form of *may non-determinism* which is simply implemented by the operation of lattices interpreting types (and is not available in coherence and hypercoherence spaces whose main purpose is precisely to reject non-determinism). This might be quite a useful feature especially for using denotational models in program verification.

8.2 Relation with the relational model

One main difference between the LL models $(\mathbf{Rel}, !\_)$ and $(\mathbf{ScottL}, !_{\mathsf{s}}\_)$ is that the Kleisli category of the latter is well-pointed (by Lemma 8.1.3) whereas the Kleisli category of the former is not. We proved in [**Ehrhard**] that the latter is the "extensional collapse" of the former. In the hierarchy of simple types based *e.g.* on a standard interpretation of integers, such a result can easily be proved using a syntactic trick[4] which however does not provide informations on the structure of this collapse and is not easily extendable to the whosle LL.

8.2.1 A duality on preorders

In contrast, a careful LL-based analysis of the collapse led to a surprising new duality which is the central concept of [**Ehrhard**]: let S be a preorder and let $u, u' \subseteq |S|$, let us write $u \perp[S]\ u'$ if

$$(\underset{S}{\downarrow} u) \cap u' \neq \emptyset \Rightarrow u \cap u' \neq \emptyset$$

that is, $u \perp[S]\ u'$ means that u' is not able to separate u from its downwards closure in S. This definition is symmetric in the following sense (the proof of these equivalences is quite easy)::

$$u \perp[S]\ u' \Leftrightarrow ((\underset{S}{\downarrow} u) \cap (\underset{S^{\bot}}{\downarrow} u') \neq \emptyset \Rightarrow u \cap u' \neq \emptyset) \Leftrightarrow u' \perp[S^{\bot}]\ u\,.$$

Given $D \subseteq \mathcal{P}(|S|)$, we define

$$D^{\bot[S]} = \{u' \subseteq |S| \mid \forall u \in D \quad u \perp[S]\ u'\}\,.$$

Observe that the following usual properties hold:

- $D \subseteq D' \Rightarrow D'^{\bot[S]} \subseteq D^{\bot[S]}$

[4] Very roughly: in a may non-deterministic extension of PCF, all compact element in the Scott hierarchy are definable, and this language can be interpreted in the relational hierarchy. Then a standard "logical relation lemma" allows to prove the announced property.

- $D \subseteq D^{\perp[S]\perp[S^\perp]}$

from which it follows that $D^{\perp[S]} = D^{\perp[S]\perp[S^\perp]\perp[S]}$.

The following result is a simple illustration on how this notion of "orthogonality" is used for proving properties of sets equal to their biorthogonal.

Lemma 8.2.1. *If $D = D^{\perp[S]\perp[S^\perp]}$ (or, equivalently, if $D = D'^{\perp[S^\perp]}$ for some $D' \subseteq \mathcal{P}(|\underline{E}|)$) then*

- *$\mathcal{I}(S) \subseteq D$ (in particular $\emptyset, |S| \in D$)*
- *D is closed under arbitrary unions.*

Proof. If $u \in \mathcal{I}(S)$ then $u \perp[S]\ u'$ holds for all $u' \subseteq |S|$, whence the first property. Towards the second one, let $\mathcal{U}$ be a subset of D. We prove that $\cup\mathcal{U} \in D = D'^{\perp[S^\perp]}$. Let $u' \in D'$, we have to prove that $u' \perp[S^\perp] \cup\mathcal{U}$, that is $\cup\mathcal{U} \perp[S]\ u'$. So assume that $\downarrow_S (\cup\mathcal{U}) \cap u' \neq \emptyset$. Since $\downarrow_S (\cup\mathcal{U}) = \cup_{u\in\mathcal{U}} \downarrow_S u$, there exists $u \in \mathcal{U}$ such that $\downarrow_S u \cap u' \neq \emptyset$, and we have $u \cap u' \neq \emptyset$ because $u \in D$, it follows that $\cup\mathcal{U} \cap u' \neq \emptyset$ as contended. Notice that it is not necessarily true that $\cap\mathcal{U} \in D$. □

8.2.2 The category of preorders with projections

We build a model of LL based on these central notions. Let us call *preorder with projection* (PP for short) any pair $E = (\underline{E}, \mathsf{D}(E))$ where $\underline{E}$ is a preorder (the *carrier of* E) and $\mathsf{D}(E)$ (the *extensionality of* E) is a subset of $\mathcal{P}(|\underline{E}|)$ such that

$$\mathsf{D}(E) = \mathsf{D}(E)^{\perp[\underline{E}]\perp[\underline{E}^\perp]}.$$

Then we can define a relation $\pi_E \subseteq \mathcal{P}(|\underline{E}|) \times \mathcal{I}(\underline{E})$ as follows (using letters $u, v \ldots$ for arbitrary subsets of $|\underline{E}|$ and letters $r, s \ldots$ for initial segments of $\underline{E}$):

$$u\ \pi_E\ r \quad \text{if} \quad u \in \mathsf{D}(E) \text{ and } \underset{\underline{E}}{\downarrow} u = r$$

This relation is a partial function $\mathcal{P}(|\underline{E}|) \to \mathcal{I}(\underline{E})$ and as such, it defines a partial equivalence relation ε_E on $\mathcal{P}(|\underline{E}|)$ given explicitely by

$$u\ \varepsilon_E\ v \quad \text{if} \quad u, v \in \mathsf{D}(E) \text{ and } \underset{\underline{E}}{\downarrow} u = \underset{\underline{E}}{\downarrow} v\,.$$

The intuition behind such an object E is as follows: $\underline{E}$ is the interpretation of a formula A of LL in **ScottL** and $|\underline{E}|$ is the interpretation *of the same formula*[5] A in **Rel**. Then $u\ \varepsilon_E\ v$ means that u and v are extensionally equivalent, and $u\ \pi_E\ r$ means that r is the "extensionalization" of u. The definitions below implement these intuitions.

[5] This identification is the main reason for which we use multisets and not sets in the definition of $!_{\mathsf{s}}S$.

We define $E^\perp = (\underline{E}^\perp, \mathsf{D}(E)^{\perp[\underline{E}]})$ so that by definition $E^{\perp\perp} = E$. Next, given PP's E_i for $i = 1, 2$, we define $E_1 \otimes E_2$ by $\underline{E_1 \otimes E_2} = \underline{E_1} \otimes \underline{E_2}$ and $\mathsf{D}(E_1 \otimes E_2) = \{u_1 \otimes u_2 \mid u_i \in \mathsf{D}(E_i) \text{ for } i = 1, 2\}^{\perp[\underline{E_1} \otimes \underline{E_2}]}$ (remember that we use $u_1 \otimes u_2 = u_1 \times u_2$). Next, given PP's E and F, we define $E \multimap F = (E \multimap F^\perp)^\perp$.

Lemma 8.2.2. *Let $w \in \mathcal{P}(|\underline{E \multimap F}|)$, the following properties are equivalent:*

1. $w \in \mathsf{D}(E \multimap F)$
2. *for all $u \in \mathsf{D}(E)$ one has $w \cdot u \in \mathsf{D}(F)$ and $w \cdot \downarrow_{\underline{E}} u \subseteq \downarrow_{\underline{F}} (w \cdot u)$*
3. *for all $u \in \mathsf{D}(E)$ one has $w \cdot u \in \mathsf{D}(F)$ and $\downarrow_{\underline{E} \multimap \underline{F}} w \cdot \downarrow_{\underline{E}} u \subseteq \downarrow_{\underline{F}} (w \cdot u)$*
4. *for all $u \in \mathsf{D}(E)$ one has $w \cdot u \in \mathsf{D}(F)$ and $\downarrow_{\underline{E} \multimap \underline{F}} w \cdot \downarrow_{\underline{E}} u = \downarrow_{\underline{F}} (w \cdot u)$*
5. *there exists $t \in \mathcal{I}(\underline{E \multimap F})$ such that for all $u \in \mathcal{P}(|\underline{E}|)$ and $r \in \downarrow \underline{E}$, if $u \, \pi_E \, r$ then $w \cdot u \, \pi_F \, t \cdot r$.*

Proof. The implication (3)⇒(4) is due to the fact that $\downarrow_{\underline{E} \multimap \underline{F}} w \cdot \downarrow_{\underline{E}} u \supseteq \downarrow_{\underline{F}} (w \cdot u)$ always holds. The implication (2)⇒(3) is due to the fact that $\downarrow_{\underline{E} \multimap \underline{F}} w \cdot \downarrow_{\underline{E}} u = \downarrow_{\underline{F}} (w \cdot \downarrow_{\underline{E}} u)$. The equivalence (4)⇔(5) is a direct application of the definitions of π_E, π_F and $\pi_{E \multimap F}$. Of course when (4) holds the t whose existence is stipulated by (5) is $\downarrow_{\underline{E} \multimap \underline{F}} w$.

So assume (1) and let us prove (2). Let $u \in \mathsf{D}(E)$. We prove first that $w \cdot u \in \mathsf{D}(F) = \mathsf{D}(F)^{\perp[\underline{F}]\perp[\underline{F}^\perp]}$ so let $v' \in \mathsf{D}(F)^{\perp[\underline{F}]}$ and let us prove that $w \cdot u \perp[\underline{F}] v'$. So assume that $\downarrow_{\underline{F}} (w \cdot u) \cap v' \neq \emptyset$, that is $w \cdot u \cap \downarrow_{\underline{F}^\perp} v' \neq \emptyset$. This is equivalent to $w \cap (u \times \downarrow_{\underline{F}^\perp} v')$ and therefore implies $w \cap \downarrow_{\underline{E} \otimes \underline{F}^\perp} (u \otimes v') \neq \emptyset$. Our assumption on w implies $w \cap (u \otimes v') \neq \emptyset$ and this finally implies $w \cdot u \cap v' \neq \emptyset$. Therefore $w \cdot u \perp[\underline{F}] v'$ as contended. Next we must prove that $w \cdot \downarrow_{\underline{E}} u \subseteq \downarrow_{\underline{F}} (w \cdot u)$ so let $b \in w \cdot \downarrow_{\underline{E}} u$. We have $\downarrow_{\underline{F}^\perp} \{b\} \in \mathsf{D}(F)^{\perp[\underline{F}]}$ and $w \cdot \downarrow_{\underline{E}} u \cap \downarrow_{\underline{F}^\perp} \{b\} \neq \emptyset$. Hence, by the same reasoning as above (using our assumption on w), $w \cdot u \cap \downarrow_{\underline{F}^\perp} \{b\} \neq \emptyset$, that is $b \in \downarrow_{\underline{F}} (w \cdot u)$.

Now assume (2) and let us prove (1) so assume that w satisfies this latter condition. Let $u \in \mathsf{D}(E)$ and $v' \in \mathsf{D}(F)^{\perp[\underline{F}^\perp]}$ and assume that $\downarrow_{\underline{E} \multimap \underline{F}} w \cap (u \otimes v') \neq \emptyset$, that is $w \cap (\downarrow_{\underline{E}} u \times \downarrow_{\underline{F}^\perp} v') \neq \emptyset$. This implies $w \cdot \downarrow_{\underline{E}} u \cap \downarrow_{\underline{F}^\perp} v' \neq \emptyset$. By our assumption (2) (second part), this implies $\downarrow_{\underline{F}} (w \cdot u) \cap \downarrow_{\underline{F}^\perp} v' \neq \emptyset$, that is $\downarrow_{\underline{F}} (w \cdot u) \cap v' \neq \emptyset$ and hence by our assumption (2) again (first part), we get $(w \cdot u) \cap v' \neq \emptyset$ which implies $w \cap (u \otimes v') \neq \emptyset$. □

8.2.2.1 Multiplicative structure

From this lemma, it results that one can define a category **PoProj** whose objects are the PP's and $\mathbf{PoProj}(E, F) = \mathsf{D}(E \multimap F)$ which has the diagonal relation $\subseteq |\underline{E}| \times |\underline{E}|$ as identity $E \to E$ and composition defined as in **Rel**.

Similarly one proves the following.

Lemma 8.2.3. *Let E_1, E_2 and F be PP's and let $w \subseteq |\underline{E_1 \otimes E_2 \multimap F}|$. The following properties are equivalent*

- $w \in \mathsf{D}(E_1 \otimes E_2 \multimap F)$;
- *for all* $u_1 \in \mathsf{D}(E_1), u_2 \in \mathsf{D}(E_2)$, *one has* $w \cdot (u_1 \otimes u_2) \in \mathsf{D}(F)$ *and* $w \cdot (\downarrow_{\underline{E_1}} u_1 \otimes \downarrow_{\underline{E_2}} u_2) \subseteq \downarrow_{\underline{F}} (w \cdot (u_1 \otimes u_2))$.

Using this lemma, we can establish associativity of the tensor product (or more precisely that the category **PoProj** is monoidal when equipped with this tensor product, and structural isomorphisms defined as in **Rel**). This requires two auxiliary properties.

Lemma 8.2.4. *Let E and F be PP's and $\theta : |\underline{E}| \to |\underline{F}|$ be a strong isomorphism. If $\forall u \in \mathsf{D}(E)\ \theta \cdot u \in \mathsf{D}(F)$ then $\theta \in \mathbf{PoProj}(E, F)$.*

This is an immediate consequence of Lemma 8.2.2 and of the fact that θ is a strong isomorphism.

Lemma 8.2.5. *Let E, F and G be PP's, then the bijection $\alpha : |\underline{G} \otimes \underline{E} \multimap \underline{F}|$ belongs to $\mathbf{PoProj}((G \otimes E \multimap F) \to (G \multimap (E \multimap F)))$ and $\alpha^{-1} \in \mathbf{PoProj}((G \multimap (E \multimap F)) \to (G \otimes E \multimap F))$.*

Proof. The idea is to apply (several times) Lemma 8.2.2. Let $t \in \mathsf{D}(G \otimes E \multimap F)$, we prove that $\alpha \cdot t \in \mathsf{D}(G \multimap (E \multimap F))$. Let $w \in \mathsf{D}(G)$ and let us prove that $(\alpha \cdot t) \cdot w \in \mathsf{D}(E \multimap F)$. So let $u \in \mathsf{D}(E)$, we prove that $((\alpha \cdot t) \cdot w) \cdot u \in \mathsf{D}(F)$ which results from our assumption on w and from the fact that $((\alpha \cdot t) \cdot w) \cdot u = t \cdot (w \otimes u)$. Then we must prove that $((\alpha \cdot t) \cdot w) \cdot \downarrow_{\underline{E}} u \subseteq \downarrow_{\underline{F}} (((\alpha \cdot t) \cdot w) \cdot u)$, which results from

$$\begin{aligned} t \cdot (w \otimes \underset{\underline{E}}{\downarrow} u) &\subseteq t \cdot (\underset{\underline{G}}{\downarrow} w \otimes \underset{\underline{E}}{\downarrow} u) \\ &= t \cdot \underset{\underline{G} \otimes \underline{E}}{\downarrow} (w \otimes u) \\ &\subseteq \underset{\underline{F}}{\downarrow} (t \cdot (w \otimes u)) \,. \end{aligned}$$

This ends the proof that $(\alpha \cdot t) \cdot w \in \mathsf{D}(E \multimap F)$. We must prove next that $(\alpha \cdot t) \cdot \downarrow_{\underline{G}} w \subseteq \downarrow_{\underline{E} \multimap \underline{F}} ((\alpha \cdot t) \cdot w)$. Let $a \in |\underline{E}|$ and let $u = \downarrow_{\underline{E}} \{a\}$, remember that $u \in \mathsf{D}(E)$. It is sufficient to prove that

$$((\alpha \cdot t) \cdot \underset{\underline{G}}{\downarrow} w) \cdot u \subseteq (\underset{\underline{E} \multimap \underline{F}}{\downarrow} ((\alpha \cdot t) \cdot w)) \cdot u \,. \tag{8.1}$$

Indeed, assume (8.1) and assume $(a, b) \in (\alpha \cdot t) \cdot \downarrow_{\underline{G}} w$ for some $b \in |\underline{F}|$, then we have $b \in ((\alpha \cdot t) \cdot \downarrow_{\underline{G}} w) \cdot u$ and hence $b \in s \cdot \downarrow_{\underline{E}} \{a\}$ where $s = \downarrow_{\underline{E} \multimap \underline{F}} ((\alpha \cdot t) \cdot w)$ which by Lemma 8.1.1 implies $(a, b) \in s$, proving our contention. So we

prove (8.1):

$$\begin{aligned}((\alpha \cdot t) \cdot \underset{\underline{G}}{\downarrow} w) \cdot u &= t \cdot (\underset{\underline{G}}{\downarrow} w \otimes u) \\ &= t \cdot \underset{\underline{G} \otimes \underline{E}}{\downarrow} (w \otimes u) \\ &\subseteq \underset{\underline{F}}{\downarrow} (t \cdot (w \otimes u)) \quad \text{by our assumption about } t \\ &= \underset{\underline{F}}{\downarrow} (((\alpha \cdot t) \cdot w) \cdot u) \\ &= \underset{\underline{F}}{\downarrow} (\underset{\underline{E} \multimap \underline{F}}{\downarrow} ((\alpha \cdot t) \cdot w) \cdot u) \\ &\subseteq \underset{\underline{E} \multimap \underline{F}}{\downarrow} ((\alpha \cdot t) \cdot w) \cdot u\end{aligned}$$

because this latter set is downwards closed in $\underline{F}$. This ends the proof that $\alpha \in \mathbf{PoProj}((G \otimes E \multimap F) \to (G \multimap (E \multimap F)))$ and it remains to prove that $\alpha^{-1} \in \mathbf{PoProj}((G \multimap (E \multimap F)) \to (G \otimes E \multimap F))$; the proof is similar (and simpler). □

Then, given PP's E_1, E_2 and E_3, we have just seen that α is a strong iso $((E_1 \otimes E_2) \otimes E_3)^{\perp} \to (E_1 \otimes (E_2 \otimes E_3))^{\perp}$, and hence it is a strong iso $(E_1 \otimes E_2) \otimes E_3 \to E_1 \otimes (E_2 \otimes E_3)$ thus establishing the monoidal structure of **PoProj**. The fact that σ is a strong iso $E_1 \otimes E_2 \to E_2 \otimes E_1$ is an immediate consequence of Lemma 8.2.4.

8.2.2.2 Additive structure

Chapter 9

Quantitative

Chapter 10

Polarisation

Chapter 11

Exponentials

Part III

Dynamic models

Chapter 12

Geometry of Interaction

12.1 A brief and partial history of the geometry of interaction

Originally the **geometry of interaction** was a research program proposed by Girard [24] aiming at a modelisation of cut-elimination by some mathematical device, as opposed to the syntactical approach introduced by Gentzen in natural deduction or sequent calculus. Girard soon came with a proposition [15] in which cut-elimination was represented by the so-called **execution formula**:

$$\mathsf{Ex}(\sigma,\pi) = (1-\sigma^2)\pi \sum_{n\geq 0} (\sigma\pi)^n (1-\sigma^2)$$

where π and σ are operators representing respectively a linear logic proof in the MELL fragment and its cut rules. The $(1-\sigma^2)$ part is a projector on the subspace associated to the conclusions of the proof, so that the formula is computing the interaction between the proof and its cuts, projecting the result on its conclusions. The main theorem of this initial version of the GoI established the strong convergence of the sum when applied to interpretations of typed proofs, which may be viewed as the GoI counterpart of the strong normalisation theorem for system F.

This initial interpretation has been revisited and reformulated in many different ways by various authors. In the first place Danos and Regnier showed that the GoI could be viewed more combinatorialy as computing an invariant of cut elimination: **persistent paths** so named because they are the paths that are consistently reduced along the cut elimination [11]; this lead to an interpretation of proof nets as some kind of automaton, called the **IAM** for **Interaction Abstract Machine** in which a token enters the proof by one of its conclusion and is routed and acted upon by transitions associated to each logical rules eventually reaching an other conclusion [12]. The trajectory followed by the token is a path called an **execution path** (or a **regular path**) and it is shown that the set of execution paths is the same as the set of persistent paths.

One can recover Girard's interpretation by viewing the **IAM** automaton as an operator acting on the Hilbert space generated by tokens; the strong convergence of the execution formula (in the typed case) is reformulated into the claim that there are a finite number of persistent/execution paths.

The interpretation has been further extended to the untyped case, following a suggestion of Girard [16], Malacaria and Regnier showed that the execution formula when applied to pure lambda-terms was still converging in a weak sense [37]; this weak convergence also has a combinatorial counterpart in terms of paths, namely that any persistent cycle has to be opened by the reduction.

The geometry of interaction was shown to be strongly related to **sharing reduction** by Gonthier-Abadi-Lévy [26] who designed an interpretation of lambda-terms into so called **sharing graphs** that merged together ideas coming from the GoI and from Lamping's implementation of beta-reduction [33] as a first concrete realisation of Lévy's optimal reduction [35, 36]. The correctness of the sharing reduction w.r.t. optimal reduction was shown by using the invariance of **consistent paths**, a reformulation of execution paths in the sharing graph formalism.

The work on optimal reduction was also carried by Asperti and Laneve who showed that Lévy's labels that were used to define families of redexes in (the retracts of) a lambda-term, could be viewed as particular paths in the graphical form of the lambda-term satisfying a geometrical condition that they called **legality** [3]. Of course legal paths were immediately recognised as another form of execution paths [4].

It was also quickly realised that the execution formula could be understood as expressing the interaction between strategies in game semantics. In particular in the Abramsky-Jagadeesan-Malacaria game model of PCF [2] the history free strategy interpreting a proof/term can be viewed as an operator acting on the space/game interpreting the type; this operator happens to be the geometry of interaction interpretation of the proof [10].

Last but not least Girard's GoI interpretation was further abstracted and shown to be a particular instance of an interpretation of proofs in traced monoidal categories [32], today refered as a **GoI situation** [1, 28]: the trace is the correct categorical way to describe the execution formula, *i.e.*, the travel of a token along an execution path, *i.e.*, the composition of strategies.

In subsequent work the geometry of interaction for MELL was further extended by Girard to additive connectives of linear logic [18], and finally completely reformulated using a new approach based on Von Neuman algebras [19]. This last version has also a combinatorial counterpart that was explicited by Seiller [43].

In this book we will however stick to the MELL fragment of linear logic, as this is the fragment that permits encoding of full lambda-calculus.

12.1.1 Straight paths

We will consider a particular class of paths that is suitable for out study. Informally **straight paths** are paths that may change direction only in axiom

and cut nodes, and that never bounce back. However, due to the combinatorial complexity of path reduction, we will have to add some technical conditions that can be ignored at first read.

We will first assume that, up to eta-expansion, proof nets don't contain exponential axioms, that is axioms with conclusions $?A^\perp$ and $!A$.

A *straight path* in a proof net $\mathcal{R}$ is a path $\gamma = (\mathsf{n}_0, (e_i)_{1\leq i\leq N})$ such that for any $1 \leq i < N$:

- if $e_i = a^-$ and $e_{i+1} = a'^+$ then a and a' are the two *distinct* conclusions of an *ax*-node.
- if $e_i = a^+$ and $e_{i+1} = a'^-$ then a and a' are the two *distinct* premises of a *cut*-node.

Note that straight paths don't form a category as the composition of two straight paths may not be straight.

We will further ask a technical but light condition on straight paths, namely that they neither start downardly, nor end upwardly in the middle of an exponential tree:

- if n_0 is a ?-node and $e_1 = a^+$ (where a is the conclusion arrow of n_0) then n_0 is a *d*-node;
- symmetrically if n_N is a ?-node and $e_N = a^-$ (where a is the conclusion arrow of n_N) then n_N is a *d*-node.

This condition is the reason why we don't want exponential axioms in proof nets, with exponential axioms a path could start downardly or end upwardly in an exponential axiom, which could lead by cut elimination to the middle of an exponential branch. It is light because a straight path can always be extended (possibly in multiple ways) so as to satisfy it. We will see in the proof of the special cut lemma what it is useful to.

It is immediate that any residual of a straight path by any reduction is straight but we can be a bit more precise (this can be skipped at first read).

A straight path may uniquely be written in the form $\gamma = \overline{\gamma_s}\gamma_a\gamma_t$ where γ_s and γ_t are (possibly empty) down-maximal descent subpaths, thus cross no cut. We call γ_a the **active part** of γ, γ_s and γ_t the *source* and *target passive parts* of γ.

If γ has a residual γ' by a one step non axiom reduction or by an axiom reduction such that the non cut conclusion of the axiom (the a_0 arrow in the axiom cut case p. 115) is not the source of γ_s or γ_t then it is readily seen that $\gamma' = \overline{\gamma'_s}\gamma'_a\gamma'_t$ where γ'_a, the active part of γ', is residual of γ_a and γ'_s and γ'_t are residuals of γ_s and γ_t.

If γ has a residual γ' by an axiom reduction such that the non cut conclusion of the axiom is source of γ_s then, assuming the notations of the axiom cut 115, $\gamma_a = a^+_{\mathsf{c}}a^-_1\overline{\delta_s}\delta_a$ where δ_s is the maximal descent subpath of γ targeted on the source node of a_1 so that $\gamma = \overline{\gamma_s}a^+_{\mathsf{c}}a^-_1\overline{\delta_s}\delta_a\gamma_t$. In this case $\gamma' = \overline{\gamma'_s}\ \overline{\delta'_s}\delta'_a\gamma'_t =$

$\overline{\delta'_s\gamma'_s}\delta'_a\gamma'_t$ where δ'_a, the active part of γ', is a residual of δ_a, and γ'_s, δ'_s and γ'_t are residuals of γ_s, δ_s and γ_t. In particular $\overline{\delta'_s}\delta'_a$ is a residual of γ_a but δ'_s is no longer active, so to speak it has switched from the active part of γ to the source passive part of γ'.

The case where the non cut conclusion of the axiom is source of γ_t is symmetric. We thus have:

Lemma 12.1.1. *In any residual γ' of a path γ, the active part of γ' is a subpath (that may be a proper subpath) of a residual of the active part of γ.*

As a conséquence if Γ is a set of straight paths and Γ' is the set of active parts of elements of Γ then any reduction of Γ is a reduction of Γ' and conversely.

12.1.2 Persistent paths

From now on all paths considered will be assumed to be straight. A (straight) path γ in $\mathcal{R}$ is *persistent* if for any sequence of *non-weakening* reductions ρ of $\mathcal{R}$, it has at least one residual in the ρ-retract of $\mathcal{R}$.

A path that crosses only weakening cuts is **normal**. If γ is normal then it has some residual by any non-weakening reduction step and all its residuals are normal. Thus a normal path is persistent.

Theorem 12.1.2 (Strong normalisation of path reduction)**.** *If γ is a finite straight path, then all sequences of reductions of γ are finite, that is lead to a (possibly empty) set of normal residuals.*

We postpone the proof to the next section but immediately state an important corollary:

Corollary 12.1.3. *A path γ is persistent iff it admits a non-weakening reduction leading to a normal residual.*

Proof. If we have a reduction yielding a normal residual of γ, since a normal path have normal residuals by any reduction, by confluence we see that γ has at least one residual by any non weakening reduction, thus γ is persistent.

Conversely if γ is persistent then let us reduce it (by non weakening cuts), the reduction being finite we must stop at some point and since γ is persistent it has some residual γ' at this point. But γ' is normal, otherwise it would cross some non weakening cut contradicting the fact that the reduction of γ is finished. □

Remark 12.1.4 (Why straight paths?). Assuming the notations of the multiplicative reduction p. 116 let γ be the bouncing (thus non straight) path $\gamma = a_0^+ a_1^-$. Then γ has no residual in the retract, thus γ is non persistent. Similar remark apply to $\gamma = a_0^+ a_1^-$ in the contraction reduction p. 118. So a non straight path bouncing on a node that is immediately premise of a cut is clealy non persistent. This is not enough to conclude that any bouncing path is not persistent, as the bouncing node could never be premise of a cut, but it is a hint that a bouncing path is morally not persistent. In other terms bouncing paths are not good

candidates for a reduction invariant. This is one main reason why we don't consider them and restrict to straight paths.

12.1.3 The special cut lemma

Given a straight path γ in a proof net $\mathcal{R}$, a cut c is **special for** γ if:

- c is an exponential cut between a non weakening ?-node and a !-node associated with a box $'\_$ (so named for consistency with the notations used in cut elimination steps p. 116);
- γ crosses c;
- the active part of γ contains no (premise or conclusion of an) auxiliary door of $'\_$. Equivalently γ doesn't cross any cut located below an auxiliary door of $'\_$, that is γ has no subpath of the form $\gamma_0 a_{c'}^{-}$ where γ_0 is a descent path starting from an auxiliary door of $'\_$ and ending on a cut c' and $a_{c'}$ is the premise of c' that don't belong to γ_0.

Therefore γ is bound to enter and leave the box $'\_$ only by its principal door, except in two possible cases:

1. the first time γ enters $'\_$ may be by an auxiliary door but in this case γ can be decomposed into $\gamma = \overline{\gamma_0}\gamma_1\gamma_2$ where γ_0 is a descent path from an auxiliary door p' of $'\_$ and γ_1 is a path contained in $'\_$, sourced on p' and targeted on the principal door of $'\_$;
2. the last time γ exits $'\_$ may be by an auxiliary door, which is symmetric to the preceding case.

Also note that if γ first enters $'\_$ for the first time by crossing the cut c (or symetrically leaves $'\_$ for the last time by crossing the cut c) our light condition on straight paths insures that it has visited (or that it will visit) the full exponential branch before (after) reaching c.

Lemma 12.1.5 (Special cut lemma)**.** *Let $\mathcal{R}$ be a proof net containing only exponential cuts and γ a straight nonnormal path. Then there is a cut c in $\mathcal{R}$ that is special for γ.*

Furthermore γ has at most one residual by the one step reduction of c, 0 if γ exchanges the premises of c, 1 otherwise. If $\gamma' \in \mathsf{Res}_{\mathsf{c}}(\gamma)$ is the residual of γ then the length of its active part is strictly less than the length of the active part of γ.

Proof (sketchy). The existence is proved by induction on $\mathcal{R}$. If $\mathcal{R}$ is obtained from $\mathcal{R}_0$ by adding a $\parr$, c, d or ! node (associated to a box) then the result comes by induction on $\mathcal{R}_0$ and the fact that special cut for γ in $\mathcal{R}_0$ is still special for γ in $\mathcal{R}$. If $\mathcal{R} = \mathcal{R}_0 \otimes \mathcal{R}_1$ then as γ is straight it lies entirery in $\mathcal{R}_0$ or $\mathcal{R}_1$ and the result comes again by induction hypothesis. Similarly if $\mathcal{R} = \mathcal{R}_0 \,\mathsf{c}\, \mathcal{R}_1$ where c is a cut that is not crossed by γ.

So we are left with the case where $\mathcal{R} = \mathcal{R}_0\, \mathsf{c}_1\, \mathcal{R}_1$ where c_1 is an exponential cut crossed by γ. Let us call b_1 the box (the principal door of which is) premise of c_1. If c_1 is not special for γ then there is an exponential cut c_2 lying below an auxiliary door of b_1 and crossed by γ. Let us call b_2 the box premise of c_2; we note that b_2 cannot contain b_1 otherwise it would also contain c_1 contradicting the fact that $\mathcal{R} = \mathcal{R}_0\, \mathsf{c}_1\, \mathcal{R}_1$ (which entails in particular that c_1 is in no box at all). From which we deduce that b_1 and b_2 are disjoint and that c_2 is distinct from c_1. If c_2 is not special for γ then we iterate the process and get a sequence of cuts c_1, c_2, ... and a sequence of boxes b_1, b_2, ... such that (the principal door of) b_k is premise of c_k, b_j and b_k are disjoint, c_j and c_k are distinct for $j < k$, c_k is crossed by γ and c_{k+1} is a cut below some auxiliary door of b_k. This sequence must be finite since $\mathcal{R}$ is finite thus ends on a c_n which is special for γ.

For the at most one assertion the only case of interest is when c is a c-cut, because no other type of cut may duplicate a path. Only the subpaths of γ that are contained in $\underline{'}$ can be duplicated but by the special cut assumption all these subpaths are sourced and/or targeted on the principal door of $\underline{'}$. Also, thanks to our light condition on straight paths, all the subpaths of γ crossing c also visit an entire exponential branch premise of c, in particular they visit one or the other of the premise of the contraction so that they may have only one residual.

Finally the length decreasing assumption is consequence of the fact that, in any type of cut, the ? premise of the cut c has been removed in the residual of γ (in the case of the dereliction cut, also the ! premise has been removed). Note that somme new nodes and edges may have been added on the auxiliary doors of (the residuals of) $\underline{'}$ thus increasing the length of γ, which is the precise reason why we consider only the active part of γ: thanks to the special cut assumption only the passive parts of γ may cross the auxiliary doors of $\underline{'}$ so that the active part is not affected by the additional edges and nodes. □

The special cut lemma will be used in various context. Typically it shows the strong normalisation of path reduction.

Corollary 12.1.6 (Strong normalization of path reduction)**.** *Any reduction of a straight path γ terminates yielding a set of normal residuals (possibly empty if γ is not persistent).*

Proof. By the special cut lemma one deduces that given a straight path γ there is a reduction sequence of γ that at each step chooses either a multiplicative or axiom cut crossed by γ if there is one, either a special cut for γ, the lemma insuring the existence of such a cut in this case. Such a reduction will be called a **special reduction** of the path γ. Any step during a special reduction of γ strictly decreases the active part of γ thus any special reduction of γ must terminate.

We therefore get a non weakening reduction that terminates. The confluence for non weakening reductions entails by a standard argument that there cannot be an infinite non weakening reduction of γ. □

12.2 An algebraic caracterisation of persistent paths

In this section we will present a purely syntactical device, the **dynamic algebra** presented by generators and relations, and show that, viewed as a simple rewriting system, it can be used to caracterize persistent paths in a proof net. In the next section we will give some models of the dynamic algebra, interpreting the elements as transitions on some set of states, and show how this builds an interpretation of proof nets as some kind of (reversible) automaton, or equivalently as some matrices (operators) acting of the state space, thus giving an account to Girard's *execution formula*. This will also help us to see how the execution formula is related to composition of strategies in game semantics, and more generally to trace in traced monoidal categories.

12.2.1 The dynamic algebra Λ^*

The dynamic algebra Λ^* is the first order equational theory given below. We will use the letters u, v, w for the closed terms of Λ^*, x and y for the constants (also called the *coefficients*).

First order signature: a set of constant and function symbols:

- The signature of a (noncommutative) monoid with zero: a binary composition symbol $.$, a constant 1 and a constant 0.
- a unary function symbol $*$ that will be denoted in exponent: $*(u) = u^*$.
- A unary function symbol !.
- Six constant symbols: the *multiplicative coefficients* p and q, the *exponential coefficients* d, r, s and t.

Equations: when dealing with closed Λ^*-terms, in order to keep notations light and unless explicitely mentionned otherwise, we will write $u = v$ for $\Lambda^* \vdash u = v$.

But associativity which has a special reading all the axiom equations of Λ^* are oriented from left to right so as to be easily seen as a rewriting system. The first set of equations is the *structural set*:

Monoid equations: composition is associative, formally $.(.(u, v), w) = .(u, .(v, w))$.
For this reason we will write $.(u, v) = uv$; we come back on this below.

The constant 1 is neutral for composition: $u1 = 1u = u$.

The constant 0 is absorbant for composition: $u0 = 0u = 0$.

Involutive antimorphism: $(u^*)^* = u$, $(uv)^* = v^*u^*$, $1^* = 1$, $0^* = 0$.

Box morphism: $!(u)!(v) = !(uv)$, $!(1) = 1$, $!(0) = 0$, $!(u)^* = !(u^*)$.

When two closed terms are provably equal using only the structural set of equations we will say that they are *structurally equal*. Structural equality is readily seen to be decidable because all the above equations when oriented from left to right form a confluent terminating rewriting system.

Composition gives Λ^* the structure of a monoid which we acknowledged by using the notation uv for $.(u, v)$. More generally we will consider terms of Λ^* up to monoid equations (neutral and associativity) which will be emphasized by calling them *words* and by identifying composition with word concatenation. Thus, up to the involution and box morphism, a Λ^* word is a finite list of coefficients, dual of coefficients and ! closed terms.

The second set of equations is the *dynamic set*:

Multiplicative anihilations: $p^*p = q^*q = 1$, $p^*q = q^*p = 0$.

Exponential anihilations: $r^*r = s^*s = 1$, $r^*s = s^*r = 0$.

Derelection commutations: $!(u)d = du$, $d^*!(u) = ud^*$.

Contraction commutations: $!(u)x = x!(u)$, $x^*!(u) = !(u)x^*$ for $x = r, s$.

Auxiliary commutations: $!(u)t = t!^2(u)$, $t^*!(u) = !^2(u)t^*$ where $!^2(u)$ stands for $!(!(u))$.

Remark 12.2.1 (Λ^* as a rewriting system on words). The dual forms of the anihilation or commutation equations are consequence of the others thanks to the involution equations. For example $q^*p = q^*(p^*)^* = (p^*q)^* = 0^* = 0$ and $d^*!(u) = d^*(!(u)^*)^* = (!(u)^*d)^* = (!(u^*)d)^* = (du^*)^* = (u^*)^*d^* = ud^*$. However as exemplified here we have to use equations from right to left to get the dual ones. As we want to read the equations as rewriting rules when necessary, we shall keep the whole set presented.

Viewed as a rewriting system on words, Λ^* is easily seen to be terminating and confluent, because coefficient only interacts on their left while dual coefficients only interact on their right.

12.2.2 An easy model

Just as to convince oneself that the Λ^* equational theory is nontrivial (it doesn't prove $0 = 1$) we shall give a first model of it that we will call the **N model of** Λ^*. A *partial permutation* on $\mathbf{N}$ is a one-to-one map σ from a subset of $\mathbf{N}$, the *domain* of σ denoted $\operatorname{dom}\sigma$, onto a subset of $\mathbf{N}$, the *codomain* of σ denoted $\operatorname{codom}\sigma$. Partial permutations compose in the natural way, namely $\sigma\tau$ is the partial permutation defined on $\operatorname{dom}\sigma\tau = \{n \in \operatorname{dom}\tau, \tau(n) \in \operatorname{dom}\sigma\}$.

Composition is associative, has as neutral $\mathsf{Id}_{\mathbf{N}}$, the identity on $\mathbf{N}$ (which is a partial permutation with full domain and codomain) that we will denote $\mathbf{1}$, and as absorbant element being the partial permutation with empty domain (and codomain) that we will denote $\mathbf{0}$.

As σ is one-to-one we may define its inverse $\sigma^* : \operatorname{codom}\sigma \to \operatorname{dom}\sigma$ by $\sigma^*\sigma = \mathsf{Id}_{\operatorname{dom}\sigma}$ and $\sigma\sigma^* = \mathsf{Id}_{\operatorname{codom}\sigma}$. We then have $(\sigma\tau)^* = \tau^*\sigma^*$, $\mathbf{1}^* = \mathbf{1}$, and $\mathbf{0}^* = \mathbf{0}$.

The set of partial permutations on $\mathbf{N}$ therefore validates the involutive monoid structure of Λ^* equations[1].

For the box morphism we need an additional structure: we fix a one-to-one map from $\mathbf{N}^2$ onto $\mathbf{N}$ denoted $(n_1, n_2) \to \langle n_1, n_2 \rangle$, for example $\langle n_1, n_2 \rangle = \frac{1}{2}(n_1 + n_2)(n_1 + n_2 + 1) + n_1$. If $n_1, \ldots, n_{k+1}$ are natural numbers we denote $\langle n_1, \ldots, n_{k+1} \rangle = \langle n_1, \langle \ldots, \langle n_k, n_{k+1} \rangle \ldots \rangle \rangle$. By the one-to-one onto assumption, for any k, any integer n may be uniquely written $n = \langle n_1, \ldots, n_{k+1} \rangle$.

Given a partial permutation σ we define $!(\sigma)$ by:

- $\mathrm{dom}\, !(\sigma) = \{n \in \mathbf{N}, n = \langle n_1, n_2 \rangle, n_2 \in \mathrm{dom}\, \sigma\}$;
- for $n = \langle n_1, n_2 \rangle \in \mathrm{dom}\, !(\sigma)$, $!(\sigma)(n) = \langle n_1, \sigma(n_2) \rangle$.

One easily checks that ! is a morphism w.r.t. the monoid structure of partial inversion, respecting $\mathbf{0}$ and the inversion, thus satisfying the box morphism equations which ends the modelization of the structural set of equations.

For the dynamic set we define the partial permutations p, q, d, r, s and t on $\mathbf{N}$ by:

- $p(n) = 2n$, $q(n) = 2n + 1$ (actually any two permutations with disjoint codomains would do, for example $p(n) = \langle n, 0 \rangle$ and $q(n) = \langle n, 1 \rangle$).
- $d(n) = \langle 0, n \rangle$ (actually any integer in place of 0 would do as well).
- $r(n) = \langle \rho(n_1), n_2 \rangle$, $s(n) = \langle \sigma(n_1), n_2 \rangle$ where $n = \langle n_1, n_2 \rangle$ and ρ and σ are any two permutations with full domain and disjoint codomain (for example one can take $\rho = p$ and $\sigma = q$).
- $t(n) = \langle \tau(n_1, n_2), n_3 \rangle$ where $n = \langle n_1, n_2, n_3 \rangle$ and τ is any permutation from $\mathbf{N}^2$ to $\mathbf{N}$ (not necessarily onto), for example one can take $\tau(n_1, n_2) = \langle n_1, n_2 \rangle$.

It is a routine but useful exercice to check that these satisfy anihilation and commutation equations. Note in particular that equations of the form $x^*x = 1$ are satisfied because all coefficients are interpreted by permutations with full domains, whereas equations of the form $x^*y = 0$ are due to the fact that x and y have disjoint codomains.

When w is a closed Λ^*-term we will denote as $w_{\mathbf{N}}$ its interpretation as a partial permutation on $\mathbf{N}$.

12.2.3 ab^* forms

Now that we know that Λ^* is non trivial we may address the question of recognising nonzero words.

A word in Λ^* is *positive* (resp. *negative*) if it is structurally equal to a word of the form $!^{k_1}(x_1) \ldots !^{k_n}(x_n)$ (resp. $!^{k_1}(x_1^*) \ldots !^{k_n}(x_n^*)$) where the x_i's are

[1] As an aside note, partial permutations (on any set) form a well known structure called an *inverse monoid* which is a slight generalisation of the group structure, see [40].

coefficients. If a is a positive word then a^* is negative (and conversely) and a^*a is provably equal to 1. An ab^***-form** is a word structurally equal to ab^* for some positive words a and b.

An ab^* form is almost normal for Λ^* viewed as a rewriting system, almost because it may contain residual rewritings such as in $!(p)d$ which is a positive word thus an ab^* form but still can be rewritten in dp. However the important fact is that an ab^* form cannot be proven equal to 0 in Λ^* since $a^*(ab^*)b = (a^*a)(b^*b) = 1$ and $0 = 1$ is false in the **N** model.

Note that ab^* forms are not the only non (provably) 0 terms, for example r^*t is not equal to any ab^* form but is non null in the **N** model if we choose appropriately the interpretation of coefficients, for example if we set $\tau(n_1, n_2) = \langle n_1, n_2\rangle$ so that t has full codomain. Actually we could add equations consistently with Λ^* canceling all terms non provably equal to some ab^* form, that is with still a non trivial model of the whole set of equations. For example we could choose $\rho(n_1) = \langle 0, n_1\rangle$, $\sigma(n_1) = \langle 1, n_1\rangle$ and $\tau(n_1, n_2) = \langle 2, n_1, n_2\rangle$ so that r, s and t have disjoint codomains thus satisfy $r^*t = s^*t = 0$. We shall not do so as we will see shortly that non ab^* forms don't appear in the scope of our study.

12.2.4 Weight of paths

Let $\mathcal{R}$ be a proof net. To each arrow a in $\mathcal{R}$ we associate a coefficient x_a in Λ^* depending on the target node of a:

Multiplicative: $x_a = p$ (resp. q) if a is left (resp. right) premise of a multiplicative node ($\parr$ or $\otimes$).

Dereliction: $x_a = d$ is a is premise of a dereliction node.

Contraction: $x_a = r$ (resp. s) if a is left (resp. right) premise of a contraction node.

Auxiliary door: $x_a = t$ if a is premise of an auxiliary door node of a box.

Other: $x_a = 1$ in all other cases of arrow a.

Recall that the depth $\mathrm{d}(\mathsf{n})$ of a node n in $\mathcal{R}$ is the number of boxes containing n (doors of a box are considered outside nodes of the box) and that the depth $\mathrm{d}(a)$ of an arrow a is the depth of its target node. We define the functor **weight** from $\mathcal{R}^*$ to Λ^* by:

- if a is an arrow in $\mathcal{R}$ then $\boldsymbol{w}(a) = !^{\mathrm{d}(a)}(x_a)$;
- if $e = a^+$ is a forward edge then $\boldsymbol{w}(e) = \boldsymbol{w}(a)$; if $e = a^-$ is a backward edge then $\boldsymbol{w}(e) = \boldsymbol{w}(a)^*$;
- $\boldsymbol{w}(\epsilon_{\mathsf{n}}) = 1$ for any node n; if γ is a path and e a composable edge $\boldsymbol{w}(\gamma e) = \boldsymbol{w}(e)\boldsymbol{w}(\gamma)$.

Remark 12.2.2. If γ and δ are two composable paths we have $\boldsymbol{w}(\gamma\delta) = \boldsymbol{w}(\delta)\boldsymbol{w}(\gamma)$ which seems to implie that the functor $\boldsymbol{w}(\_)$ is contravariant. This is due to the fact we choose to denote path composition by concatenation; if we used categorical composition we would have $\boldsymbol{w}(\delta \circ \gamma) = \boldsymbol{w}(\delta)\boldsymbol{w}(\gamma)$ making explicit the fact that $\boldsymbol{w}(\_)$ is covariant indeed.

As a simple but significant remark we note that the weight of a normal path is in ab^* form, a first step towards the algebraic caracterization of persistent paths.

12.2.5 Regular paths

A straight path γ is **regular** if $\boldsymbol{w}(\gamma) = 0$ is not a consequence of the equational theory Λ^*, equivalently if $\boldsymbol{w}(\gamma) \neq 0$ in some non trivial model of Λ^*. Thanks to the remark at the end of the preceding section a normal path is therefore regular, which generalizes in:

Theorem 12.2.3 (ab^* theorem)**.** *Let γ be a straight path in a proof net $\mathcal{R}$; then, using the equations of Λ^* oriented from left to right, $\boldsymbol{w}(\gamma)$ may be rewritten either into 0, in which case γ is not persistent, or into an ab^* form, in which case γ is persistent.*

As a consequence γ is persistent iff γ is regular.

Remark 12.2.4 (Why straight paths (part 2)?). Let $\gamma = a_0^+ a_1^-$ in the multiplicative (p. 116) or contraction (p. 118) reduction case; then $\boldsymbol{w}(\gamma) = p^*q$ (in the multiplicative case) or r^*s (in the contraction case) $= 0$, thus γ is not regular. Any such bouncing path is not regular, but as already remarked it could be persistent if no cut ever reaches its bouncing node. This makes a second reason for limiting our study to straight paths.

Proof. The fact that $\boldsymbol{w}(\gamma)$ *rewrites* into 0 or an ab^* form is, by confluence and termination of the rewriting system, consequence of the fact that $\boldsymbol{w}(\gamma)$ is provably equal to 0 or an ab^* form. We prove this by induction on the length of the special reduction of γ.

If γ crosses some axiom cut then it is immediate that $\boldsymbol{w}(\gamma)$ is preserved by the one step reduction since all the edges involved have weight 1.

If γ crosses a multiplicative cut then we have two cases: either one of the crossing exchanges the premises of the cut in which case γ is not persistent. Using the notations of the multiplicative cut case (p. 116), γ has a subpath of the form $a_i^+ a_{\mathsf{c}}^+ {a'_{\mathsf{c}}}^- {a'_j}^-$ or ${a'_i}^+ {a'_{\mathsf{c}}}^+ a_{\mathsf{c}}^- a_j^-$ with $i \neq j$. Such a subpath has weight p^*q if $i = 1$ and $j = 0$, q^*p if $i = 0$ and $j = 1$. Thus $\boldsymbol{w}(\gamma) = 0$.

Otherwise each crossing of the multiplicative cut respects the premises, that is, is of the form $a_i^+ a_{\mathsf{c}}^+ {a'_{\mathsf{c}}}^- {a'_i}^-$ or ${a'_i}^+ {a'_{\mathsf{c}}}^+ a_{\mathsf{c}}^- a_i^-$ where $i = 0$ or 1. Such subpaths have weight $p^*p = 1$ or $q^*q = 1$.

If we fire the multiplicative cut, γ has one residual γ' and each crossing subpath becomes $a_i^+ {a'_i}^-$ or ${a'_i}^+ a_i^-$ which have weight 1 because a_i and a'_i are

premise of cut nodes in the retract. Since nothing else changed between γ and γ' we deduce that $\boldsymbol{w}(\gamma) = \boldsymbol{w}(\gamma')$, thus get the result by induction[2].

If there are no more multiplicative cuts crossed by γ the special reduction chooses a special cut c which is therefore an exponential cut. We use the notations of the exponential reduction steps p. 116.

We begin by assuming that γ starts and ends outside the box $\overline{\ }$ and never crosses an auxiliary door of $\overline{\ }$. Therefore γ may be decomposed into $\gamma = \gamma_0\alpha_1\delta_1\beta_1^*\gamma_1\ldots\gamma_{k-1}\alpha_k\delta_k\beta_k^*\gamma_k$ where the γ_i's are subpaths outside $\overline{\ }$, the δ_i's are subpaths entirely contained in $\overline{\ }$, the α_i's and the β_i's are the c-crossing subpaths of the form $a^+_{\epsilon_i}a^+_{\mathsf{c}}{a'_{\mathsf{c}}}^-{a'_0}^-$. Note that all edges in α_i and β_i have weight 1 except a_{ϵ_i}, the premise of the exponential node, the weight of which is an exponential coefficient x_i. Also, as the subpath δ_i is lying entirely inside $\overline{\ }$ we have $\boldsymbol{w}(\delta_i) = !(u_i)$ where u_i is the weight of δ_i into the subnet contained in $\overline{\ }$.

If γ exchanges the premises of c then γ is not persistent. This can happen only in the case c is a contraction cut and there is an i such that $\alpha_i = a^+_{\epsilon_i}a^+_{\mathsf{c}}{a'_{\mathsf{c}}}^-{a'_0}^-$ while $\beta_i = a^+_{1-\epsilon_i}a^+_{\mathsf{c}}{a'_{\mathsf{c}}}^-{a'_0}^-$. We deduce that $\boldsymbol{w}(\alpha_i\delta_i\beta_i^*) = y_i^*!(u_i)x_i$ where x_i and y_i are respectively r and s or s and r. Thus $\boldsymbol{w}(\alpha_i\delta_i\beta_i^*) = 0$ by the commutation and anihilation equations for 0 and s, so that $\boldsymbol{w}(\gamma) = 0$.

If γ never exchanges the premises of c, then for each i we have $\alpha_i = \beta_i$ so that $\boldsymbol{w}(\alpha_i\delta_i\beta_i^*) = x_i^*!(u_i)x_i = x_i^*x_i!^c(u_i) = !^c(u_i)$ where x_i and c are respectively: d and 0 if c is a dereliction cut, r or s and 1 if c is a contraction cut, t and 2 if c is a commutative cut. Thus $\boldsymbol{w}(\gamma) = v_kx_k^*!(u_k)x_kv_{k-1}\ldots v_1x_1^*!(u_1)x_1v_0$ (where the v_i's are the weights of the γ_i subpaths) $= v_k!^c(u_k)v_{k-1}\ldots v_1!^c(u_1)v_0$.

The residual of α_i by the reduction of c is $\alpha'_i = a^+_{\epsilon_i}{a'_0}^-$ in which the arrow a_{ϵ_i} is now premise of a cut node, so that the weight of α'_i is 1. Thus the c-crossing subpath $\alpha_i\delta_i\alpha_i^*$ have residual $\alpha'_i\delta_i{\alpha'_i}^*$, with weight $!^c(u_i)$ because in the retract δ_i is now lying into c boxes (c being defined as before, depending on the nature of the cut c). Thus the residual γ' of γ has weight $\boldsymbol{w}(\gamma') = v_k!^c(u_k)v_{k-1}\ldots v_1!^c(u_1)v_0$, that is $\boldsymbol{w}(\gamma) = \boldsymbol{w}(\gamma')$.

If now γ starts inside $\overline{\ }$ then $\gamma = \delta_0\beta_0^*\gamma_0\alpha_1\delta_1\beta_1^*\gamma_1\ldots\gamma_{k-1}\alpha_k\delta_k\beta_k^*\gamma_k$ where δ_0 is lying inside $\overline{\ }$ targeted on the principal door. If γ exchanges the premises of c the computation is just the same as before and we conclude that γ is not persistent and that $\boldsymbol{w}(\gamma) = 0$. So we may suppose $\alpha_i = \beta_i$ for $i = 1,\ldots,k$ so that $\boldsymbol{w}(\gamma) = v_kx_k^*!(u_k)x_kv_{k-1}\ldots v_1x_1^*!(u_1)x_1v_0x_0^*!(u_0)$ from which we deduce that $\boldsymbol{w}(\gamma) = v_k!^c(u_k)v_{k-1}\ldots v_1!^c(u_1)v_0!^c(u_0)x_0^*$.

On the other hand $\gamma' = \delta_0{\beta'_0}^*\gamma_0\alpha'_1\delta_1{\alpha'_1}^*\ldots\alpha'_k\delta_k{\alpha'_k}^*\gamma_k$ and since each δ_i lies into c boxes in the retract, $\boldsymbol{w}(\gamma')$ is $\boldsymbol{w}(\gamma') = v_k!^c(u_k)v_{k-1}\ldots v_1!^c(u_1)v_0!^c(u_0)$. Thus $\boldsymbol{w}(\gamma) = \boldsymbol{w}(\gamma')x_0^*$.

Last γ could enter $\overline{\ }$ by one auxiliary door. As c is special this can happen only the very first time γ enters $\overline{\ }$, that is γ starts below an auxiliary door of

[2]There is a tricky case when γ admits a crossing of the cut that doesn't visit both premises of the multiplicative nodes. This may happen only at the beginning or end of γ, for example if $\gamma = a^+_{\mathsf{c}}{a'_{\mathsf{c}}}^-{a'_0}^-\delta$. In this case the weight of the crossing subpath is not preserved since it is p^* before the reduction and 1 after. This is not a problem since $\boldsymbol{w}(\gamma) = \boldsymbol{w}(\delta)p^*$ so that the result comes by induction on δ.

$\underline{'}$, from which we deduce that γ has the form $\gamma = \sigma_0^*\delta_0\beta_0^*\gamma_0\alpha_1\delta_1\beta_1^* \dots \alpha_k\delta_k\beta_k^*\gamma_k$ where σ_0 is a descent path starting from inside $\underline{'}$ and crossing an auxiliary door of $\underline{'}$, the α_i's, β_i's, γ_i's and δ_i's are as before. If γ exchanges the premises of c we have $\boldsymbol{w}(\gamma) = 0$ as before. Otherwise $\alpha_i = \beta_i$ for $i \neq 0$ so that $\boldsymbol{w}(\gamma) = v_k x_k^*!(u_k)x_k \dots x_1^*!(u_1)x_1v_0x_0^*!(u_0)b_0^*$ where $b_0 = \boldsymbol{w}(\sigma_0)$ is a positive word since σ_0 is a descent path. Therefore $\boldsymbol{w}(\gamma) = v_k!^c(u_k) \dots !^c(u_1)v_0!^c(u_0)x_0^*b_0^*$.

The residual of γ is $\gamma' = {\sigma'_0}^*\delta_0{\beta'_0}^*\gamma_0{\alpha'_1}\delta_1{\alpha'_1}^* \dots {\alpha'_k}\delta_k{\alpha'_k}^*\gamma_k$ where σ'_0 being the residual of σ_0 is still a descent path. Therefore we can compute $\boldsymbol{w}(\gamma') = v_k!^c(u_k) \dots !^c(u_1)v_0!^c(u_0){b'_0}^*$ where b'_0, the weight of σ'_0, is a positive word.

In summary wherever γ starts, if it doesn't exchange the premises of c then $\boldsymbol{w}(\gamma)$ has the shape wb^* for some positive word b (possibly equal to 1), while $\boldsymbol{w}(\gamma')$ has the shape wb'^* for some positive word b' (possibly equal to 1). The situation is symmetric at the end of γ so that we finally get: either γ exchanges the premises of c in which case γ is not persistent and $\boldsymbol{w}(\gamma)$ rewrites to 0, or $\boldsymbol{w}(\gamma) = awb$ and $\boldsymbol{w}(\gamma') = a'wb'^*$ for some positive words (possibly equal to 1) a, a', b and b' and some word w.

In this second case if γ is non persistent it is also the case of γ' so by induction $\boldsymbol{w}(\gamma') = 0$ which entails $w = 0$ thus $\boldsymbol{w}(\gamma) = 0$. If γ, thus γ', is persistent then by induction $\boldsymbol{w}(\gamma')$ is equal to an ab^* form which entails that w has an ab^* form by confluence of the rewriting sytem and finally that $\boldsymbol{w}(\gamma)$ has an ab^* form. □

Remark 12.2.5 (The non preservation of weight). It is impossible to reach equality of weights of γ and γ' in all cases. This is one reason why we had to restrict to special reduction which allows a strict control on the way γ may visit the box.

To get preservation of weight we should have that $b_0x_0 = b'_0$ in the case γ starts below an auxiliary door of $\underline{'}$. If we consider each case of exponential cut, this leads to new equations that we could be tempted to add to our system: $td = 1$, $tr = rt$, $ts = st$ and $t^2 = t!(t)$. Unfortunately (exercise for the reader) the first one, together with the Λ^* equations, allows to prove $0 = 1$, making the attempt unworthy.

Another solution to this problem would be to slightly change the syntax of MELL, firstly making the promotion rule functorial: from $\vdash \Gamma, A$ deduce $\vdash ?\Gamma, !A$ and secondly adding a digging rule to recover the regular promotion: from $\vdash \Gamma, ??A$ deduce $\vdash \Gamma, ?A$. This has the defect to contradict the subformula property in cut free proofs, but the advantage that one can redefine exponential cut elimination steps in a much more regular way: intuitively, after taking the appropriate action on the box (removing it for dereliction, erasing all its content for weakening, duplicating it for contraction, pushing it inside for the commutative cut), the ? node cut on the principal door is simply moved on the auxiliary doors. In particular with such a reduction the weight of paths would be invariant by reduction.

Corollary 12.2.6 (Extensionability of regular paths)**.** *Let γ be a regular path sourced on a node n which is neither a conclusion node, nor a w-node. Then*

there is an edge e such that $e\gamma$ is regular. Symmetrically if γ doesn't end on a conclusion node nor a w-node then there is an edge e such that γe is regular.

Proof. Since γ is regular its weight rewrites into $\boldsymbol{w}(\gamma) = ab^*$ for some positive words a and b.

If γ begins upwardly from a node n, note that n cannot be an ax-node, thus has a unique conclusion arrow a the weight of which has the form $!^d(x)$ where d is the depth of the target node of a and x is the coefficient associated to a. Therefore the path $a^-\gamma$ is straight and has weight $\boldsymbol{w}(a^-\gamma) = ab^*!^d(x^*)$ which is in ab^* form. Thus $a^-\gamma$ is regular.

If γ begins downardly from a node n then n being a non w-node has at least one premise a_0 of weight $!^d(x_0)$. Thus $a_0^+\gamma$ is straight and has weight $\boldsymbol{w}(a_0^+\gamma) = ab^*!^d(x_0)$. The words a and b being positive we have $a = !^{d_1}(y_1)\dots!^d_k(y_k)$ and $b = !^{e_1}(z_1)\dots!^{e_l}(z_l)$ where the d_i's and e_j's are integers and the y_i's and z_j's are coefficients. So $\boldsymbol{w}(a_0^+\gamma) = !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{e_1}(z_1^*)!^d(x_0)$.

By the ab^*-theorem this being the weight of a straight path has to rewrite into an ab^* form and in view of the rewriting rules this means the $!^d(x_0)$ will move to its left using only commutation rules until reaching one of two possibilities:

- $!^d(x_0)$ has traversed the whole b^* part, that is

  $$\boldsymbol{w}(a_0^+\gamma) = !^{d_1}(y_1)\dots!^d_k(y_k)!^{d'}(x_0)!^{e'_l}(z_l^*)\dots!^{e'_1}(z_1^*)$$

  which is an ab^* form, thus $a_0^+\gamma$ is regular.

- $!^d(x_0)$ has stopped inside the b^* part at a point where no commutation rule can be applied. Thus

  $$\boldsymbol{w}(a_0^+\gamma) = !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{e_j}(z_j^*)!^{d'}(x_0)!^{e'_{j-1}}(z_{j-1}^*)\dots!^{e'_1}(z_1^*).$$

  Since no commutation rule applies between $!^{e_j}(z_j)$ and $!^{d'}(x_0)$ and since $\boldsymbol{w}(a_0^+\gamma)$ must rewrite in ab^* form or 0, an anihilation rule has to apply. Thus $e_j = d'$ and $z_j = x_0$ or x_1 where x_1 is the coefficient associated to the other premise of n if any, so that $!^{e_j}(z_j^*)!^{d'}(x_0) = !^{d'}(x_\epsilon^* x_0)$ by the box morphism rule.

  If $z_j = x_0$ then we get:

  $$\begin{aligned}\boldsymbol{w}(a_0^+\gamma) &= !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{d'}(x_0^*x_0)!^{e'_{j-1}}(z_{j-1}^*)\dots!^{e'_1}(z_1^*)\\ &= !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{e_{j+1}}(z_{j+1}^*)!^{e'_{j-1}}(z_{j-1}^*)\dots!^{e'_1}(z_1^*)\end{aligned}$$

  which is in ab^* form, thus $a_0^+\gamma$ is regular

  Otherwise n is a binary node having a second premise a_1 with weight $!^d(x_1)$ and $z_j = x_1$; in particular x_1 satisfies the same commutation rules than x_0 and we therefore have:

  $$\begin{aligned}\boldsymbol{w}(a_1^+\gamma) &= !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{e_j}(z_j^*)!^{d'}(x_1)!^{e'_{j-1}}(z_{j-1}^*)\dots!^{e'_1}(z_1^*)\\ &= !^{d_1}(y_1)\dots!^d_k(y_k)!^{e_l}(z_l^*)\dots!^{e_{j+1}}(z_{j+1}^*)!^{e'_{j-1}}(z_{j-1}^*)\dots!^{e'_1}(z_1^*)\end{aligned}$$

  which is in ab^* form so that $a_1^+\gamma$ is regular.

□

Remark 12.2.7. As a consequence maximal regular paths are those starting and ending in a w or in an conclusion node. However in the sequel we shall consider that a path starting of ending into a w-node is not regular, for example by putting 0 as the weight of the conclusion arrow of each w-node. Thus maximal regular paths are regular paths starting and ending in a conclusion.

12.3 Proof nets as operators

The ab^* theorem has another important consequence.

Theorem 12.3.1. *Let S be any non trivial model of Λ^*, that is an involutive monoid with a morphism ! and elements p, q, d, r, s and t satisfying Λ^* equations. If $\mathcal{R}$ is any proof net and γ any straight path in $\mathcal{R}$ then γ is persistent iff $\boldsymbol{w}(\gamma) \neq 0$ in S.*

Proof. We know that if γ is persistent then $\boldsymbol{w}(\gamma)$ rewrites to an ab^* form which cannot be zero in S, S being a non trivial model of Λ^*. Conversely if γ is non persistent $\boldsymbol{w}(\gamma)$ rewrites to 0 thus is 0 in S. □

This means that any model of Λ^* is suitable for computing persistent paths, which explains the variety of instances of the GoI that may be found in the litterature: the context semantics of Gontier-Abadi-Lévy, Danos-Regnier's Interaction Machines, and to begin with, the original presentation of Girards as an interpretation of proof nets as operators on the Hilbert space.

This section is devoted to the description of Girard's initial results with a strong emphasis on the combinatorial point of view. Apart from its historical interest it introduces basic concepts and terminology that are used in the whole theory and is also an introduction to further Girard's work extending the GoI to additives and later reconstructing the whole theory within the framework of Von Neumann algebras, two topics that will not be covered in this book.

12.3.1 Lifting partial permutations to operators on ℓ^2

It is not necessary to be expert in functional analysis to read the following, we will use very little properties of Hilbert spaces and operator algebras and will give the (very) basic definitions without proofs. Fundamentals on (separable) Hilbert spaces may be found in any good handbook, *e.g.*, [42].

12.3.1.1 Operators terminology

The Hilbert space ℓ^2 (technically we should write $\ell^2(\mathbf{N})$) is the complex vector space of sequences $x = (x_k)_{k\geq 0}$ of complex numbers such that $\|x\|_2 = \sum_{k\geq 0} |x_k|^2 < \infty$, equipped with the inner product $\langle x, y\rangle = \sum_{k\geq 0} x_k \bar{y}_k$.

For our purpose ℓ^2 should be thought of as generated by the particular sequences $e_i = (\delta_{ik})_{k\geq 0}$ (where δ_{ik} is the Kronecker symbol, 1 if $i = k$, 0

otherwise) that form a Hilbert basis: it is orthonormal for the inner product, that is $\langle e_i, e_j \rangle = \delta_{ij}$ for any i, j, and any element $x = (x_k)_{k \geq 0} \in \ell^2$ may uniquely be written as the infinite linear combination $x = \sum_{k \geq 0} x_k e_k$. Conversely the coordinates of the vector x may be computed by $x_k = \langle x, e_k \rangle$ for each k. Except for the topological constraint on $\|\_\|_2$, we may view ℓ^2 as an infinite dimensional euclidian space with a denumerable orthonormal basis.

A bounded operator u on ℓ^2 is a linear map $u : \ell^2 \to \ell^2$ satisfying $\|u\| = \sup_{\|x\|_2 \leq 1} \|u(x)\|_2 < \infty$. Being linear maps, operators can be composed and composition preserves boundedness so that bounded operators form a monoid with the identity operator as neutral. Bounded operators can also be summed and thus form a (non commutative) linear algebra.

There is a canonical duality on operators, namely adjointness: if u is a bounded operator, its adjoint u^* is the unique bounded operator satisfying $\langle u^*(x), y \rangle = \langle x, u(y) \rangle$ for all $x, y \in \ell^2$. Adjointness enjoy the standard properties: $(u^*)^* = u$, $(uv)^* = v^* u^*$, and also some properties relative to the norm making the algebra of bounded operators a $\mathbf{C}^*$-algebra [5].

The kernel of a bounded operator u is the closed subspace of vectors x such that $u(x) = 0$. The domain of u is the orthogonal subspace of its kernel. The codomain of u is the image of u, that is the closed subspace of vectors of the form $u(x)$ for $x \in \ell^2$.

12.3.1.2 Partial permutations and operators

For the GoI interpretation we shall consider only a small subset of the algebra of bounded operators. Given a partial permutation σ on $\mathbf{N}$ we can lift it into an operator u_σ on ℓ^2 defined by its action on the Hilbert basis (e_i):

$$u_\sigma(e_i) = \begin{cases} e_{\sigma(i)} & \text{if } i \in \operatorname{dom} \sigma \\ 0 & \text{otherwise} \end{cases}$$

When $i \notin \operatorname{dom} \sigma$ we will set $e_{\sigma(i)} = 0$ so as to write simply: $u_\sigma(e_i) = e_{\sigma(i)}$ for all $i \in \mathbf{N}$. An operator of the form u_σ for a partial permutation σ will be called a **monomial**.

We have $\langle u_\sigma^*(e_i), e_j \rangle = \langle e_i, u_\sigma(e_j) \rangle = \langle e_i, e_{\sigma(j)} \rangle$ so that $\langle u_\sigma^*(e_i), e_j \rangle = 1$ iff $i = \sigma(j)$ iff $j = \sigma^*(i)$, 0 otherwise. Therefore $\langle u_\sigma^*(e_i), e_j \rangle = \langle e_{\sigma^*(i)}, e_j \rangle$ for all i, j, which shows that $u_\sigma^* = u_{\sigma^*}$.

From this we deduce that the domain of u_σ is the codomain of u_σ^*, that is the subspace generated by the e_i's for $i \in \operatorname{dom} \sigma = \operatorname{codom} \sigma^*$. Symetrically the codomain of u_σ is the domain of u_σ^*, that is the subspace generated by the e_j's for $j \in \operatorname{codom} \sigma = \operatorname{dom} \sigma^*$. For this reason we will slightly improperly say that two monomials u_σ and $u_{\sigma'}$ have *disjoint* domains (instead of *orthogonal* domains) when $\operatorname{dom} \sigma$ and $\operatorname{dom} \sigma'$ are disjoint, *i.e.*, when $\sigma' \sigma^* = 0$.

Up to isomorphism the space $\ell^2 \otimes \ell^2$ is the Hilbert space $\ell^2(\mathbf{N} \times \mathbf{N})$ of doubly indexed sequences of complex numbers $x = (x_{kl})_{k,l \geq 0}$ such that $\sum_{k,l \geq 0} |x_{kl}|^2 < \infty$. Given x and y in ℓ^2 we denote $x \otimes y$ the $\ell^2 \otimes \ell^2$ element $x \otimes y = (x_k y_l)_{k,l \geq 0}$.

Just as before the space $\ell^2 \otimes \ell^2$ admits a Hilbert basis $(e_{ij})_{i,j\geq 0}$ where e_{ij} is the sequence $(\delta_{(i,j),(k,l)})_{k,l\geq 0}$, so that $e_{ij} = e_i \otimes e_j$ for all i, j.

The one-to-one map $\langle \_, \_ \rangle : \mathbf{N}^2 \xrightarrow{\sim} \mathbf{N}$ used to construct the box morphism on the $\mathbf{N}$ model may be lifted into an isomorphism $\varphi : \ell^2 \otimes \ell^2 \xrightarrow{\sim} \ell^2$ by setting $\varphi(e_i \otimes e_j) = e_{\langle i,j \rangle}$ for all i, j. This in turn can be used to define ! on operators by:

$$!(u) : \ell^2 \xrightarrow{\varphi^{-1}} \ell^2 \otimes \ell^2 \xrightarrow{\mathsf{Id}_{\ell^2} \otimes u} \ell^2 \otimes \ell^2 \xrightarrow{\varphi} \ell^2$$

which is immediately seen to be a morphism on the algebra of bounded operators, satisfying $!(u)^* = !(u^*)$. When $u = u_\sigma$ is a monomial this reduces to $u_\sigma(e_{\langle i,j \rangle}) = e_{\langle i,\sigma(j) \rangle}$ so that we have

$$!(u_\sigma) = u_{!(\sigma)}$$

Taking p, q, d, r, s and t as the monomials obtained by lifting the corresponding partial permutations defined in section 12.2.2, we therefore get a new model of the equational theory Λ^*, called the **Hilbert space model**.

Remark 12.3.2. As said before the Hilbert space model was the original presentation by Girard of the dynamic algebra. It was clear but not completely explicit that the operators used to interpret proof nets were monomials. Even more implicit were the equations these operators had to satisfy; they were extracted afterward thus defining the equational theory Λ^*. So the presentation chosen here reverses the chronology.

12.3.2 Graphs and matrices

In this section we will be working within the Hilbert space model of Λ^*, thus consider elements of Λ^* as bounded operators. Let $\mathcal{R}$ be a proof net together with its weight functor $\boldsymbol{w}(\_)$ in Λ^*. We want to give some description of maximal regular paths, that is regular paths starting and ending into conclusions of $\mathcal{R}$. We will begin with the matrix interpretation of proof nets, that was the original presentation of the GoI by Girard, made possible thanks to the linear algebra structure of the Hilbert space model.

We will use the functorial relation between graphs and matrices that we briefly recall: if G is an oriented weighted graph, with weights in some (semi)-ring, then one can represent it by its *weight matrix* W_G, a square matrix indexed by the set of G-nodes; the coefficient $(W_G)_{\mathsf{n}\mathsf{n}'}$ at row n, column n', is the sum of the weights of the arrows sourced on n' and targeted on n. Matrix calculation then shows that W_G^k is the weight matrix of the graph G^k of paths of length k in G.

If G' is another weighted graph having the same set of nodes we can consider paths aa' where a is an arrow in G and a' is an arrow in G'. Again matrix calculation shows that the weight matrix of these paths is $W_{G'}W_G$.

12.3.2.1 The execution formula

We call **interface nodes** in $\mathcal{R}$ the conclusion nodes and the nodes one conclusion of which is premise of a *cut*-node, that we will simply call *cut*-premise nodes. We denote by $\mathsf{C}(\mathcal{R})$ the set of *cut*-premise nodes and by $\mathsf{I}(\mathcal{R})$ the set of interface nodes.

We consider maximal straight normal paths between interface nodes. These are of two kinds:

Axiom paths: have the form $\delta_1^*\delta_2$ where δ_1 and δ_2 are two maximal descent paths starting with the two distinct conclusions of an *ax*-node and ending in two (not necessarily distinct) interface nodes. We will write $\alpha : \mathsf{n} \to \mathsf{n}'$ to express the fact that α is an axiom path from nodes n to n'. Note that there are exactly two axiom paths crossing a given *ax*-node, that are inverse one to the other.

Cut paths: have the form $a_{\mathsf{c}}^{+}{a'_{\mathsf{c}}}^{-}$ where a_{c} and a'_{c} are the two distinct premises of a cut c. A cut path have weight equal to 1.

A maximal straight path is a straigth path from a conclusion or *w*-node of $\mathcal{R}$ to a conclusion or *w*-node of $\mathcal{R}$. Thanks to the remark 12.2.7, we will not consider paths starting or ending in a *w*-node and call *execution set* the set of maximal regular paths starting and ending in $\mathsf{I}(\mathcal{R}) \setminus \mathsf{C}(\mathcal{R})$.

Let $\Pi(\mathcal{R})$ and $\Sigma(\mathcal{R})$ be the two weighted graphs having the same set of nodes, namely $\mathsf{I}(\mathcal{R})$, the interface nodes of $\mathcal{R}$ and axiom paths as arrows of $\Pi(\mathcal{R})$, cut paths as arrows of $\Sigma(\mathcal{R})$.

We associate to $\Pi(\mathcal{R})$ and $\Sigma(\mathcal{R})$ their weight matrices indexed by the interface nodes and with coefficients in the operator algebra Λ^*: $\pi(\mathcal{R})$, the *proof matrix* also denoted $\pi_{\mathcal{R}}$, and $\sigma(\mathcal{R})$, the *cut matrix* also denoted $\sigma_{\mathcal{R}}$. If n and n' are two interface nodes then the coefficient at row n, column n' of $\pi(\mathcal{R})$ and $\sigma(\mathcal{R})$ are respectively:

$$\pi(\mathcal{R})_{\mathsf{n}\mathsf{n}'} = \sum_{\alpha:\mathsf{n}'\to\mathsf{n}} \boldsymbol{w}(\alpha),$$

$$\sigma(\mathcal{R})_{\mathsf{n}\mathsf{n}'} = \begin{cases} 1 & \text{if } \mathsf{n} \text{ and } \mathsf{n}' \text{ are the two distinct premises of a cut node,} \\ 0 & \text{otherwise.} \end{cases}$$

An easy computation shows that $\sigma_{\mathcal{R}}^2$ is the matrix having 1 on the diagonal positions n, n for $\mathsf{n} \in \mathsf{C}(\mathcal{R})$, 0 elsewhere, so that $\sigma_{\mathcal{R}}^2$ may be considered as a projector on the subspace generated by the *cut*-premise nodes (this will be made more precise in the next section).

Say that γ is a Π-*path of length* k or simply a Π^k-*path* if γ is regular of the form $\gamma = \alpha_0\sigma_1\alpha_1 \dots \sigma_k\alpha_k$ where the α_i's are axiom paths and the σ_i's are cut paths. Denote as $\Pi^k(\mathcal{R})$ the set of Π-paths of length k. Note that $\Pi^0(\mathcal{R})$ is the set of axiom paths, that is the set of arrows of $\Pi(\mathcal{R})$.

Given a Π^k-path $\gamma : \mathsf{n} \to \mathsf{n}'$ define π_γ to be the matrix with $\boldsymbol{w}(\gamma)$ at row n' and column n, 0 elsewhere. We thus have:

$$(\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k\pi_{\mathcal{R}} = \sum_{\gamma\in\Pi^k(\mathcal{R})} \pi_\gamma$$

We now have all the ingredients to understand that the execution formula:

$$\begin{aligned}\mathsf{Ex}(\mathcal{R}) &= (1-\sigma_{\mathcal{R}}^2)\sum_{k\geq 0}(\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k\pi_{\mathcal{R}}(1-\sigma_{\mathcal{R}}^2)\\ &= (1-\sigma_{\mathcal{R}}^2)\sum_{k\geq 0}\sum_{\gamma\in\Pi^k(\mathcal{R})}\pi_\gamma(1-\sigma_{\mathcal{R}}^2)\end{aligned}$$

represents the weight matrix of the graph of maximal regular paths in $\mathcal{R}$: the sum part is the weight matrix of all paths from interface nodes to interface nodes, the $1-\sigma_{\mathcal{R}}^2$ on both sides eliminate all the paths that are not sourced or targeted on a conclusion node.

The remaining question is to check whether the infinite sum in the execution formula converges in some sense. We present two answers in the following sections, after discussing the properties of the execution formula.

12.3.2.2 Elementary properties of the matrix interpretation of $\mathcal{R}$

We begin with some properties of (weight of) paths that we express in the framework of operator algebra, although they actually only use the equational theory so are valid in any model.

Lemma 12.3.3. *If $\alpha : \mathsf{n} \to \mathsf{n}'$ and $\alpha' : \mathsf{n} \to \mathsf{n}''$ are two distinct axiom paths sourced on a same node n then the monomials $\boldsymbol{w}(\alpha)$ and $\boldsymbol{w}(\alpha')$ have disjoint domains. Symetrically, if $\alpha : \mathsf{n}' \to \mathsf{n}$ and $\alpha' : \mathsf{n}'' \to \mathsf{n}$ have the same target then $\boldsymbol{w}(\alpha)$ and $\boldsymbol{w}(\alpha')$ have disjoint codomains.*

Proof. Write $\alpha = \delta_1^*\delta_2$ and $\alpha' = {\delta'_1}^*\delta'_2$. Since α and α' are distinct we must have $\delta_1 \neq \delta'_1$, otherwise δ_2 and δ'_2 would begin with the same premise of the ax-node source of δ_1 and δ'_1, thus would be equal too.

Hence we have $\delta_1 = \delta_{10}a_0\delta$ and $\delta'_1 = \delta'_{10}a_1\delta$ where a_0 and a_1 are the two distinct premises of a binary node (multiplicative or contraction). Therefore $\boldsymbol{w}(\delta_1) = \boldsymbol{w}(\delta)x_0\boldsymbol{w}(\delta_{10})$ and $\boldsymbol{w}(\delta'_1) = \boldsymbol{w}(\delta)x_1\boldsymbol{w}(\delta'_{10})$ where $\boldsymbol{w}(\delta)$, $\boldsymbol{w}(\delta_{10})$ and $\boldsymbol{w}(\delta'_{10})$ are positive words, and x_0, x_1 are the weights of a_0 and a_1. In particular we have $x_0^*x_1 = 0$ thus $\boldsymbol{w}(\delta_1^*)\boldsymbol{w}(\delta'_1) = 0$ thus $\boldsymbol{w}(\alpha)\boldsymbol{w}({\alpha'}^*) = \boldsymbol{w}(\delta_2)\boldsymbol{w}(\delta_1^*)\boldsymbol{w}(\delta'_1)\boldsymbol{w}(\delta'_2)^* = 0$. □

Corollary 12.3.4. *If γ, γ' are two distinct Π^k-paths sourced on a same interface node n then $\boldsymbol{w}(\gamma)$ and $\boldsymbol{w}(\gamma')$ have disjoint domains. Similarly if γ and γ' are targeted on the same node n' then $\boldsymbol{w}(\gamma)$ and $\boldsymbol{w}(\gamma')$ have disjoint codomains.*

Proof. Since γ and γ' are distinct but sourced on a same node there is some i such that $\gamma = \alpha_0 \dots \sigma_i\alpha_i \dots \sigma_k\alpha_k$ and $\gamma' = \alpha_0 \dots \sigma_i\alpha'_i \dots \sigma'_k\alpha'_k$ where α_i and α'_i

are two distinct axiom paths sourced on the same node. Thus $\boldsymbol{w}(\alpha_i)$ and $\boldsymbol{w}(\alpha'_i)$ have disjoint domains, from which it is immediate to deduce that it is the same for $\boldsymbol{w}(\gamma)$ and $\boldsymbol{w}(\gamma')$. □

Remark 12.3.5. The fact that γ and γ' have the same lenght is not mandatory; in fact the weights of any two paths starting from a same node and diverging at some point have disjoint domains because the point of divergence must be a binary node taken upwardly, each path choosing his own premise. Then one may generalize the reasonning proving lemma 12.3.3.

Let us now see some properties of the matrix interpretation of $\mathcal{R}$. Firstly we define the Hilbert space $(\ell^2)^{\mathsf{I}(\mathcal{R})} = \bigoplus_{\mathsf{n}\in\mathsf{I}(\mathcal{R})} \ell^2_{\mathsf{n}}$ where for each $\mathsf{n} \in \mathsf{I}(\mathcal{R})$, $\ell^2_{\mathsf{n}} = \ell^2$ is a copy of ℓ^2. Let $e_{\mathsf{n}i}$ be the column vector having e_i at position n, 0 elsewhere. Then the family of vectors $(e_{\mathsf{n}i},\ \mathsf{n} \in \mathsf{I}(\mathcal{R}),\ i \in \mathbf{N})$ is a Hilbert basis of the space $(\ell^2)^{\mathsf{I}(\mathcal{R})}$.

Note that $\sigma^2_{\mathcal{R}} e_{\mathsf{n}i} = e_{\mathsf{n}i}$ if n is a *cut*-premise node, 0 otherwise, so that as announced above σ^2_R is the projector on the subspace of $(\ell^2)^{\mathsf{I}(\mathcal{R})}$ generated by $(e_{\mathsf{n}i},\ \mathsf{n} \in \mathsf{C}(\mathcal{R}), i \in \mathbf{N})$. Thus $1 - \sigma^2_{\mathcal{R}}$ is the projector on the dual subpace generated by the $e_{\mathsf{n}i}$'s for all conclusion nodes n.

A matrix E indexed by $\mathsf{I}(\mathcal{R})$ and whose coefficients live in Λ^* is a bounded operator on the space $(\ell^2)^{\mathsf{I}(\mathcal{R})}$. In particular it has an adjoint E^* that is given by $(E^*)_{\mathsf{n}\mathsf{n}'} = (E)^*_{\mathsf{n}'\mathsf{n}}$ for each $\mathsf{n}, \mathsf{n}' \in \mathsf{I}(\mathcal{R})$.

Proposition 12.3.6. *For any $k \geq 0$ the matrix $\varepsilon_k = (\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k \pi_{\mathcal{R}}$ is hermitian, that is satisfy $\varepsilon^*_k = \varepsilon_k$.*

For any $k \geq 0$ and any $\mathsf{n}, \mathsf{n}' \in \mathsf{I}(\mathcal{R})$ the coefficient $(\varepsilon_k)_{\mathsf{n}\mathsf{n}'}$ is a sum of monomials; the monomials occuring on row n have pairwise disjoint codomains and the monomials occuring on column n' have pairwise disjoint domains.

The matrices ε_k act as partial permutations on the basis $(e_{\mathsf{n}i})$, i.e.*, there is a partial permutation σ_k on the set $\mathsf{I}(\mathcal{R}) \times \mathbf{N}$ such that for any $\mathsf{n} \in \mathsf{I}(\mathcal{R})$ and $i \in \mathbf{N}$, $\varepsilon_k e_{\mathsf{n}i} = e_{\mathsf{n}'j}$ if $\sigma_k(\mathsf{n}, i) = (\mathsf{n}', j)$, 0 otherwise.*

Proof. As seen before the coefficient $(\varepsilon_k)_{\mathsf{n}\mathsf{n}'}$ is the sum of the weights of Π^k-paths $\gamma : \mathsf{n}' \to \mathsf{n}$. Since $\boldsymbol{w}(\gamma : \mathsf{n} \to \mathsf{n}')^* = \boldsymbol{w}(\overline{\gamma} : \mathsf{n}' \to \mathsf{n})$, we have $(\varepsilon^*_k)_{\mathsf{n}\mathsf{n}'} = (\varepsilon_k)^*_{\mathsf{n}'\mathsf{n}} = \sum_{\gamma:\mathsf{n}\to\mathsf{n}'} \boldsymbol{w}(\gamma)^* = \sum_{\gamma:\mathsf{n}'\to\mathsf{n}} \boldsymbol{w}(\gamma) = (\varepsilon_k)_{\mathsf{n}\mathsf{n}'}$.

If w_1 and w_2 are mononomials occuring on column n' of ϵ_k then there are two Π-paths γ_1 and γ_2 of length k that have the same source n' and such that $w_1 = \boldsymbol{w}(\gamma_1)$, $w_2 = \boldsymbol{w}(\gamma_2)$. Thus w_1 and w_2 have disjoint domains. Symetrically if w_1 and w_2 occurs on a same row, they have disjoint codomains.

By definition of $e_{\mathsf{n}i}$ and matrix calculation we have $(\varepsilon_k e_{\mathsf{n}i})_{\mathsf{n}'} = (\varepsilon_k)_{\mathsf{n}'\mathsf{n}}(e_i)$. For a given n, monomials occuring in the column vector $((\varepsilon_k)_{\mathsf{n}'\mathsf{n}})_{\mathsf{n}'\in\mathsf{I}(\mathcal{R})}$ have disjoint domains thus there is at most one n' such that $(\varepsilon_k e_{\mathsf{n}i})_{\mathsf{n}'}$ is nonzero. Furthermore if such an n' exists it is a sum of monomials having disjoint domains thus there is a unique one, u_σ, such that $u_\sigma(e_i) = e_{\sigma(i)} \neq 0$. Therefore $\varepsilon_k e_{\mathsf{n}i} = e_{\mathsf{n}'\sigma(i)}$ which shows that ε_k acts on the basis $(e_{\mathsf{n}i})$.

Suppose $\varepsilon_k e_{\mathsf{m}j} = \varepsilon_k e_{\mathsf{n}i} = e_{\mathsf{n}'\sigma(i)}$. Thus there is a monomial u_τ occuring on row n', column m of ε_k such that $u_\tau(e_j) = u_\sigma(e_i)$. Since monomials on row n'

have disjoint codomains we must have $\mathsf{m} = \mathsf{n}$ and $u_\tau = u_\sigma$, that is $\tau = \sigma$, thus $j = i$ since σ is a partial permutation. This shows that ε_k is injective, thus acts as a partial permutation on the $e_{\mathsf{n}i}$'s. □

Remark 12.3.7. For simplicity we have stated the proposition for matrices of the form $(\pi_{\mathcal{R}}\sigma_R)^k\pi_R$, however it holds verbatim for matrices of the form $(\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k$.

12.3.3 Strong topology and strong normalization

Let us remind that a sequence (u_k) of bounded operators converges strongly towards 0 when $\|u_k\|$ converge to 0. In his first paper on GoI "An interpretation of system F" [15], Girard showed:

Theorem 12.3.8. *If $\mathcal{R}$ is a typed proof net in MELL_2 (MELL with second order quantifiers) then the operator $\sigma_{\mathcal{R}}\pi_{\mathcal{R}}$ is stronly nilpotent, thus the execution formula converges for the strong topology.*

12.3.3.1 A short account on Girard's proof

The result is obtained by a method analogous to the proof of strong normalisation for system F adapted to the framework of linear logic, which is sometimes called *linear reducibility*.

The first step is to define an orthogonality relation on Λ^* (viewed as the algebra of bounded operators on ℓ^2): $u \perp v$ iff uv is strongly nilpotent. We then define the orthogonal of a set S of operators as $S^\perp = \{v \in \Lambda^*,\ \forall u \in S,\ u \perp v\}$. A *type* is a reflexive set, that is a set S equal to its biorthogonal. Then is given an interpretation of MELL_2 formulas defined by induction, *e.g.*, $(A \otimes B)^* = (A^\perp \parr B^\perp)^{*\perp} = \{pup^* + qvq^*, u \in A^*, v \in B^*\}^{\perp\perp}$, $!(A)^* = (?A^\perp)^{*\perp} = \{!(u),\ u \in A^*\}^{\perp\perp}$, $(\forall\alpha A^\perp)^* = (\bigcup_{X \text{ type}} A[X/\alpha]^*)^\perp$.

The last step is to prove that when $\mathcal{R}$ is a proof net of type A whose cut formulas are respectively $A_1, \ldots, A_k$ then $\pi_{\mathcal{R}} \in (A \parr (A_1 \otimes A_1^\perp) \parr \cdots \parr (A_k \otimes A_k^\perp))^*$ while $\sigma_{\mathcal{R}} \in (A^\perp \otimes (A_1^\perp \parr A_1) \otimes \ldots \otimes (A_k^\perp \parr A_k))^*$ which by definition of orthogonality entails that $\sigma_R\pi_R$ is strongly nilpotent.

12.3.3.2 A more path oriented approach

Let (σ_k) be a sequence of partial permutations and suppose (u_{σ_k}) converges strongly towards 0. Let $x = \sum_{i\geq 0} \lambda_i e_i \in \ell^2$ be such that $\|x\|^2 = \sum_{i\geq 0} |\lambda_i|^2 \leq 1$. The monomial u_{σ_k} acting as a partial permutation on the Hilbert basis (e_i), the $u_{\sigma_k}(e_i)$'s form an orthonormal system of vectors. We therefore get $\|u_{\sigma_k}(x)\|^2 = \|\sum_{i\geq 0} \lambda_i u_{\sigma_k}(e_i)\|^2 \leq \sum_{i\geq 0} |\lambda_i|^2 \leq 1$ which shows that $\|u_{\sigma_k}\| \leq 1$. Furthemore if $i \in \operatorname{dom}\sigma_k$ then $\|u_{\sigma_k}(e_i)\| = \|e_{\sigma_k(i)}\| = 1$ which shows that $\|u_{\sigma_k}\| = 1$ if $\operatorname{dom}\sigma_k$ is nonempty, 0 otherwise.

Thus the fact that $\|u_{\sigma_k}\|$ converges towards 0 means that the sequence is finite: there is some K such that for all $k \geq K$, $\|u_{\sigma_k}\| = 0$, thus $u_{\sigma_k} = 0$. The $\varepsilon_k = (\pi_R\sigma_R)^k\pi_R$ acting as partial permutations on the basis, Girard's theorem shows that the sum converges because it is actually finite.

The execution formula being a matrix representation of the execution set (the set of maximal regular paths), this means that the execution set is finite and therefore suggest the slightly more general result:

Theorem 12.3.9. *Let $\mathcal{R}$ be a (not necessarily typed) proof net; if $\mathcal{R}$ is strongly normalizing then the set of regular paths in $\mathcal{R}$ is finite and the operator $\sigma_R\pi_R$ is nilpotent. Therefore the execution formula is finite, thus converges.*

Remark 12.3.10. This means that the strong convergence of the execution is actually consequence of the strong normalisation of typed proof nets, which explains why Girard's proof of strong convergence is analogous to the proof of strong normalization of typed proof nets.

Proof. By induction on the strong normalization norm of $\mathcal{R}$, it results from the equivalence between regular and persistent, and the fact that, if $\mathcal{R}'$ is a retract by a one step reduction of $\mathcal{R}$, the set of persistent paths in $\mathcal{R}$ is the lifting (by the lifting functor) of the set of persistent paths in $\mathcal{R}'$. By induction hyptohesis the latter is finite, thus the former is finite. □

Remark 12.3.11. The converse, if $\mathcal{R}$ has finitely many regular/persistent paths then $\mathcal{R}$ is strongly normalizable, is also true but trickier to prove.

12.3.4 Weak topology and cycles

We are still left with the question of the convergence of the execution formula in the case $\mathcal{R}$ is not strongly normalizable, *e.g.*, when $\mathcal{R}$ is the translation of a λ-term. This was firstly addressed by Girard who proposed to use the weak topology on operators [17]; the proposition was proved adequate for dealing with untyped proof nets by Malacaria and Regnier [37].

12.3.4.1 Weak topology

The weak topology is one of the numerous topologies on operator algebras. A sequence (u_k) of bounded operators converges weakly towards 0 if for all $x, y \in \ell^2$, the inner product $\langle u(x), y\rangle$ converges towards 0 which in turn is equivalent to ask that for any $i, j \in \mathbf{N}$, the inner product $\langle u_k(e_i), e_j\rangle$ converges to 0.

When applied to monomial operators this yields a combinatorial caracterisation: given a sequence (σ_k) of partial permutations, the sequence (u_{σ_k}) converges weakly to 0 iff for all $i, j \in \mathbf{N}$ there is some K such that for all $k \geq K$, if $i \in \operatorname{dom}\sigma_k$ then $\sigma_k(i) \neq j$. This because $\langle u_{\sigma_k}(e_i), ej\rangle = \langle e_{\sigma_k(i)}, e_j\rangle$ takes only value 0 or 1, 1 iff $\sigma_k(i) = j$.

If we specialize even further on the sequence $(u_\sigma^k)_{k\geq 0}$ for a given partial permutation σ we get a caracterisation of weak nilpotency:

Lemma 12.3.12. *The operator u_σ is weakly nilpotent,* i.e.*, the sequence $(u_\sigma^k)_{k\geq 0}$ converges weakly to 0 iff for all $k > 0$, u_σ^k has no fixpoint in the basis (e_i) of ℓ^2, that iff for all $k > 0$ and $i \in \mathbf{N}$, $u_\sigma^k(e_i) \neq e_i$ or equivalently if for all $k > 0$, σ^k has no fixpoint.*

Proof. Suppose for all $k > 0$, σ^k has no fixpoint and let $i, j, l \in \mathbf{N}$ such that $u_\sigma^l(e_i) = e_j$, *i.e.*, $\sigma^l(i) = j$. Let $k > 0$; then $\sigma^{l+k}(i) \neq j$ if defined, otherwise $j = \sigma^l(i)$ would be a fixpoint of σ^k. Thus for all $k > l$, $u_\sigma^k(e_i) \neq e_j$ so that u_σ is weakly nilpotent.

Conversely suppose $\sigma^k(i) = i$ for some $k > 0$. Then $u_\sigma^{kl}(e_i) = e_i$ for all $l \geq 0$ and u_σ cannot be weakly nilpotent. □

12.3.4.2 Cycles

A **straight cycle** in a proof net $\mathcal{R}$ is a straight path γ such that γ is composable with itself and $\gamma\gamma$ is a straight path (thus also a straight cycle).

Theorem 12.3.13 (No square persistent path)**.** *If γ is a straight cycle in $\mathcal{R}$ then the square path $\gamma\gamma$ is not persistent.*

Proof. By induction on a special reduction of γ. If γ is normal, that is crosses no *cut* then γ changes direction at most one time in an axiom node thus cannot be a straight cycle because a straight cycle has to change direction an even number of times in order to begin and end in the same direction.

Let c be a special cut for γ, thus also special for $\gamma\gamma$. If γ has no residual by the c-elimination step then γ, thus $\gamma\gamma$, is not persistent. Otherwise let γ' be the residual of γ. If γ' is not composable with itself then $\gamma\gamma$ has no residual and is therefore not persistent. If γ' is composable with itself then γ' is a straight cycle and we may conclude that $\gamma'\gamma'$ is not persistent by induction on the length of the special reduction. □

Corollary 12.3.14. *If γ is a straight cycle then the domain and codomain of $\boldsymbol{w}(\gamma)$ are disjoint.*

Proof. Let $w = \boldsymbol{w}(\gamma)$. Since $\gamma\gamma$ is not persistent, $ww = 0$. Let $e_j \in \operatorname{dom} w \cap \operatorname{codom} w$. Since $e_j \in \operatorname{codom} w$ there is an i such that $e_j = w(e_i)$. Therefore $w(e_j) = w^2(e_i) = 0$ thus $e_j \notin \operatorname{dom} w$, a contradiction. □

Theorem 12.3.15. *Let $\mathcal{R}$ be a proof net (typed or untyped). Then the matrix $\pi\sigma$ viewed as an operator on $(\ell^2)^{\mathsf{l}(\mathcal{R})}$ is weakly nilpotent. Thus the execution formula weakly converges.*

Proof. We show that for any $k > 0$, $\mu_k = (\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k$ has no fixpoint, which according to lemma 12.3.12 entails the weak nilpotency of $\pi_{\mathcal{R}}\sigma_{\mathcal{R}}$.

Let $e_{\mathsf{n}i}$ be a basis vector and suppose that $\mu_k e_{\mathsf{n}i} = e_{\mathsf{n}i}$. By the proposition 12.3.6, since all monomials occuring at column n of μ_k have disjoint domains, at most one of them, say u_σ, is such that $u_\sigma(e_i) = e_{\sigma(i)} \neq 0$. If such a u_σ exists, since $\mu_k e_{\mathsf{n}i} = e_{\mathsf{n}i}$, it occurs at row n thus at the diagonal position n, n in the matrix μ_k and we have $\sigma(i) = i$.

Therefore u_σ is the weight of a path γ of the form $\sigma_1\alpha_1 \ldots \sigma_k\alpha_k$ starting and ending downardly in n: γ is a straight cycle that satisfies $w(\gamma)(e_i) = e_i$, a contradiction. □

12.3.5 Conclusion

In this section operator theory has been used essentially to turn the equationnal theory Λ^* or its $\mathbf{N}$ model of partial permutations into a linear algebra, that is to enrich Λ^* with a sum. This allowed for the matrix interpretation of proof net and for expressing the execution formula, then to define the topological ingredients describing the convergence of the execution formula. However, the two convergences may actually be reduced to combinatorial properties: strong convergence is consequence of strong normalisation of the proof net wich in turn is equivalent to the finiteness of its set of execution paths, weak convergence is consequence of the no square cycle property. This suggests that there might be some more general and more abstract notion of proof net still satisfying one or the other notion of convergence, which is somehow one motivation of Girard's subsequent work, firstly for extending the GoI to additives [18], and later for adapting the theory in the framework of Von Neumann algebras [19].

Back to our MELL proof nets it is important for the sequel to note that all these combinatorial properties, that is all the properties not involving operator topology, are actually valid in any model of Λ^*, thus provable in the equational theory itself. Typically the fact that the weights of two distinct straigth paths starting from a same node in the same direction have disjoint domains is also provable in the equational theory. We are soon going to see that this is the key point allowing us to view the GoI interpretation as a token machine.

12.4 The Interaction Abstract Machine

Let us rephrase the disjoint domains property stated in corollary 12.3.4 in a slightly more general way; we say that two paths γ and γ' are **initially separing** if $\gamma = e_1 \dots e_{k-1} e_k \dots e_N$ and $\gamma' = e_1 \dots e_{k-1} e'_k \dots e'_{N'}$ where $e_k = a^-$ and $e'_k = a'^-$ are upward edges associated to the distinct premises a and a' of a binary node ($\otimes$ or c). Symetrically γ and γ' are **finally separing** if $\overline{\gamma}$ and $\overline{\gamma}'$ are initially separing.

Theorem 12.4.1 (Separing paths)**.** *If γ and γ' are two regular initially separing paths then $\boldsymbol{w}(\gamma)$ and $\boldsymbol{w}(\gamma')$ have disjoint domains in the $\mathbf{N}$ model (thus in the Hilbert model also), that is:*

$$\boldsymbol{w}(\gamma')\boldsymbol{w}(\gamma)^* = 0$$

Symetrically if γ and γ' are finally separing then γ and γ' have disjoint codomains:

$$\boldsymbol{w}(\gamma')^*\boldsymbol{w}(\gamma) = 0$$

Proof. The proof is adapted from the one of the corollary. Let $\delta = e_1 \dots e_{k-1}$ be the common prefix of γ and γ'. We show that $\boldsymbol{w}(\delta e'_k)\boldsymbol{w}(\delta e_k)^* = 0$ which entails the result.

Let a_k and a'_k be the arrows underlying e_k and e'_k so that $e_k = a_k^-$ and $e'_k = a'^-_k$. By hypothesis, a_k and a'_k are the two premises of a binary node,

thus $\boldsymbol{w}(e'_k)\boldsymbol{w}(e_k)^* = \boldsymbol{w}(a'_k)^*\boldsymbol{w}(a_k) = 0$. In the **N** model, $\boldsymbol{w}(\delta)\boldsymbol{w}(\delta)^*$ is a partial identity thus $\boldsymbol{w}(\delta e'_k)\boldsymbol{w}(\delta e_k)^* = \boldsymbol{w}(e'_k)\boldsymbol{w}(\delta)\boldsymbol{w}(\delta)^*\boldsymbol{w}(e_k)^* = 0$ in the **N** model. □

Remark 12.4.2. With a light generalization of the ab^* theorem 12.2.3, we can deduce that $\boldsymbol{w}(\gamma')\boldsymbol{w}(\gamma)^* = 0$ in any model of the equational theory Λ^*, *i.e.*, that $\Lambda^* \vdash \boldsymbol{w}(\gamma')\boldsymbol{w}(\gamma)^* = 0$. Indeed $\boldsymbol{w}(\delta e'_k)\boldsymbol{w}(\delta e_k)^* = \boldsymbol{w}(e_k^*\delta^*\delta e'_k)$ is the weight of a path which is however non straight. Nevertheless the ab^* theorem still holds for non straight paths, although a bit more technical to prove, therefore $\boldsymbol{w}(\gamma)\boldsymbol{w}(\gamma')^*$ is provably equal to 0 since it cannot rewrite into an ab^* form.

The disjoint domain property expresses the determinism of the GoI: consider the **N** model interpretation of Λ^* and let $i \in \mathbf{N}$ and n a node in $\mathcal{R}$. Then for any N there is at most one regular path γ of length N starting downardly from n such that $i \in \operatorname{dom} \boldsymbol{w}(\gamma)$ and at most one regular path γ' of length N starting upwardly from n such that $i \in \operatorname{dom} \boldsymbol{w}(\gamma')$.

This suggests to turn the GoI interpretation into a token machine: in the **N** model tokens are integers, in the Hilbert model tokens are the basis vectors e_i. If one enters upwardly by a conclusion of $\mathcal{R}$ with a token then one will find at most one maximal regular path, thus an execution path, such that the input token is in the domain of its weight. This execution path can be computed step by step, letting the weight of each edge act on the token.

In order to build on this idea we will first introduce a slight variation of the **N** model.

12.4.1 The interaction model

Let $\Sigma_{\mathcal{S}}$ be the first order signature containing the following symbols:

- a constant symbol $\square$ (the *empty token*);
- two constant symbols P and Q (the *multiplicative tokens*);
- a constant symbol D;
- two unary symbols of function $R(\_)$ and $S(\_)$;
- a binary symbol of function $T(\_, \_)$.

An **interaction token** is a closed term on the signature $\Sigma_{\mathcal{S}}$. An **exponential token** is an interaction token built on the sub-signature $\{D, R(\_), S(\_), T(\_, \_)\}$.

An **interaction stack** is a sequence $\pi = (u_i)_{i\geq 0}$ of interaction tokens that are almost all equal to $\square$, *i.e.*, there is $N \in \mathbf{N}$ such that $u_i = \square$ for all $i > N$ which we write $\pi = u_0 \cdots u_N$. Note that this writing is not unique, *e.g.*, we have $u_0 \cdots u_N = u_0 \cdots u_N \cdot \square = \ldots$ Two stacks $\pi = u_0 \cdots u_N$ and $\pi' = u'_0 \cdots u'_{N'}$ are equal if one, say π, is prefix of the other, that is $u_i = u'_i$ for $i \leq N$, and for $N + 1 \leq i \leq N'$, $u'_i = \square$. The *size* of the stack π is the smallest N such that $u_i = \square$ for all $i \geq N$. The empty stack is the stack of size 0 and is denoted $\square$.

Remark 12.4.3. We define stacks as *infinite* sequence so that every stack, including the empty stack, has as many first elements as needed: for any π, any l there are unique tokens $u_1, \ldots, u_l$ and a unique π_l such that $\pi = u_1 \cdot \cdots \cdot u_l \cdot \pi_l$.

We denote $\mathcal{S}$ the set of interaction stacks. The **interaction model** or simply **$\mathcal{S}$ model** is the set of partial permutations on the set $\mathcal{S}$. The interpretation of Λ^*-terms in the interaction model is given by:

- $p_{\mathcal{S}}(\pi) = P \cdot \pi, \quad q_{\mathcal{S}}(\pi) = Q \cdot \pi;$
- $d_{\mathcal{S}}(\pi) = D \cdot \pi;$
- $r_{\mathcal{S}}(u \cdot \pi) = R(u) \cdot \pi, \quad s_{\mathcal{S}}(u \cdot \pi) = S(u) \cdot \pi;$
- $t_{\mathcal{S}}(u_0 \cdot u_1 \cdot \pi) = T(u_1, u_0) \cdot \pi$
- if σ is a partial permutation on interaction stacks then $!_{\mathcal{S}}(\sigma)$ is defined by: $!_{\mathcal{S}}(\sigma)(u \cdot \pi) = u \cdot \sigma(\pi)$.

When w is a closed Λ^*-term we denote by $w_{\mathcal{S}}$ its interpretation as a partial permutation on $\mathcal{S}$.

It is an exercise to check that these constructions satisfie the Λ^* equations. Observe in particular that for $x = r, s, t$, the interpretation $x_{\mathcal{S}}$ has full domain ($x^*_{\mathcal{S}} x_{\mathcal{S}} = 1$) because any stack has a first element and that $!_{\mathcal{S}}(1) = 1$ for the same reason.

12.4.1.1 Embedding the interaction model in the N model.

Recall from section 12.2.2 that the **N** model is based on a pairing function $\langle \_, \_ \rangle : \mathbf{N}^2 \xrightarrow{\sim} \mathbf{N}$ that we associate on the left: $\langle n_1, \ldots, n_{k+1} \rangle = \langle n_1, \langle \ldots, \langle n_k, n_{k+1} \rangle \ldots \rangle \rangle$. We suppose further that $\langle 0, 0 \rangle = 0$, which is true for each of the 2 well known pairing functions.

We define a mapping $\mathbf{N}(\_)$ from the interaction model to the **N** model by:

- $\mathbf{N}(\Box) = 0$.
- $\mathbf{N}(P) = 1$, $\mathbf{N}(Q) = 2$, $\mathbf{N}(D) = 3$. These value are arbitrary, any other would do provided $\mathbf{N}(P) \neq \mathbf{N}(Q)$.
- $\mathbf{N}(R(u)) = \rho(\mathbf{N}(u))$, $\mathbf{N}(S(u)) = \sigma(\mathbf{N}(u))$ where ρ and σ are the two partial permutations used to define r and s in the **N** model.
- $\mathbf{N}(T(u_0, u_1)) = \tau(\mathbf{N}(u_0), \mathbf{N}(u_1))$ where τ is the partial permutation used to define t in the **N** model.
- Thanks to the condition $\langle 0, 0 \rangle = 0$ if the stacks $u_0 \ldots u_N$ and $u_0 \ldots u_M$ are equal then $\langle \mathbf{N}(u_0), \ldots, \mathbf{N}(u_N), 0 \rangle = \langle \mathbf{N}(u_0), \ldots, \mathbf{N}(u_M), 0 \rangle$ so that we may define $\mathbf{N}(u_1 \cdot \cdots \cdot u_N) = \langle \mathbf{N}(u_1), \ldots, \mathbf{N}(u_N), 0 \rangle$.

- Since $\langle \_, \_ \rangle$ is one-to-one, $\mathbf{N}(\_)$ is injective on stacks. Thus if σ is a partial permutation on stacks we may define the partial permutation $\mathbf{N}(\sigma)$ on $\mathbf{N}$ by:
$$\mathbf{N}(\sigma)(\mathbf{N}(\pi)) = \mathbf{N}(\sigma(\pi)).$$

Theorem 12.4.4. *For any closed Λ^*-term w the set $\mathbf{N}(\mathcal{S}) \subset \mathbf{N}$ is invariant by the action of $w_{\mathbf{N}}$ and we have:*
$$\mathbf{N}(w_{\mathcal{S}}) = w_{\mathbf{N}} \mid_{\mathbf{N}(\mathcal{S})} .$$
Hence the restriction to $\mathbf{N}(\mathcal{S})$ of the $\mathbf{N}$ model is a submodel isomorphic for the Λ^* structure to the interaction model.

The proof is immediate, by induction on w.

12.4.2 The Interaction Abstract Machine (IAM)

The $\mathbf{IAM}_0$ is the machine defined by the weighting of a proof net $\mathcal{R}$ in the interaction model. A state of the machine is a pair (e, π) where e is an edge in $\mathcal{R}$ and π is an interaction stack. The transitions are:

- $(a_0^-, \pi) \to (a_1^+, \boldsymbol{w}_{\mathcal{S}}(a_1)(\pi))$ if a_0 and a_1 are the two conclusions of an ax-node;
- $(a_0^+, \pi) \to (a_1^-, \pi)$ if a_0 and a_1 are the two premises of a *cut*-node (the weight of the premise of a *cut*-node is always 1);
- $(a^+, \pi) \to (a'^+, \boldsymbol{w}_{\mathcal{S}}(a')(\pi))$ if a and a' are respectively premise and conclusion of a same node;
- $(a'^-, \pi) \to (a^-, \boldsymbol{w}_{\mathcal{S}}(a)^*(\pi))$ if a and a' are respectively premise and conclusion of a same node and $\pi \in \operatorname{codom} \boldsymbol{w}_{\mathcal{S}}(a')$.

A $\mathbf{IAM}_0$-run is given by an interaction stack π_0, the *input stack* of the run, and a sequence of transitions $(e_1, \pi_1) \to (e_2, \pi_2) \to \cdots \to (e_N, \pi_N)$ such that $\pi_1 = \boldsymbol{w}_{\mathcal{S}}(e_1)(\pi_0)$. From the definition of transitions, in particular the constraint on the codomain in the last clause, we immediately get:

Theorem 12.4.5. *A sequence $(e_i, \pi_i)_{1 \le i \le N}$ of $\mathbf{IAM}_0$ states is a $\mathbf{IAM}_0$-run on input π_0 iff for each $1 \le i \le N$, $\pi_i = \boldsymbol{w}_{\mathcal{S}}(e_i)(\pi_{i-1})$.*

When this is the case $\gamma = (e_i)_{1 \le i \le N}$ is a regular path, $\boldsymbol{w}_S(\gamma)(\pi_0) = \pi_N$ and for any regular path γ' starting in the same direction than γ, if $\pi_0 \in \operatorname{dom} \boldsymbol{w}_S(\gamma')$ then one of γ and γ' is prefix of the other.

In other terms the $\mathbf{IAM}_0$ is a deterministic device for computing regular paths. The deterministic observation is due to the already observed fact that all transitions *in the same direction* available on a given node have pairwise disjoint domains, thus for a given state at most one upward and at most one downard transition may be applied. Actually the $\mathbf{IAM}_0$ is *bideterministic* in the sense that at any point during a run one can reverse direction, in which case there will be no other choice than rewinding the run, eventually ending on the starting node in the input state.

12.4.2.1 The interaction model with 2 stacks

Let $\pi = (u_i)_{i\geq 0}$ be an interaction stack. The *d-prefix of* π is the d-tuple $(u_0, \ldots, u_{d-1})$ and the *d-suffix of* π the substack π_d defined by $\pi = u_0 \cdot \cdots \cdot u_{d-1} \cdot \pi_d$. If σ is a partial permutation on interaction stacks then $!^d(\sigma)$ acts only on d-suffixes, leaving the d-prefixes unchanged, that is:

$$!^d_{\mathcal{S}}(\sigma)(u_0 \cdot \cdots \cdot u_{d-1} \cdot \pi) = u_0 \cdot \cdots \cdot u_{d-1} \cdot \sigma(\pi)$$

This suggests that instead of acting at depth d on the stack we may track the current depth step by step by splitting the stack in a prefix B of elements at depth less than d and an active part E at depth d. Elements move from one stack to the other when the depth changes, from E to B when entering a box, from B to E when exiting a box.

Building on this idea let $\mathcal{B}$ be the set of **bistacks**, *i.e.*, pairs (E, B) where E, the **balanced stack**, and B, the **box stack**[3], are interaction stacks. Given a partial permutation σ on $\mathcal{S}$ let $\bar{\sigma}$ be the partial permutation on $\mathcal{B}$ defined by $\bar{\sigma}(E, B) = (\sigma(E), B)$. In particular if x is a Λ^* coefficient we set $x_{\mathcal{B}} = \bar{x}_{\mathcal{S}}$

Let $\beta : \mathcal{B} \to \mathcal{B}$ be given by $\beta(E, u \cdot B)) = (u \cdot E, B)$. Then we have:

$$\overline{!^d_{\mathcal{S}}(\sigma)} = \beta^d \circ \bar{\sigma} \circ \beta^{*d}.$$

Accordingly we define $!_{\mathcal{B}}(\sigma) = \beta\sigma\beta^*$. When w is a closed Λ^*-term we will denote $w_{\mathcal{B}}$ its interpretation as a partial permutation on $\mathcal{B}$.

Theorem 12.4.6. *For any Λ^* closed term w we have:*

$$w_{\mathcal{B}} = \bar{w}_{\mathcal{S}}$$

*As a consequence the set of partial permutations on $\mathcal{B}$ form a model of Λ^*, the $\mathcal{B}$ **model**, which is isomorphic to the $\mathcal{S}$ model.*

If a is an arrow in a proof net $\mathcal{R}$ with associated coefficient x_a then we define its **$\mathcal{B}$-weigth** $\boldsymbol{w}_{\mathcal{B}}(a)$ by:

- $\boldsymbol{w}_{\mathcal{B}}(a) = x_{a\mathcal{B}}$ is a is not exiting a box, *i.e.*, a is not premise of a !-node or a p-node;
- $\boldsymbol{w}_B(a) = \beta$ if a is premise of a !-node;
- $\boldsymbol{w}_{\mathcal{B}}(a) = t_{\mathcal{B}}\beta$ if a is premise of a p-node.

The definition of $\mathcal{B}$-weight is extended to edges and paths in the natural way.

Remark 12.4.7. Contrarily to the $\mathbf{N}$ or the $\mathcal{S}$-weight, the $\mathcal{B}$-weight of a path is not the interpretation in $\mathcal{B}$ of its Λ^* weight, that is, we don't have in general $\boldsymbol{w}_{\mathcal{B}}(\gamma) = \boldsymbol{w}(\gamma)_{\mathcal{B}}$. However both terms are strongly related:

Theorem 12.4.8. *If γ is a path starting at depth d_s and ending at depth d_t in a proof net $\mathcal{R}$ then:*

$$\beta^{d_s}\boldsymbol{w}_{\mathcal{B}}(\gamma)(\beta^*)^{d_t} = \boldsymbol{w}(\gamma)_{\mathcal{B}} = \overline{\boldsymbol{w}_{\mathcal{S}}(\gamma)}.$$

[3]The letter B being taken by the *B*ox stack, we choose E for the balanced stack because of the french translation *Équilibré* of *Balanced*.

12.4.2.2 Exponential branches

Let n be the root node of a ?-tree, d the depth of n, n_0 be a node in the tree at depth d_0, and $\delta : \mathsf{n}_0 \to \mathsf{n}$ be the corresponding exponential branch. Put $l = d_0 - d$ the number of boxes that are exited between n_0 and n, that we call the *lift* of the exponential branch δ. We define a term $u_\delta[x_0, \dots, x_l]$ with exactly $l+1$ free variables on the sub-signature $\{R(\_), S(\_), T(\_,\_)\}$ of $\Sigma_{\mathcal{S}}$ by induction on the length of δ:

- if δ has length 0 then $l = 0$ and $u_\delta = x_0$;
- if $\delta = \delta_0 a^+$ where a is the left (resp right) premise of a c-node then $u_\delta = R(u_{\delta_0})$ (resp. $S(u_{\delta_0})$);
- if $\delta = \delta_0 a^+$ where a is the premise of a p-node then the target node of δ_0 is the source node of a, thus lies in a box at depth $d+1$ so that the lift of δ_0 is $l-1$. By induction u_{δ_0} depends on variables x_0, …, x_{l-1} and we define $u_\delta = T(u_{\delta_0}, x_l)$.

Lemma 12.4.9. *With the notations just defined for any stacks E and B and any tokens u_0, …, u_l:*

$$\boldsymbol{w}_{\mathcal{B}}(\delta)(u_0 \cdot E, u_1 \cdots u_l \cdot B) = (u_\delta[u_0, u_1, \dots, u_l] \cdot E, B)$$

In particular, if n_0 is a d-node leaf of the exponential tree and a_0 is its premise then

$$\boldsymbol{w}_{\mathcal{B}}(a_0^+\delta)(E, u_1 \cdots u_l \cdot B) = (u_\delta[D, u_1, \dots, u_l] \cdot E, B)$$

Proof. The second part is immediate consequence of the first one and of the definition of the weight $d_{\mathcal{B}}$ of a_0: $d_{\mathcal{B}}(E, B) = (D \cdot E, B)$.

To keep notations light we set $w = \boldsymbol{w}_{\mathcal{B}}(\delta)$ and $w_0 = \boldsymbol{w}_{\mathcal{B}}(\delta_0)$. The proof is by induction on δ.

If δ has length 0 then $u_\delta = x_0$, $l = 0$ and $w = 1$ thus $w(u_0 \cdot E, B) = (u_\delta[u_0] \cdot E, B)$.

If $\delta = \delta_0 a^+$ where a is left premise of a c-node then $w = r_{\mathcal{B}} w_0$ and by induction hypothesis $w_0(u_0 \cdot E, u_1 \cdots u_l \cdot B) = (u_{\delta_0}[u_0, \dots, u_l] \cdot E, B)$ thus

$$\begin{aligned} w(u_0 \cdot E, u_1 \cdots u_l \cdot B) &= r_{\mathcal{B}} w_0(u_0 \cdot E, u_1 \cdots u_l \cdot B) \\ &= r_{\mathcal{B}}(u_{\delta_0}[u_0, \dots, u_l] \cdot E, B) \\ &= R(u_{\delta_0}[u_0, \dots, u_l]) \cdot E, B) \\ &= (u_\delta[u_0, \dots, u_l] \cdot E, B) \end{aligned}$$

by definition of u_δ. Same computation replacing $r_{\mathcal{B}}$ by $s_{\mathcal{B}}$ and R by S if a is right premise of a c-node.

If $\delta = \delta_0 a^+$ where a is premise of a p-node then $w = t_{\mathcal{B}} \beta w_0$. By induction hypothesis $w_0(u_0 \cdot E, u_1, \cdots, u_{l-1} \cdot B) = (u_{\delta_0}[u_0, \dots, u_{l-1}] \cdot E, B)$ for any stack

B. Thus

$$\begin{aligned} w(u_0 \cdot E, u_1 \cdots u_l \cdot B) &= t_{\mathcal{B}} \beta w_0(u_0 \cdot E, u_1 \cdots u_l \cdot B) \\ &= t_{\mathcal{B}} \beta(u_{\delta_0}[u_0, \ldots, u_{l-1}] \cdot E, u_l \cdot B) \\ &= t_{\mathcal{B}}(u_l \cdot u_{\delta_0}[u_0, \ldots, u_{l-1}] \cdot E, B) \\ &= (T(u_{\delta_0}[u_1, \ldots, u_{l-1}], u_l) \cdot E, B) \\ &= (u_\delta[u_0, \ldots, u_l] \cdot E, B) \end{aligned}$$

by definition of u_δ. □

12.4.2.3 The IAM

The **IAM** is the machine given by a proof net $\mathcal{R}$ together with its weight function $\boldsymbol{w}_{\mathcal{B}}$. The states of the machine are the pairs $(e, (E, B))$ where e is an edge in $\mathcal{R}$ and $(E, B) \in \mathcal{B}$ is a bistack. The transitions are deduced from the $\mathbf{IAM}_0$ as follows:

- $(a_0^-, (E, B))) \to (a_1^+, \boldsymbol{w}_{\mathcal{B}}(a_1)(E, B))$ if a_0 and a_1 are the two conclusions of an *ax*-node;
- $(a_0^+, (E, B)) \to (a_1^-, (E, B))$ if a_0 and a_1 are the two premises of a *cut*-node;
- $(a^+, (E, B)) \to (a'^+, \boldsymbol{w}_{\mathcal{B}}(a')(E, B))$ if a and a' are respectively premise and conclusion of a same node;
- $(a'^-, (E, B)) \to (a^-, \boldsymbol{w}_{\mathcal{B}}(a)^*(E, B))$ if a and a' are respectively premise and conclusion of a same node and $(E, B) \in \operatorname{codom} \boldsymbol{w}_{\mathcal{B}}(a')$.

A **IAM**-run is defined as a $\mathbf{IAM}_0$-run: an input bistack (E_0, B_0) and a sequence of transitions $(e_{i-1}, (E_{i-1}, B_{i-1})) \to (e_i, (E_i, B_i))$. We get the same result as before: the **IAM** is a bideterministic device for computing regular paths.

Remark 12.4.10. All transitions but the ones traversing a box frontier leave the B stack invariant. A token is popped from the E stack and pushed on the B stack each time a box is entered, and conversely each time a box is exited, thus, if we start from a node at depth 0 with $B_0 = \square$, the empty stack, then at each step the size of B_i is the current depth, that is the number of boxes that have been entered. This is the reason why we call B the *box stack*.

12.4.3 Legal paths

Legal paths were defined by Asperti and Laneve in [3] in the framework of λ-calculus but their definition is easily adaptable to proof net. In this section we will suppose that proof nets have no exponential axioms, that is no axiom node the conclusion of which have type $!A$ and $?A^\perp$. As a consequence of this hypothesis the leaves of any exponential tree are either d-nodes or w-nodes.

Note that up to some η-expansion of exponential axioms this hypothesis is realised and is in particular satisfied for nets that are translation of lambda-terms.

In this section we will only consider $\mathcal{B}$-weights of paths. To keep notations light for any path γ we will denote $\boldsymbol{w}_\gamma = \boldsymbol{w}_B(\gamma)$.

12.4.3.1 Well balanced paths

Well balanced paths (**w.b.p.** in short) are defined by induction:

- If a and a' are the two premises of a *cut*-node then $a^+ a'^-$ is a w.b.p.
- If γ is a w.b.p. sourced and targeted on multiplicative nodes n and n' and if a_0, a_1 and a'_0, a'_1 are the premises of respectively n and n' then for $\epsilon = 0, 1$, $a_\epsilon^+ \gamma a_\epsilon'^-$ is a w.b.p.
- If γ is a w.b.p. sourced on the root node n of a ?-tree and targeted on a !-node n', n_0 is a d-node leaf of the exponential tree, $\delta : \mathsf{n}_0 \to \mathsf{n}$ is the corresponding exponential branch, a_0 is the premise of n_0, and a' is the premise of the !-node then $a_0^+ \delta \gamma a'^-$ and $a'^+ \overline{\gamma} \overline{\delta} a_0^-$ are w.b.p.
- If a and a' are the two conclusions of an *ax*-node and γa^- and $a'^+ \gamma'$ are w.b.p. then $\gamma a^- a'^+ \gamma'$ is a w.b.p.

The following properties of w.b.p. are easily checked by induction:

Proposition 12.4.11. *Let γ be a w.b.p. in a proof net $\mathcal{R}$. Then:*

- *γ starts downardly and ends upwardly. Thus every w.b.p. is straight.*
- *$\overline{\gamma}$ is a w.b.p.*
- *The source and target nodes of γ, if not ax nodes, are dual: $\otimes$ and $\parr$ or ? and !.*
- *Furthermore if $\mathcal{R}$ is typed then the first and last edge of γ have dual types A and $A^\perp$.*
- *If $\mathcal{R}$ reduces in $\mathcal{R}'$ and γ has a non trivial residual γ' in $\mathcal{R}'$ then γ' is a w.b.p.*

Remark 12.4.12. Note the similarity between the definition of w.b.p. and the cut elimination steps. The main difference is that w.b.p. jump in one step from the conclusion of ?-trees to their leaves, where this is multiple cut elimination steps. For this reason and also because w.b.p. always link dual nodes they are sometimes called **virtual cuts**.

Theorem 12.4.13 (Balanced invariant)**.** *Let γ be a regular w.b.p. in a proof net $\mathcal{R}$. There is a partial permutation σ_γ on stacks such that* $\operatorname{dom} \boldsymbol{w}_\gamma = \mathcal{S} \times \operatorname{dom} \sigma_\gamma$ *and for any $(E, B) \in \operatorname{dom} \boldsymbol{w}_\gamma$:*

$$\boldsymbol{w}_\gamma(E, B) = (E, \sigma_\gamma(B)).$$

Remark 12.4.14. In other terms a **IAM** run starting from the source node of a w.b.p. γ with the input bistack (E, B) doesn't depend on E until arriving at the end of γ. Intuitively the proof below shows that all tokens pushed on the balanced stack are popped before reaching the end of γ, but that no token originally in E are popped.

Proof. For simplicity we will drop the subscript $\mathcal{B}$ in the coefficients, *e.g.*, write p for $p_{\mathcal{B}}$. Note that the property for γ entails the property for $\overline{\gamma}$ because γ is balanced iff $\overline{\gamma}$ is balanced and $\boldsymbol{w}_{\overline{\gamma}} = \boldsymbol{w}_\gamma^*$.

Assume γ is a w.b.p. We show by induction on the definition of γ that there is a partial permutation σ_γ on stacks such that for any stack B, if $(E, B) \in \operatorname{dom} \boldsymbol{w}_\gamma$ for some E then $(E', B) \in \operatorname{dom} \boldsymbol{w}_\gamma$ for any E' and $\boldsymbol{w}_\gamma(E', B) = (E', \sigma_\gamma(B))$.

In the base case, $\gamma = a^+ a'^-$ for two premises a, a' of a cut then $\boldsymbol{w}_\gamma = 1$ so actually any (E, B) is in $\operatorname{dom} \boldsymbol{w}_\gamma$ and we just have to take $B' = B$.

Suppose $\gamma = a_\epsilon^+ \gamma_0 a_\epsilon'^-$ where a_ϵ and a'_ϵ are premise of multiplicative nodes. Then $\boldsymbol{w}_\gamma = x^* \boldsymbol{w}_{\gamma_0} x$ where $x = p$ if $\epsilon = 0$, q if $\epsilon = 1$.

Let $(E, B) \in \operatorname{dom} \boldsymbol{w}_\gamma$. Then $\boldsymbol{w}_\gamma(E, B) = x^* \boldsymbol{w}_{\gamma_0} x(E, B) = x^* \boldsymbol{w}_{\gamma_0}(X \cdot E, B)$ where X is P or Q depending on the value of ϵ. Thus $(X \cdot E, B) \in \operatorname{dom} \boldsymbol{w}_{\gamma_0}$ so by induction $(X \cdot E', B) \in \operatorname{dom} \boldsymbol{w}_{\gamma_0}$ for any stack E' and we have $\boldsymbol{w}_{\gamma_0}(X \cdot E', B) = (X \cdot E', \sigma_{\gamma_0}(B))$, thus $\boldsymbol{w}_\gamma(E', B) = x^*(X \cdot E', \sigma_{\gamma_0}(B)) = (E', \sigma_{\gamma_0}(B))$ which shows that $(E', B) \in \operatorname{dom} \boldsymbol{w}_\gamma$ and that we may take $\sigma_\gamma = \sigma_{\gamma_0}$.

Suppose $\gamma = a_0^+ \delta \gamma_0 a'^-$ where a_0 is premise of a d-node n_0, δ is the exponential branch from n_0 to the root of its exponential tree and a' is premise of a !-node. Then $\boldsymbol{w}_\gamma = \beta^* \boldsymbol{w}_{\gamma_0} \boldsymbol{w}_\delta$ where $\boldsymbol{w}_\delta = \boldsymbol{w}_{\mathcal{B}}(a_0^+ \delta)$. Let l be the lift of the exponential branch δ. By lemma 12.4.9 we have a $\Sigma_{\mathcal{S}}$-term $u_\delta[x_0, \ldots, x_l]$ such that $\boldsymbol{w}_\delta(E, u_1 \cdots u_l \cdot B) = (u_\delta[D, u_1, \ldots, u_l] \cdot E, B)$ for any E, B and tokens $u_1, \ldots, u_l$.

Let $(E, B) \in \operatorname{dom} \boldsymbol{w}_\gamma$; by definition of stacks there are unique tokens $u_1, \ldots, u_l$ and a unique stack B_l such that $B = u_1 \cdots u_l \cdot B_l$. Writing $u = u_\delta[D, u_1, \ldots, u_l]$ for short we have $\boldsymbol{w}_\gamma(E, B) = \beta^* \boldsymbol{w}_{\gamma_0} \boldsymbol{w}_\delta(E, B) = \beta^* \boldsymbol{w}_{\gamma_0}(u \cdot E, B_l)$. Thus $(u \cdot E, B_l) \in \operatorname{dom} \boldsymbol{w}_{\gamma_0}$ so by induction $(u \cdot E', B_l) \in \operatorname{dom} \boldsymbol{w}_{\gamma_0}$ for any E' and $\boldsymbol{w}_{\gamma_0}(u \cdot E', B_l) = (u \cdot E', \sigma_{\gamma_0}(B_l))$. Thus $\boldsymbol{w}_\gamma(E', B) = \beta^*(u \cdot E', \sigma_{\gamma_0}(B_l)) = (E', u \cdot \sigma_{\gamma_0}(B_l))$. We get the result by setting $\sigma_\gamma(B) = u_\delta[D, u_1, \ldots, u_l] \cdot \sigma_{\gamma_0}(B_l)$.

Last suppose $\gamma = \gamma_1 \gamma_2$ where γ_1 and γ_2 are w.b.p. respectively targeted and sourced on an ax-node. Put $w_i = \boldsymbol{w}_{\gamma_i}$ so that $\boldsymbol{w}_\gamma = w_2 w_1$ and suppose $(E, B) \in \operatorname{dom} \boldsymbol{w}_\gamma$. Then $(E, B) \in \operatorname{dom} w_1$ by induction we have σ_1 such that $w_1(E, B) = (E, \sigma_1(B))$, thus $(E, \sigma_1(B)) \in \operatorname{dom} w_2$. By induction $(E', B) \in \operatorname{dom} w_1$ and $(E', \sigma_1(B)) \in \operatorname{dom} w_2$ for any E'. Furthermore we have σ_2 such that $w_2(E', \sigma_1(B)) = (E', \sigma_2(\sigma_1(B))$ so we get the result by setting $\sigma_\gamma = \sigma_2 \circ \sigma_1$. □

12.4.3.2 Box cycles

Recall that $\mathrm{d}(\mathsf{n})$ is the depth of the node n, that is the number of boxes containing n, and that the depth of an arrow a and its associated edges $\mathrm{d}(a) = \mathrm{d}(a^+) = \mathrm{d}(a^-)$ is the depth of the target node of a.

Lemma 12.4.15. *Let γ be a path in a proof net starting and ending from (not necessarily distinct) nodes at same depth d and crossing only nodes at depth at least d. Then for any bistack $(E, B) \in \operatorname{dom}\gamma$ there is a unique E' such that:*

$$\boldsymbol{w}_\gamma(E, B) = (E', B)$$

Note that the hypothesis on γ entails that the whole path is either starting and ending at depth 0, or contained in a box.

Proof. The uniquess is consequence of the fact that $\boldsymbol{w}_\gamma$ is a partial permutation on bistacks.

We reason by induction on γ. Let $\gamma = e\gamma'$ where e is the first edge of γ. Let n and n' be the source and target of e.

If $\mathrm{d}(\mathsf{n}) = \mathrm{d}(\mathsf{n}') = d$ then $\boldsymbol{w}_e = x^\epsilon_{\mathcal{B}}$ for some Λ^*-coefficient x, ϵ being 1 or $*$ depending on the direction of e. Thus $\boldsymbol{w}_e(E, B) = (x^\epsilon_{\mathcal{S}}(E), B)$. By induction hypothesis, since γ' starts and ends at depth d, $\boldsymbol{w}_{\gamma'}(x^\epsilon_{\mathcal{S}}(E), B) = (E', B)$ for some E' thus $\boldsymbol{w}_\gamma(E, B) = \boldsymbol{w}_{\gamma'}\boldsymbol{w}_e(E, B) = (E', B)$.

If $\mathrm{d}(\mathsf{n}) \neq \mathrm{d}(\mathsf{n}')$ then since $\mathrm{d}(\mathsf{n}') \geq \mathrm{d}(\mathsf{n})$, $\mathrm{d}(\mathsf{n}') = \mathrm{d}(\mathsf{n}) + 1$ which means that the edge e is entering a box b', *i.e.*, n is a box door node (! or p-node) and n' lies inside b'. Thus $\boldsymbol{w}_{\mathcal{B}}(e) = \beta^* x^*_{\mathcal{B}}$ where $x = t$ if n is a p-node, 1 if n is a !-node. Now since γ, thus γ', ends at depth d there is an edge $e' : \mathsf{n}'' \to \mathsf{n}'''$ in γ' that exit b'; in particular $\mathrm{d}(\mathsf{n}'') = d+1$ and n''' is a b' door at depth $\mathrm{d}(\mathsf{n}''') = d$. Let e' be the first such edge occuring in γ'; we have $\boldsymbol{w}_{e'} = y_{\mathcal{B}}\beta$ where y is t or 1 depending on the type of n'''. Write $\gamma' = \gamma''e'\gamma'''$ where $\gamma'' : \mathsf{n}' \to \mathsf{n}''$ is entirely lying inside b', thus crosses only nodes at depth greater than $d+1$, and γ'' starts on n''' and crosses only nodes at depth at least d.

So $\boldsymbol{w}_{\mathcal{B}}(\gamma) = \boldsymbol{w}_{\gamma'''}\, y_{\mathcal{B}}\beta\, \boldsymbol{w}_{\gamma''}\, \beta^* x^*_{\mathcal{B}}$ and using the induction hypothesis on γ'' and γ''' we can compute its action on (E, B) (we dropped the surrounding parentheses for readability):

$$\begin{array}{llll}
E,\quad B & \xrightarrow{x^*_{\mathcal{B}}} & u \cdot E_1,\quad B & \text{where } u \cdot E_1 = x^*_{\mathcal{S}}(E) \\
& \xrightarrow{\beta^*} & E_1,\quad u \cdot B & \\
& \xrightarrow{\boldsymbol{w}_{\gamma''}} & E_2,\quad u \cdot B & \text{by induction hypothesis on } \gamma'', \text{ for some } E_2 \\
& \xrightarrow{\beta} & u \cdot E_2,\quad B & \\
& \xrightarrow{y_{\mathcal{B}}} & v \cdot E_3\quad B & \text{where } v \cdot E_3 = y_{\mathcal{S}}(u \cdot E_2) \\
& \xrightarrow{\boldsymbol{w}_{\gamma'''}} & E'\quad B & \text{by induction hypothesis on } \gamma''', \text{ for some } E'.
\end{array}$$

□

Let b be a box in a proof net. A *b-path* is a path starting upwardly and ending downardly on b-door nodes (the !-node or some p-node) and such that all nodes crossed are inside b. We define **!-cycles** and **?-cycles** by induction (see figure 12.1 below for the general picture):

!-cycle, base case: a b-path starting and ending on the !-node of a box b.

?**-cycle:** $\gamma_1 \delta \overline{\gamma}_2$ where:

- $\gamma_1 : \mathsf{n}_1 \to \mathsf{n}$ and $\gamma_2 : \mathsf{n}_2 \to \mathsf{n}$ are two w.b.p. sourced respectively on some ?-nodes n_1 and n_2 and targeted on the !-node n; when n_1 and n_2 are both root nodes of some exponential trees the ?-cycle is said **initial**. When n_1 and n_2 are both dereliction nodes the ?-cycle is said **final**.
- δ is a !-cycle at n.

!**-cycle, induction case:** $\delta_0 \theta_1 \delta_1 \dots \theta_k \delta_k$ such that there is a box b satisfying:

- each δ_i is a b-path
- δ_0 starts upwardly on the !-node n of b, δ_k ends downwardly on n;
- for $i > 0$, δ_i starts upwardly on a p-node p_i of b; for $i < k$, δ_i ends downwardly on p_{i+1} (the p_is are not necessarily distinct);
- each θ_i is a ?-cycle starting downwardly and ending upwardly at p_i.

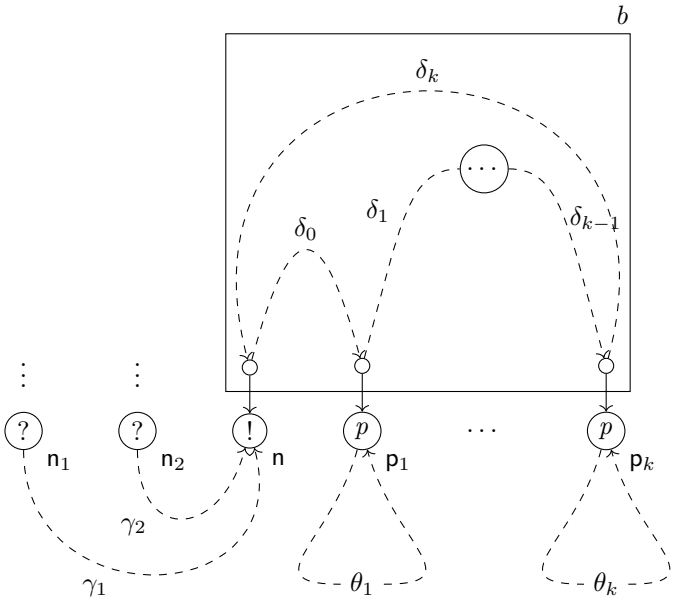


Figure 12.1: The general form of a ?-cycle: γ_i are w.b.p., θ_i are ?-cycles, the p_is are not necessarily distinct

Theorem 12.4.16 (Box invariant)**.** *Let δ be either a !-cycle or an initial ?-cycle. If δ is regular then there is a partial permutation τ_δ on stacks such that for any token u, any stack E and any stack B if $(E, B) \in \operatorname{dom} \boldsymbol{w}_\delta$ then $E \in \operatorname{dom} \tau_\delta$ and:*

$$\boldsymbol{w}_\delta(u \cdot E, B) = (u \cdot \tau_\delta(E), B)$$

Furthermore if δ is an initial ?-cycle, thus $\delta = \gamma_1\delta_0\overline{\gamma}_2$ for some w.b.p. γ_1 and γ_2, then $\gamma_1 = \gamma_2$.

Proof. By induction on δ. If δ is a b-path sourced and targeted on the !-node n of a box b then $\delta = a^-\delta_0 a^+$ where a is the premise of n and δ_0 is a path starting and ending on the source node of a, lying entirely in b, thus satisfying the conditions of the lemma 12.4.15. So we have $\boldsymbol{w}_\delta = \boldsymbol{w}_B(a)\boldsymbol{w}_{\delta_0}\boldsymbol{w}_B(a)^* = \beta\boldsymbol{w}_{\delta_0}\beta^*$ and we can compute its action on $(u \cdot E, B)$:

$$\begin{array}{rcrcl} u \cdot E, \quad B & \xrightarrow{\beta^*} & E, & u \cdot B & \\ & \xrightarrow{\boldsymbol{w}_{\delta_0}} & E', & u \cdot B & \text{where } E' \text{ is given by the lemma} \\ & \xrightarrow{\beta^*} & u \cdot E', & B & \end{array}$$

so that defining $\tau_\delta(E) = E'$ we get the result.

If $\delta = \gamma_1\delta_0\overline{\gamma}_2$ where γ_1 and γ_2 are w.b.p. and δ_0 is a !-cycle then we have $\boldsymbol{w}_\delta = \boldsymbol{w}^*_{\gamma_2}\boldsymbol{w}_{\delta_0}\boldsymbol{w}_{\delta_1}$ and:

$$\begin{array}{rcrcl} u \cdot E, \quad B & \xrightarrow{\boldsymbol{w}_{\delta_1}} & u \cdot E, & \sigma_{\gamma_1}(B)) \\ & \xrightarrow{\boldsymbol{w}_{\delta_0}} & u \cdot \tau_{\delta_0}(E), & \sigma_{\gamma_1}(B)) \end{array}$$

so that $\boldsymbol{w}_\delta(u \cdot E, B) = \boldsymbol{w}^*_{\delta_2}(u \cdot \tau_{\delta_0}(E), \sigma_{\gamma_1}(B))$.

If $\gamma_2 \neq \gamma_1$, as they are w.b.p. targeted on the same node and sourced on non-ax nodes they cannot be suffix one of the other, thus $\boldsymbol{w}_{\gamma_1}$ and $\boldsymbol{w}_{\gamma_2}$ have disjoint codomains. But $(u \cdot E, \sigma_{\gamma_1}(B)) = \boldsymbol{w}_{\gamma_1}(u \cdot E, B) \in \operatorname{codom} \boldsymbol{w}_{\gamma_1}$ and we have seen that the domain and codomain of a w.b.p. only depend on the box stack, thus $(u \cdot \tau_{\delta_0}(E), \sigma_{\gamma_1}(B)) \in \operatorname{codom} \boldsymbol{w}_{\gamma_1}$. Therefore, δ being supposed regular, we must have $\gamma_2 = \gamma_1$ and the computation may be continued:

$$\begin{aligned} \boldsymbol{w}_\delta(u \cdot E, B) &= \boldsymbol{w}^*_{\gamma_1}(u \cdot \tau_{\delta_0}(E), \sigma_{\gamma_1}(B)) \\ &= (u \cdot \tau_{\delta_0}(E), B) \end{aligned}$$

so setting $\tau_\delta = \tau_{\delta_0}$ we are done.

The last case is $\delta = \delta_0\theta_1\delta_1 \dots \theta_k\delta_k$, where:

- the δ_is are b-paths: for each $i > 0$, $\delta_i = a^-_{i-1}\delta'_0 a^+_i$ where the a_is are the premises of the door nodes and δ'_0 is entirely contained in b; in particular $a_0 = a_k$ is the premise of the !-node of b. Thus $\boldsymbol{w}_{\delta_0} = t_{\mathcal{B}}\beta\boldsymbol{w}_{\delta'_0}\beta^*$, $\boldsymbol{w}_{\delta_i} = t_{\mathcal{B}}\beta\boldsymbol{w}_{\delta'_i}\beta^* t^*_{\mathcal{B}}$ for $1 \leq i < k$ and $\boldsymbol{w}_{\delta_k} = \beta\boldsymbol{w}_{\delta'_k}\beta^* t^*_{\mathcal{B}}$.

  Since δ'_i lies entirely in the box b by lemma 12.4.15 we have a partial permutation ρ_i such that for any $(E, B) \in \operatorname{dom} \boldsymbol{w}_{\delta'_i}$, $\boldsymbol{w}_{\delta'_i}(E, B) = (\rho_i(E), B)$.

- the θ_is are ?-cycles starting and ending from p-nodes of b: $\theta_i = \alpha_i\theta'_i\alpha^*_i$ where α_i is the descent path starting from p_i down to the root node n_i of the exponential tree and θ'_i is a ?-cycle at n_i.

By induction hypothesis for any u, E, B such that $(u \cdot E, B) \in \operatorname{dom} \boldsymbol{w}_{\theta'_i}$ we have $\boldsymbol{w}_{\theta'_i}(u \cdot E, B) = (u \cdot \tau_{\theta'_i}(E), B)$ for some E_i depending only on E.

Denote as x_i the weight of α_i, so that $\boldsymbol{w}_{\theta_i} = x_i^* \boldsymbol{w}_{\theta'_i} x_i$. By lemma 12.4.9 there is a $\Sigma_{\mathcal{S}}$-term $u_{\alpha_i}[x_0, x_1, \dots, x_{l_i}]$ such that $x_i(u_0 \cdot E, u_1 \cdots u_{l_i} \cdot B_i) = (u_{\alpha_i}[u_0, \dots, u_{l_i}] \cdot E, B_i)$.

Putting all this together we have

$$\boldsymbol{w}_\delta = \beta \boldsymbol{w}_{\delta'_k} \beta^* t_{\mathcal{B}}^* \, x_k^* w_{\theta'_k} x_k \, \dots \, t_{\mathcal{B}} \beta \boldsymbol{w}_{\delta'_1} \beta^* t_{\mathcal{B}}^* \, x_1^* w_{\theta'_1} x_1 \, t_{\mathcal{B}} \beta \boldsymbol{w}_{\delta'_0} \beta^*$$

and we may compute the action of $\boldsymbol{w}_\delta$ on the bistack $(u \cdot E, B)$:

$$\begin{array}{lrll}
u \cdot E, \quad B & \xrightarrow{\beta^*} & E, \quad u \cdot B & \\
& \xrightarrow{\boldsymbol{w}_{\delta'_0}} & u_0 \cdot E'_0, \quad u \cdot B & u_0 \text{ and } E'_0 \text{ defined by } u_0 \cdot E'_0 = \rho_0(E) \\
& \xrightarrow{\beta} & u \cdot u_0 \cdot E'_0, \quad B & \\
& \xrightarrow{t_{\mathcal{B}}} & T(u_0, u) \cdot E'_0, \quad B & \\
& \xrightarrow{x_1} & u_{\alpha_1} \cdot E'_0, \quad B_1 & \text{where } B_1 \text{ is defined by } B = u_1 \cdots u_{l_1} \cdot B_1 \\
& & & \text{and } u_{\alpha_1} = u_{\alpha_1}[T(u_0, u), u_1, \dots, u_{l_i}] \\
& \xrightarrow{\boldsymbol{w}_{\theta'_1}} & u_{\alpha_1} \cdot E_1, \quad B_1 & \text{where } E_1 = \tau_{\theta'_i}(E'_0) \\
& \xrightarrow{x_1^*} & T(u_0, u) \cdot E_1, \quad B & \\
& \xrightarrow{t_{\mathcal{B}}^*} & u \cdot u_0 \cdot E_1, \quad B & \\
& \xrightarrow{\beta^*} & u_0 \cdot E_1, \quad u \cdot B & \\
& \xrightarrow{\boldsymbol{w}_{\delta'_1}} & u_1 \cdot E'_1, \quad u \cdot B & \text{where } u_1 \cdot E'_1 = \rho_1(u_0 \cdot E_1) \\
& \xrightarrow{\beta} & u \cdot u_1 \cdot E'_1, \quad B & \\
& \xrightarrow{t_{\mathcal{B}}} & T(u_1, u) \cdot E'_1, \quad B & \\
& & \vdots & \\
& \longrightarrow & T(u_{k-1}, u) \cdot E'_{k-1}, \quad B & \\
& \xrightarrow{x_k} & u_{\alpha_k} \cdot E'_{k-1}, \quad B_k & \\
& \xrightarrow{\boldsymbol{w}_{\theta'_k}} & u_{\alpha_k} \cdot E_k, \quad B_k & \\
& \xrightarrow{x_k^*} & u \cdot u_{k-1} \cdot E_k, \quad B & \\
& \xrightarrow{\beta^*} & u_{k-1} \cdot E_k, \quad u \cdot B & \\
& \xrightarrow{\boldsymbol{w}_{\delta'_k}} & E', \quad u \cdot B & \text{where } E' = \rho_k(u_{k-1} \cdot E_k) \\
& \xrightarrow{\beta} & u \cdot E', \quad B &
\end{array}$$

All the actions being one-to-one, the output E' is in one-to-one correspondance with the input E. Thus defining $\tau_\delta(E) = E'$, τ_δ is a partial permutation and we are done. □

Remark 12.4.17. This calculation shows a difference between w.b.p.'s weights and box cycles weights: we've seen that the former leave the balanced box invariant in particular because no transition ever pops a token from the initial input stack. In other words at any point during the run along a w.b.p. the balanced stack contains the input stack E as a substack. This is not the case for box cycles, who leave the box stack invariant but actually do pop tokens from the input stack B each time the box is exited. However the computation shows that any token popped will not be looked up, that is the following transitions will not depend on its value, until it is pushed back when coming back to the box before ending the cycle.

12.4.3.3 Legal paths

Recall that a *final* ?-cycle θ is a path of the form $\theta = \gamma_1\delta\overline{\gamma}_2$ where γ_1 and γ_2 are w.b.p. sourced on some dereliction nodes and δ is a !-cycle. When $\gamma_1 = \gamma_2$ we say that θ is *well parenthesised*. A w.b.p. γ in a proof net $\mathcal{R}$ is **legal** if any final ?-cycle contained in γ is well parenthesised.

Theorem 12.4.18. *A w.b.p. is legal iff it is regular.*

Proof. Let γ be a regular w.b.p. and $\theta = \gamma_1\theta_0\overline{\gamma}_2$ be a final ?-cycle contained in γ. As θ is final for $i = 1, 2$, we may decompose γ_i as $\gamma_i = \delta_i\gamma_i'$ where $\delta_i : \mathsf{d}_i \to \mathsf{n}_i$ is a descent path from a dereliction node d_i to the root node n_i of the corresponding exponential tree and γ_i' is a w.b.p. so that $\theta' = \gamma_i'\theta_0\overline{\gamma}_2'$ is an initial ?-cycle. Since θ and θ' are subpaths of γ that is supposed regular, θ and θ' are regular.

Let l_i be the lift of δ_i. By the exponential branch lemma 12.4.9 there is a $\Sigma_{\mathcal{S}}$ term u_{δ_i} such that $\boldsymbol{w}_{\delta_i}(u \cdot E, u_1 \cdots u_{l_i} \cdot B) = (u_{\delta_i}[u, u_1, \ldots, u_{l_i}] \cdot E, B)$ for any stacks E and B and any tokens u, u_1, …, u_{l_i}. Since θ is regular there is some stack E such $(u \cdot E, u_1 \cdots u_{l_i} \cdot B) \in \operatorname{dom}\theta$ and since θ' is an initial ?-cycle, using the box invariant theorem 12.4.16 we may compute the action of $\delta_1\theta'$ as:

$$\begin{array}{rcl} (u \cdot E, \quad u_1 \cdots u_{l_i} \cdot B) & \xrightarrow{\boldsymbol{w}_{\delta_1}} & (u_{\delta_1}[u, u_1, \ldots, u_{l_i}] \cdot E,\ B) \\ & \xrightarrow{\boldsymbol{w}_{\theta'}} & (u_{\delta_1}[u, u_1, \ldots, u_{l_i}] \cdot E',\ B) \end{array}$$

By the second part of the box invariant theorem, since $\theta' = \gamma_1'\theta_0\overline{\gamma}_2'$ is regular we have $\gamma_1' = \gamma_2'$, thus $\mathsf{n}_1 = \mathsf{n}_2$. Therefore Since δ_1 and δ_2 are maximal (because starting from dereliction nodes) exponential branches to the root of the now same exponential tree, if they are distinct they are finally separing thus $\boldsymbol{w}_{\delta_1}$ and $\boldsymbol{w}_{\delta_2}$ have disjoint codomains and therefore $(u_{\delta_1}[u, u_1, \ldots, u_{l_i}] \cdot E', B) \notin \operatorname{codom}\boldsymbol{w}_{\delta_2}$, contradicting our hypothesis that $(u \cdot E, u_1 \cdots u_{l_i} \cdot B) \in \operatorname{dom}\boldsymbol{w}_\theta = \operatorname{dom}\boldsymbol{w}_{\mathcal{B}}(\delta_1\theta'\overline{\delta}_2)$. Thus $\delta_1 = \delta_2$ and therefore $\gamma_1 = \gamma_2$.

For the converse we reason by induction on a special reduction of γ and show that if γ is legal then it has a residual γ' that is legal; by induction hypothesis γ' is thus regular and therefore by the equivalence regular/persistent, so is γ. Let c be the cut being reduced. If c is multiplicative or an axiom cut the result is immediate because by definition a w.b.p. cannot exchange the premises of a multiplicative cut, thus has a unique residual, and because the reduction preserves box cycles and w.b.p.

If c is exponential (non weakening) suppose for the contradiction that γ has no residual. Then γ has a subpath θ that exchanges the premises a_1 and a_2 of a contraction node premise of c. With the notations of the contraction elimination step p. 118, θ has the form $\theta = a_1^+ a_c^+ a'^- \delta a'^+ a_c^- a_2^-$ where δ is a subpath contained in the box b'. In particular $a_c^+ a'^- \theta a'^+ a_c^-$ is an initial ?-cycle, thus, since $a_1 \neq a_2$ are distinct, θ is (contained in) a ?-cycle that is not well parenthesised, contradicting our legality hypothesis on γ.

Therefore γ has a residual γ'. Let θ' be a final ?-cycle contained in γ'. Then one easily verifies that its lifting $\theta = \mathcal{L}\theta'$ is a final ?-cycle. With the notations of figure 12.1 we have $\theta = \alpha\gamma_1\delta_0\theta_1\delta_1 \dots \theta_k\delta_k\overline{\gamma}_1\overline{\alpha}$ where α is an exponential branch from a dereliction node d to the root node n_1 of the exponential tree and $\theta' = \alpha'\gamma'_1\delta'_0\theta'_1\delta'_1 \dots \theta'_k\delta'_k\overline{\gamma}''_1\overline{\alpha}''$ where δ'_i and θ'_i are residuals of δ_i and θ_i, γ'_1 and γ''_1 are residuals of γ_1 and α' and α'' are residuals of α. We are to show that $\gamma'_1 = \gamma''_1$, from which one immediately deduce that $\alpha' = \alpha''$ because both exponential branches are rooted on the same node and are residual of the same α.

Both γ'_1 and γ''_1 are targeted on the same !-node, thus have a common suffix. Let σ' be their longest common suffix. If $\gamma'_1 = \sigma'$ then σ' is a w.b.p. suffix of the w.b.p. γ''_1 thus we must have $\gamma''_1 = \sigma'$: by definition of w.b.p. the suffix of a w.b.p. is a w.b.p. only if it starts on an axiom node, which is not the case of σ'.

If σ' is a proper suffix of γ'_1, and also by symmetry of γ''_1, $\gamma'_1 = \rho' {a'_0}^+ \sigma'$ and $\gamma''_1 = \rho'' {a'_1}^+ \sigma'$ where, by maximality of σ', a'_0 and a'_1 are distinct premises of a binary node n'.

If n' is residual of a node n, then a'_0 and a'_1 are residuals of the premises a_0 and a_1 of n and γ_1 being the lift of γ'_1 has the form $\gamma_1 = \rho a_0^+ \sigma = \rho a_1^+ \sigma$ where σ is the lift of σ', ρ is the lift of ρ' and ρ''. Thus $a_0 = a_1$ and therefore $a'_0 = a'_1$, a contradiction.

Therefore n' is a node that has been added by the reduction of c, that is a ?-node of the same type as the ?-node premise of c, lying below an auxiliary door of (copies of) the box b premise of c.

From which we deduce that a'_0 and a'_1 are conclusion of some p-nodes auxiliary doors of (copies of) b. Since γ'_1 and γ''_1 are residuals of γ_1 we deduce that γ_1 exits the box b by some auxiliary door. But γ_1 being a w.b.p. cannot end downardly so after exiting b it must descend to a cut and cross this cut. Since c is special for γ and γ_1 is a subpath of γ, c is special for γ_1, a contradiction.

So the only possible case is that $\sigma' = \gamma'_1 = \gamma''_1$: thus θ' is well parenthesised and we have shown that γ' is legal. □

Chapter 13

Execution

Chapter 14

Games

Chapter 15

Traces

Part IV

Reaping the fruits of linearity

TODO

Appendices

Appendix A

Graphs

A.1 Basic definitions

We write ϵ for the empty sequence.

A **graph** is a quadruple $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathsf{s}, \mathsf{t})$ where $\mathcal{N}(\mathcal{G}) \coloneqq \mathcal{N}$ is the set of **nodes**, $\mathcal{A}(\mathcal{G}) \coloneqq \mathcal{A}$ is the set of **arrows**, and $\mathsf{s}_{\mathcal{G}} \coloneqq \mathsf{s}$ (the **source** function) and $\mathsf{t}_{\mathcal{G}} \coloneqq \mathsf{t}$ (the **target** function) are maps from $\mathcal{A}$ to $\mathcal{N}$. A graph is *finite* if it has finitely many nodes and arrows. Let n be a node and a be an arrow: if $\mathsf{s}(a) = n$ (resp. $\mathsf{t}(a) = n$) then a is called an **outgoing arrow** (resp. **incoming arrow**) of n. An **incident arrow** of n is any of an incoming or outcoming arrow.

use *vertex* rather than *node*? — Olivier

use *arc* rather than *arrow*? — Olivier

An **edge** e is given by an arrow a together with a direction. We write $e = a^+$ if e follows the arrow, and $e = a^-$ if e takes the opposite direction, meaning that we extend s and t to the set $\mathcal{E}(\mathcal{G})$ of edges of $\mathcal{G}$, by setting: $\mathsf{s}(a^+) \coloneqq \mathsf{s}(a)$, $\mathsf{t}(a^+) \coloneqq \mathsf{t}(a)$, $\mathsf{s}(a^-) \coloneqq \mathsf{t}(a)$ and $\mathsf{t}(a^-) \coloneqq \mathsf{s}(a)$.

Let $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{A}_1, \mathsf{s}_1, \mathsf{t}_1)$ and $\mathcal{G}_2 = (\mathcal{N}_2, \mathcal{A}_2, \mathsf{s}_2, \mathsf{t}_2)$. We say a $\mathcal{G}_2$ is a **subgraph** of $\mathcal{G}_1$ if $\mathcal{N}_2 \subseteq \mathcal{N}_1$, $\mathcal{A}_2 \subseteq \mathcal{A}_1$, and s_2 (resp. t_2) is the restriction of s_1 (resp. t_1) to $\mathcal{A}_2$.

A **graph morphism** from $\mathcal{G}_1$ to $\mathcal{G}_2$ is given by functions $f_{\mathcal{N}} : \mathcal{N}_1 \to \mathcal{N}_2$ and $f_{\mathcal{A}} : \mathcal{A}_1 \to \mathcal{A}_2$ such that $\mathsf{s}_2(f_{\mathcal{A}}(a)) = f_{\mathcal{N}}(\mathsf{s}_1(a))$ and $\mathsf{t}_2(f_{\mathcal{A}}(a)) = f_{\mathcal{N}}(\mathsf{t}_1(a))$ for each $a \in \mathcal{A}_1$. We denote $f \coloneqq (f_{\mathcal{N}}, f_{\mathcal{A}}) : \mathcal{G}_1 \to \mathcal{G}_2$ in this case. We say f is a **graph isomorphism** when $f_{\mathcal{N}}$ and $f_{\mathcal{A}}$ are bijections: in this case, $f^{-1} \coloneqq (f_{\mathcal{N}}^{-1}, f_{\mathcal{A}}^{-1})$ is also a graph morphism, and we write $f : \mathcal{G}_1 \xrightarrow{\sim} \mathcal{G}_2$.

The **empty graph** is the only graph whose set of nodes is empty. The **sum** of $\mathcal{G}_1$ and $\mathcal{G}_2$ is the graph $\mathcal{G}_1 + \mathcal{G}_2 \coloneqq (\mathcal{N}_1 + \mathcal{N}_2, \mathcal{A}_1 + \mathcal{A}_2, \mathsf{s}, \mathsf{t})$ where the sum of sets is their disjoint union, and $\mathsf{s}((i, a)) \coloneqq \mathsf{s}_i(a)$ and $\mathsf{t}((i, a)) \coloneqq \mathsf{t}_i(a)$.

A.2 Paths

Two edges e_0 and e_1 are **composable** if the target node of e_0 is the source node of e_1. A (possibly infinite) **path** in a graph $\mathcal{G}$ is a pair $\gamma = (n, \vec{e})$ where n is a node (the **source** of the path, also noted $\mathbf{s}(\gamma)$) and $\vec{e}$ is a (possibly infinite)

sequence of edges $(e_i)_{1\leq i\leq N}$ (with $N \in \mathbf{N}\cup\{\infty\}$), such that any two consecutive edges in $\vec{e}$ are composable:

- if $N > 0$ then $\mathsf{s}(e_1) = n$;
- and, for any $1 \leq i < N$, $\mathsf{t}(e_i) = \mathsf{s}(e_{i+1})$.

Note that we really should call these **undirected paths** as arrows can be crossed forwardly or reversely; as this is the only notion of path we need we choose to drop the *undirected* mention.

The **length** $|\gamma|$ of the path is N, and we say γ is *finite* if $|\gamma| \in \mathbf{N}$. We also use the notation ϵ_n for the empty path (n, ϵ) which has length 0.

Each path γ induces a sequence of nodes of length $|\gamma|+1$: an **occurrence of a node** in γ is an item in this sequence together with its rank in the sequence. As is usual, we will often abuse terminology and call node such an occurrence. A (occurrence of a) node n' is **internal** to γ if $n' = \mathsf{s}(e_i)$ with $i > 1$ (or equivalently $n' = \mathsf{t}(e_i)$ with $i < N$).

Let $\gamma = (n, (e_i)_{1\leq i\leq N})$ be a path. The **target** $\mathbf{t}(\gamma)$ of γ is n if γ is empty ($N = 0$), $\mathsf{t}(e_N)$ if γ is finite and nonempty ($1 \leq N < \infty$), undefined otherwise ($N = \infty$). When $\mathsf{s}(\gamma) = n$ and $\mathsf{t}(\gamma) = n'$, we say γ is a **path from** n **to** n'. Observe that the source and target of γ, as well as its sequence of internal nodes, are uniquely determined by the sequence of edges of γ, unless it is an empty path ϵ_n, in which case $\mathsf{s}(\gamma) = \mathsf{t}(\gamma) = n$ and γ has no internal node. We will thus often identify a path with its sequence of edges.

the question is mostly about the source node: target node and internal nodes are uniquely defined from edges even in the empty case — Olivier

what is the target of the empty sequence of edges? — Lionel

If $\gamma_0 = (n_0, (e_i)_{1\leq i\leq N_0})$ and $\gamma_1 = (n_1, (e_i)_{N_0+1\leq i\leq N_0+N_1})$ are two paths of respective lengths $N_0 < \infty$ and N_1 such that $\mathsf{t}(\gamma_0) = n_1 = \mathsf{s}(\gamma_1)$, we say that they are **composable** and write $\gamma_0\gamma_1$ for their **composition** or **concatenation**:

$$\gamma_0\gamma_1 = (n_0, (e_i)_{1\leq i\leq N_0+N_1})$$

so that $\mathsf{s}(\gamma_0\gamma_1) = \mathsf{s}(\gamma_0) = n_0$ and $\mathsf{t}(\gamma_0\gamma_1) = \mathsf{t}(\gamma_1)$ (possibly undefined). A path γ is said to be **closed** if $\mathsf{s}(\gamma) = \mathsf{t}(\gamma)$, and **open** otherwise.

A **prefix** (resp. **suffix**; **subpath**) of γ is any path γ' such that we can write $\gamma = \gamma'\gamma_2$ (resp. $\gamma = \gamma_1\gamma'$; $\gamma = \gamma_1\gamma'\gamma_2$).

When defined, composition is associative and the empty path ϵ_n is neutral when composed on the left with any path of source n, on the right with any path of target n. We thus have defined a small category $\mathcal{G}^*$ on $\mathcal{G}$ the objects of which are the nodes of $\mathcal{G}$, the identities of which are the empty paths and the morphisms of which are the finite paths. We call $\mathcal{G}^*$ the **category of paths of** $\mathcal{G}$.[1]

We define the **reverse** $\overline{e}$ of an edge e by $\overline{e} := a^-$ if $e = a^+$, and $\overline{e} := a^+$ if $e = a^-$. Then, for any *finite* path γ in $\mathcal{G}$ with edges $(e_i)_{1\leq i\leq N}$, we define the reverse path $\overline{\gamma} := (\mathsf{t}(\gamma), (\overline{e_{N+1-i}})_{1\leq i\leq N})$ so that $\mathsf{s}(\overline{\gamma}) = \mathsf{t}(\gamma)$ and $\mathsf{t}(\overline{\gamma}) = \mathsf{s}(\gamma)$.

[1] Strictly speaking, $\mathcal{G}^*$ is not the category freely generated by $\mathcal{G}$, which is rather the category of *directed* finite paths. One can consider $\mathcal{G}^*$ as the free category generated by the symmetric closure of $\mathcal{G}$, whose arrows are the edges of $\mathcal{G}$.

The reverse operation is compatible with composition in the sense that:

$$\overline{\epsilon_n} = \epsilon_n \text{ and,}$$
$$\overline{\gamma_0\gamma_1} = \overline{\gamma_1}\,\overline{\gamma_0}$$

for any finite paths γ_0 and γ_1. The category $\mathcal{G}^*$ is thus an **involutive category**.

We say a path γ **crosses** an arrow a if either a^+ or a^- occurs as an edge of γ. And we say γ **crosses** an edge e if e or $\overline{e}$ is a subpath of γ. A path γ is **simple** if it does not cross the same arrow (or, equivalently, the same edge) twice. We say two paths γ_1 and γ_2 are (arrow-)**disjoint** if they have no crossed arrow in common. If $\mathsf{t}(\gamma_1) = \mathsf{s}(\gamma_2)$ then the composition $\gamma_1\gamma_2$ is simple iff γ_1 and γ_2 are both simple and are disjoint. A **cyclic path**, also called a **cycle**, is a non-empty simple path γ that is closed. A graph is said to be **acyclic** if it has no cycle.

A path γ is **elementary** if no node occurs twice in γ, except maybe as its source and target: in that last case, we say γ is an **elementary cycle**. Note that an elementary path is always simple, and that an elementary cycle is in particular a cycle (it is indeed non-empty because, because the empty path has only one node occurrence).

Proposition A.2.1. *A graph is acyclic iff it has no elementary cycle.*

Proof. Given a cycle γ, consider any cyclic subpath γ' of γ, of minimum length: γ' must be elementary. □

We say two elementary paths γ_1 and γ_2 are **elementarily composable** if they are composable and $\gamma_1\gamma_2$ is elementary, *i.e.*:

- $\mathsf{t}(\gamma_1) = \mathsf{s}(\gamma_2)$ and this node has no other occurrence in γ_1 nor in γ_2;
- γ_1 and γ_2 share no internal node;
- $\mathsf{s}(\gamma_1)$ does not occur in γ_2, except maybe as its target;
- $\mathsf{t}(\gamma_2)$ does not occur in γ_1, except maybe as its source.

We write $n \simeq_{\mathcal{G}} n'$ if there exists a path from n to n'.

Proposition A.2.2. *There exists a path from n to n' iff there exists a simple (resp. elementary) path from n to n'*

Proof. Given a path γ, consider any subpath γ' of γ with the same endpoints, of minimum length: γ' must be elementary. □

We obtain that $\simeq_{\mathcal{G}}$ is an equivalence relation, and say n and n' are **connected** in $\mathcal{G}$, if $n \simeq_{\mathcal{G}} n'$. A **connected component** of $\mathcal{G}$ is an equivalence class for $\simeq_{\mathcal{G}}$. A graph is **connected** if it is not empty and any two nodes are connected by a path: in other words, it has exactly one connected component.

Lemma A.2.3 (Acyclic Connected Components)**.** *In a finite* acyclic graph*, the number of* connected components *is the number of nodes minus the number of arrows.*

Proof. By induction on the number of nodes.

- The empty graph has no node, no arrow and no connected component.
- Assume the graph contains at least one node. Let n be a node, if it has p arrows attached to it, we remove the node and all these arrows, we lose one node, p arrows and we create $p-1$ connected components (we cannot create more than $p-1$ connected components, and if we create strictly less than $p-1$ connected components, there was a cycle in the graph). We can then apply the induction hypothesis.

□

Lemma A.2.4 (Acyclicity and Connectedness)**.** *A graph with k arrows and $k+1$ nodes is* acyclic *if and only if it is* connected*.*

Proof. If the graph is acyclic, we apply Lemma A.2.3. If the graph is connected, we go by induction on the number of nodes:

- If there is 1 node, there is no arrow and the graph is acyclic.
- If there are at least $k \geq 2$ nodes, there are $k-1$ arrows. By connectedness each node has at least one arrow attached to it. Each arrow touches at most two nodes thus there must be a node n which is an endpoint of only one arrow a. We erase n and a, and we apply the induction hypothesis.

□

A path γ is **directed** if it contains no reversed arrow. A **directed cycle** is then a cycle that is also a directed path. A **directed acyclic graph** is a graph with no directed cycle. Setting $n \preccurlyeq_{\mathcal{G}} n'$ if there exists a directed path in $\mathcal{G}$ from n to n', we obtain that $\preccurlyeq_{\mathcal{G}}$ is a preorder relation. Moreover, this preorder is an order iff $\mathcal{G}$ is directed acyclic.

Examples: path, length, cycle, connected component **TODO**

Appendix B

Abstract Reduction Systems

We present some basic results about rewriting theory in the setting of abstract reduction systems. The material presented here is strongly inspired from [44].

B.1 Definitions and Notations

An **abstract reduction system** (ARS) $\mathcal{A}$ is a pair $(A, \to)$ where A is a set and $\to$ is a binary relation on A (*i.e.* a subset of $A \times A$).

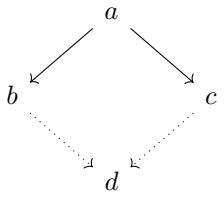
Given an ARS $\mathcal{A} = (A, \to)$, we use the following notations:

- $a \to b$ if $(a, b) \in \to$. b is called a **1-step reduct** of a.
- $a \leftarrow b$ if $b \to a$.
- $a \to^{=} b$ if $a = b$ or $a \to b$ ($\to^{=}$ is the reflexive closure of $\to$).
- $a \to^{+} b$ if there exists a finite sequence $(a_k)_{0 \leq k \leq N}$ ($N \geq 1$) of elements of A such that $a = a_0$, $a_N = b$ and for $0 \leq k \leq N-1$, $a_k \to a_{k+1}$ ($\to^{+}$ is the transitive closure of $\to$).
- $a \to^{*} b$ if $a = b$ or $a \to^{+} b$ ($\to^{*}$ is the reflexive transitive closure of $\to$). b is called a *reduct* of a.
- $a \simeq b$ if there exists a finite sequence $(a_k)_{0 \leq k \leq N}$ ($N \geq 0$) of elements of A such that $a = a_0$, $a_N = b$ and for $0 \leq k \leq N-1$, $a_k \to a_{k+1}$ or $a_k \leftarrow a_{k+1}$ ($\simeq$ is the reflexive symmetric transitive closure of $\to$).
- If a is an element of A, the **restriction** of $\mathcal{A}$ to a is the ARS $\mathcal{A}\restriction_a = (A\restriction_a, \to \cap (A\restriction_a \times A\restriction_a))$ where $A\restriction_a = \{b \in A \mid a \to^{*} b\}$ (*i.e.* the set of all reducts of a).

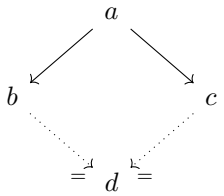
A sequence $(a_k)_{0 \leq k < N}$ (with $N \in \mathbf{N}$ such that $N \geq 1$, or $N = \infty$) of elements of A, such that $a_{k-1} \to a_k$ for each $0 < k < N$, is called a **reduction sequence** (starting from a_0 and ending on a_{N-1}, if $N \neq \infty$). When $N \in \mathbf{N}$, the reduction sequence is **finite** and its length is $N - 1$. We use the notation $a \to^{k} b$ if there exists a finite reduction sequence of length k starting from a and ending on b.

B.2 Confluence

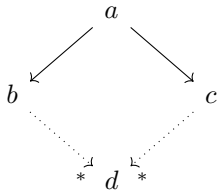
An ARS $(A, \to)$ has the **diamond property** if for any a, b and c in A with $a \to b$ and $a \to c$, there exists some d in A such that both $b \to d$ and $c \to d$. Thus diagrammatically:



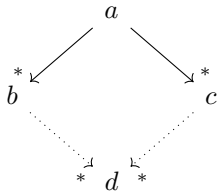
An ARS $(A, \to)$ is **sub-confluent** if for any a, b and c in A with $a \to b$ and $a \to c$, there exists some d in A such that both $b \to^{=} d$ and $c \to^{=} d$. Thus diagrammatically:



An ARS $(A, \to)$ is **locally confluent** if for any a, b and c in A with $a \to b$ and $a \to c$, there exists some d in A such that both $b \to^{*} d$ and $c \to^{*} d$. Thus diagrammatically:



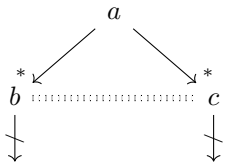
An ARS $(A, \to)$ is **confluent** if for any a, b and c in A with $a \to^{*} b$ and $a \to^{*} c$, there exists some d in A such that both $b \to^{*} d$ and $c \to^{*} d$. Thus diagrammatically:



An ARS $(A, \to)$ is thus confluent if $(A, \to^{*})$ has the diamond property.

A **normal form** in an ARS $(A, \to)$ is an element a of A such that there is no b in A with $a \to b$ (*i.e.* a has no reduct, but itself). A **$\to$-minimal** element is a normal form of $(A, \leftarrow)$.

An ARS $(A, \to)$ has the (weak) **unique normal form property** if for any a in A and any two normal forms b and c in A with $a \to^* b$ and $a \to^* c$, we have $b = c$. Thus diagrammatically:

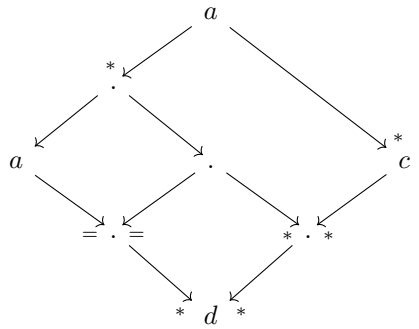


Proposition B.2.1 (Confluence Properties)**.** *For any ARS,*

- diamond property $\Longrightarrow$ sub-confluent $\Longrightarrow$ confluent $\Longrightarrow$ locally confluent*,*
- confluent $\Longrightarrow$ unique normal form*.*

Proof. We prove the four implications:

- If $a \to b$ and $a \to c$, the diamond property gives some d such that $b \to d$ and $c \to d$, thus $b \to^= d$ and $c \to^= d$.
- By induction on the length of the reduction sequence from a to b. The following figure might help.



  - If $a = b$, we have $b \to^* c$ and $c \to^* c$.
  - If $a \to b$, we use an induction on the length of the reduction sequence from a to c:
    - If $a = c$, we have $b \to^* b$ and $c \to^* b$.
    - If $a \to c$, by sub-confluence, there exists d such that $b \to^= d$ and $c \to^= d$.
    - If $a \to c'$ and $c' \to^* c$, by sub-confluence, we have some d' such that $b \to^= d'$ and $c' \to^= d'$. If $c' = d'$ we have $b \to^* c$ and $c \to^* c$. If $c' \to d'$, by induction hypothesis, there exists d such that $d' \to^* d$ and $c \to^* d$ (thus $b \to^* d$).

  - If $a \to^* a'$ and $a' \to b$, by induction hypothesis we have d' such that $a' \to^* d'$ and $c \to^* d'$. By the case above, there exists d such that $b \to^* d$ and $d' \to^* d$. We then conclude with $c \to^* d$.

- If $a \to b$ and $a \to c$, confluence gives some d such that $b \to^* d$ and $c \to^* d$.
- If $a \to^* b$ and $a \to^* c$ with b and c normal forms, confluence gives some d such that $b \to^* d$ and $c \to^* d$. But since b and c are normal forms, we must have $b = d$ and $c = d$.

□

B.3 Normalization

An ARS $(A, \to)$ is **weakly normalizing** if for any a in A there exists a normal form b in A such that $a \to^* b$ (b is a reduct of a).

An ARS $(A, \to)$ is **well founded** if every non-empty subset of A contains a $\to$-*minimal* element.

Lemma B.3.1 (Well Foundedness)**.** *An ARS $(A, \to)$ is* well founded *if and only if it satisfies the following induction principle:*

$$\forall P \ \ (\forall b\,((\forall a\,(a \to b) \Rightarrow Pa) \Rightarrow Pb)) \Rightarrow \forall b\,Pb$$

Proof. In the first direction, given a predicate P such that $\forall b\,((\forall a\,(a \to b) \Rightarrow Pa) \Rightarrow Pb)$, we define B to be $\{a \in A \mid \neg Pa\}$. If B is empty we are done: P is valid for all the elements of A. Otherwise, by well foundedness, B has a $\to$-minimal element b. The hypothesis on P thus gives us Pb which contradicts the fact that $b \in B$.

In the second direction, given a subset B of A with no $\to$-minimal element, we show that B is empty. We define the predicate Px as $x \notin B$: to prove that B is empty, by induction it sufficient to prove $\forall b\,((\forall a\,(a \to b) \Rightarrow Pa) \Rightarrow Pb))$. Let b be such that $\forall a\,(a \to b) \Rightarrow Pa$, that is $\forall a(a \to b) \Rightarrow (a \notin B)$. As a consequence, if $b \in B$, it is $\to$-minimal in B, a contradiction. Hence $b \notin B$, which establishes the induction. □

An ARS $(A, \to)$ is **strongly normalizing** if $(A, \leftarrow)$ is *well founded*. That is any non-empty subset B of A contains an element with no 1-step reduct in B.

An ARS is **convergent** if it is both confluent and strongly normalizing.

Lemma B.3.2 (Descending Chain Condition)**.** *A* strongly normalizing *ARS $(A, \to)$ does not contain any infinite reduction sequence.*

Proof. Let $(a_k)_{0 \leq k < \infty}$ be an infinite reduction sequence, we define $B = \{a \in A \mid \exists k \in \mathbf{N}, a = a_k\}$. B is not empty since $a_0 \in B$, thus it contains an element b with no 1-step reduct in B. There exists some k such that $b = a_k$ and thus $b \to a_{k+1}$, a contradiction. □

The converse property is a consequence of the Axiom of Dependent Choices.

Lemma B.3.3 (Transitive Strong Normalization)**.** *If* $(A, \to)$ *is* strongly normalizing *then* $(A, \to^+)$ *is strongly normalizing.*

Proof. Let B be an non-empty subset of A, we define $B' = \{a \in A \mid \exists b \in B, a \to^* b\}$. B' is not empty ($B \subseteq B'$) thus B' contains an element b with no 1-step reduct for $\to$ in B'. Since $b \to^* c$ with $c \in B$ implies $b = c$ (if $b \to b' \to^* c$ then b' is in B' and is a 1-step reduct of b), we have $b \in B$ and b has no 1-step reduct for $\to^+$ in B. □

An ARS $(A, \to)$ is **μ-decreasing** if μ is a function from A to a set with a well founded relation $<$ such that whenever $a \to b$, we have $\mu(a) > \mu(b)$.

An ARS $(A, \to)$ is **weakly μ-decreasing** if μ is a function from A to a set with a well founded relation $<$ such that, for any a in A which is not a normal form, there exists some b in A such that $a \to b$ and $\mu(a) > \mu(b)$.

An ARS $(A, \to)$ is **μ-increasing** if μ is a function from A to $\mathbf{N}$ such that whenever $a \to b$, we have $\mu(a) < \mu(b)$.

Proposition B.3.4 (Normalization Properties)**.** *For any ARS,* μ-decreasing *for some* $\mu \implies$ strongly normalizing $\implies$ weakly μ-decreasing *for some* $\mu \implies$ weakly normalizing.

Proof. Let $\mathcal{A} = (A, \to)$ be an ARS.

- Let B be a non-empty subset of A and E be its image by μ. E is a non-empty set and $<$ is a well founded relation thus E has a $<$-minimal element e. Let b be such that $\mu(b) = e$, b has no 1-step reduct in B otherwise we would have $b \to c$ and thus $e = \mu(b) > \mu(c)$ with $\mu(c) \in E$ contradicting the $<$-minimality of e in E.

- Since $\mathcal{A}$ is strongly normalizing, it is *id*-decreasing where *id* is the identity function. If a is not a normal form, let b be any 1-step reduct of a, we have $id(a) \to id(b)$.

- Given an a in A, $\mu(A \restriction_a)$ is a non-empty set ($\mu(a) \in \mu(A \restriction_a)$) and $<$ is a well founded relation thus $\mu(A \restriction_a)$ has a $<$-minimal element e. Let c be such that $\mu(c) = e$, $a \to^* c$ since $c \in A \restriction_a$, and c is a normal form. Otherwise there exists d such that $c \to d$ and $e = \mu(c) > \mu(d)$ contradicting the $<$-minimality of e in $\mu(A \restriction_a)$.

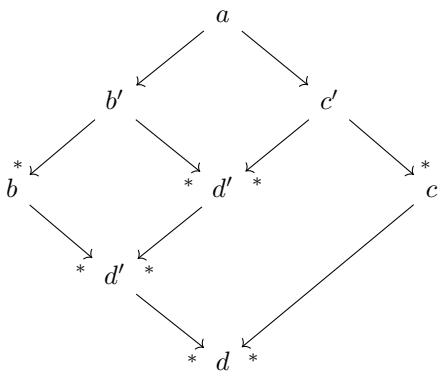
□

Proposition B.3.5 (Weak Normalization and Confluence)**.** *For any ARS,* weakly normalizing $\wedge$ unique normal form $\implies$ confluent.

Proof. If $a \to^* b$ and $a \to^* c$, by weak normalization, there exist two normal forms b' and c' such that $b \to^* b'$ and $c \to^* c'$, thus $a \to^* b'$ and $a \to^* c'$. By uniqueness of the normal form, we have $b' = c'$. □

Proposition B.3.6 (Newman's Lemma)**.** *For any ARS,* strongly normalizing $\wedge$ locally confluent $\implies$ confluent.

Proof. Let $\mathcal{A} = (A, \to)$ be a strongly normalizing and locally confluent ARS, since the relation $\leftarrow$ is well founded, we can reason by induction on it (Lemma B.3.1). We prove this way that for any a, $\mathcal{A} \restriction_a$ is confluent. We assume that for any 1-step reduct a' of a, $\mathcal{A} \restriction_{a'}$ is confluent. Assume $a \to^* b$ and $a \to^* c$. If $a = b$ or $a = c$ the result is immediate. If $a \to b' \to^* b$ and $a \to c' \to^* c$, by local confluence, we have d' such that $b' \to^* d'$ and $c' \to^* d'$. By confluence of $\mathcal{A} \restriction_{b'}$, there exists d'' such that $b \to^* d''$ and $d' \to^* d''$, thus $c' \to^* d''$. By confluence of $\mathcal{A} \restriction_{c'}$, there exists d such that $d'' \to^* d$ and $c \to^* d$, thus $b \to^* d$ and we conclude.



□

Proposition B.3.7 (Increasing Normalization)**.** *For any ARS and any μ,* locally confluent $\wedge$ μ-increasing $\wedge$ weakly normalizing $\Longrightarrow$ strongly normalizing.

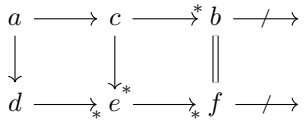
Proof. Let $\mathcal{A} = (A, \to)$ be an ARS, we first prove by induction on k that $a \to^* b$ with b normal form and $\mu(b) - \mu(a) \leq k$ implies $\mathcal{A} \restriction_a$ is strongly normalizing.

- If $k = 0$, a is a normal form, $A \restriction_a = \{a\}$ and the result is immediate.

- If $k > 0$, we can decompose the reduction sequence from a to b into $a \to c \to^* b$. We have $\mu(c) > \mu(a)$ thus $\mu(b) - \mu(c) < k$ with $c \to^* b$ and, by induction hypothesis, $\mathcal{A} \restriction_c$ is strongly normalizing. By Propositions B.3.6 and B.2.1, $\mathcal{A} \restriction_c$ also has the unique normal form property.

  Let d be an arbitrary 1-step reduct of a, by local confluence, there exists some e such that both $c \to^* e$ and $d \to^* e$. By weak normalization, let f be a normal from of e, we necessarily have $f = b$ (unique normal form of c) thus $d \to^* b$, $\mu(b) - \mu(d) < k$ (since $\mu(d) > \mu(a)$) and, by induction

hypothesis, $\mathcal{A}\restriction_d$ is strongly normalizing.



Now let B be an non-empty subset of $\mathcal{A}\restriction_a$. If a is in B and has no 1-step reduct in B, we are done. Otherwise, we have $a \to d \to^* b$ for some d and some $b \in B$. We have proved that $\mathcal{A}\restriction_d$ is strongly normalizing. We define $B' = B \cap \mathcal{A}\restriction_d$. B' is a non-empty set since it contains b and thus it has an element c with no 1-step reduct in B'. c also belongs to B and has no 1-step reduct in B by construction (c is a reduct of d so any reduct of c is a reduct of d as well).

Given a non-empty subset B of A, let a be an element of B, by weak normalization, a has a normal form b thus $\mathcal{A}\restriction_a$ is strongly normalizing. By defining $B' = B \cap \mathcal{A}\restriction_a$, we prove just as above that B contains an element with no 1-step reduct in B, showing that $\mathcal{A}$ is strongly normalizing. □

B.4 Simulation

Let $\mathcal{A} = (A, \to_A)$ and $\mathcal{B} = (B, \to_B)$ be two ARSs, a function φ from A to B is a **simulation** if for every a and a' in A, $a \to_A a'$ entails $\varphi(a) \to_B^* \varphi(a')$. It is a **strict simulation** if $a \to_A a'$ entails $\varphi(a) \to_B^+ \varphi(a')$.

Proposition B.4.1 (Anti Simulation of Strong Normalization)**.** *If φ is a* strict simulation *from $\mathcal{A}$ to $\mathcal{B}$ and $\mathcal{B}$ is strongly normalizing, then $\mathcal{A}$ is* strongly normalizing *as well.*

Proof. By Lemma B.3.3, $(B, \to_B^+)$ is strongly normalizing, thus $\leftarrow^+$ is a well founded relation. We can conclude with Proposition B.3.4 since $\mathcal{A}$ is then φ-decreasing. □

Proposition B.4.2 (Anti Simulation of Unique Normal Form)**.** *If φ is a* simulation *from $\mathcal{A}$ to $\mathcal{B}$ which preserves normal forms (*i.e. *if a is a normal form in $\mathcal{A}$ then $\varphi(a)$ is a normal form in $\mathcal{B}$) and is injective on normal forms (*i.e. *no two different normal forms of $\mathcal{A}$ have the same image through φ), then the unique normal form property for $\mathcal{B}$ entails the* unique normal form property *for $\mathcal{A}$.*

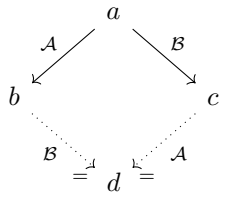
Proof. Assume b and c are normal forms with $a \to_A^* b$ and $a \to_A^* c$, then $\varphi(a) \to_B^* \varphi(b)$ and $\varphi(a) \to_B^* \varphi(c)$ with $\varphi(b)$ and $\varphi(c)$ normal forms. This entails $\varphi(b) = \varphi(c)$ by unique normal form for $\mathcal{B}$, and finally $b = c$ since φ is injective on normal forms. □

B.5 Commutation

In this section, we consider two ARSs $\mathcal{A} = (A, \to_{\mathcal{A}})$ and $\mathcal{B} = (A, \to_{\mathcal{B}})$ on the same set A.

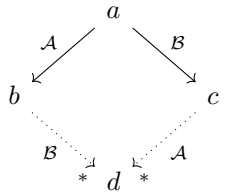
The ARS $\mathcal{A} \bowtie \mathcal{B}$ is defined as $(A, \to_{\mathcal{A}\bowtie\mathcal{B}})$ with $\to_{\mathcal{A}\bowtie\mathcal{B}} = \to_{\mathcal{A}} \cup \to_{\mathcal{B}}$. Note that $\to^*_{\mathcal{A}\bowtie\mathcal{B}} = (\to^*_{\mathcal{A}} \cup \to^*_{\mathcal{B}})^*$

We say that $\mathcal{A}$ and $\mathcal{B}$ **sub-commute** if for any a, b and c in A such that $a \to_{\mathcal{A}} b$ and $a \to_{\mathcal{B}} c$, there exists d such that $b \to^{=}_{\mathcal{B}} d$ and $c \to^{=}_{\mathcal{A}} d$. Thus diagrammatically:



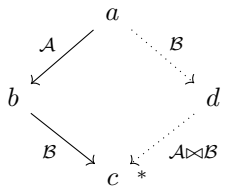
With this definition, an ARS is *sub-confluent* if it *sub-commutes* with itself.

We say that $\mathcal{A}$ and $\mathcal{B}$ **locally commute** if for any a, b and c in A such that $a \to_{\mathcal{A}} b$ and $a \to_{\mathcal{B}} c$, there exists d such that $b \to^*_{\mathcal{B}} d$ and $c \to^*_{\mathcal{A}} d$. Thus diagrammatically:



With this definition, an ARS is *locally confluent* if it *locally commutes* with itself.

We say that $\mathcal{A}$ **quasi-commutes over** $\mathcal{B}$, if for any a, b and c in A such that $a \to_{\mathcal{A}} b$ and $b \to_{\mathcal{B}} c$, there exists d such that $a \to_{\mathcal{B}} d$ and $d \to^*_{\mathcal{A}\bowtie\mathcal{B}} c$. Thus diagrammatically:



Proposition B.5.1 (Commutation of Strong Normalization)**.** *If $\mathcal{A} = (A, \to_{\mathcal{A}})$ and $\mathcal{B} = (A, \to_{\mathcal{B}})$ are two ARSs, and* $\mathcal{A}$ quasi-commutes over $\mathcal{B}$ *then if $\mathcal{A}$ and $\mathcal{B}$ are strongly normalizing then $\mathcal{A} \bowtie \mathcal{B}$ is* strongly normalizing*.*

Proof. Let B_0 be a non-empty subset of A, we define $B = \{a \in A \mid \exists b \in B_0, a \to^*_{\mathcal{A}\bowtie\mathcal{B}} b\}$ which is non-empty as well ($B_0 \subseteq B$) and is such that $a \to^*_{\mathcal{A}\bowtie\mathcal{B}} b$ with $b \in B$ entails $a \in B$. By strong normalization of $\mathcal{B}$, the subset B' of B containing the elements of B with no 1-step $\to_{\mathcal{B}}$-reduct in B is not empty. By

strong normalization of $\mathcal{A}$, B' contains an element a with no 1-step $\rightarrow_{\mathcal{A}}$-reduct in B'. If a has no 1-step $\rightarrow_{\mathcal{A}}$-reduct in B, we have an element with no 1-step $\rightarrow_{\mathcal{A}\triangleright\triangleleft\mathcal{B}}$-reduct in B. Otherwise $a \rightarrow_{\mathcal{A}} b$ for some b which is in B and not in B' thus there exists $c \in B$ such that $b \rightarrow_{\mathcal{B}} c$. By quasi-commutation, we have d such that $a \rightarrow_{\mathcal{B}} d \rightarrow^*_{\mathcal{A}\triangleright\triangleleft\mathcal{B}} c$. We have $d \in B$ since $d \rightarrow^*_{\mathcal{A}\triangleright\triangleleft\mathcal{B}} c$ but this contradicts the fact that $a \in B'$.

This means a cannot have a 1-step $\rightarrow_{\mathcal{A}\triangleright\triangleleft\mathcal{B}}$-reduct in B, so that a belongs to B_0 and has no 1-step $\rightarrow_{\mathcal{A}\triangleright\triangleleft\mathcal{B}}$-reduct in B_0. □

Appendix C

Basic concepts of category theory

C.1 Categories, functors and natural transformations

A category $\mathcal{C}$ consists of:

- a class of objects $\mathsf{Obj}(\mathcal{C})$
- for each $X, Y \in \mathsf{Obj}(\mathcal{C})$, of a class of morphisms $\mathcal{C}(X, Y)$ from X to Y,
- for each $X \in \mathsf{Obj}(\mathcal{C})$, of a special element Id_X of $\mathcal{C}(X, X)$ called identity at X
- and, for each triple $(X, Y, Z) \in \mathcal{C}^3$, of a composition operation

$$\begin{array}{rcl} \circ : \mathcal{C}(X,Y) \times \mathcal{C}(Y,Z) & \to & \mathcal{C}(X,Z) \\ (f,g) & \mapsto & g \circ f \end{array}$$

such that the following equations hold (for $f \in \mathcal{C}(X, Y)$, $g \in \mathcal{C}(Y, Z)$ and $h \in \mathcal{C}(Z, V)$):

$$f \circ \mathsf{Id}_X = f \qquad \mathsf{Id}_Y \circ f = f \qquad h \circ (g \circ f) = (h \circ g) \circ f$$

We often denote composition as simple juxtaposition and Id_X as X.

Example C.1.1. The category **Set** has sets as objects and functions as morphisms. It underlies most categories whose objects are sets endowed with a structure and morphisms are functions "preserving" this structure in some sense, for instance:

- monoids and homomorphisms of monoids

- groups and homomorphisms of groups
- given a field, vector spaces on this field and linear functions
- topological spaces and continuous functions.

Example C.1.2. The category **Rel** is less usual but very important for us. Its objects are sets but now $\mathbf{Rel}(X,Y) = \mathcal{P}(X \times Y)$, whose elements are seen as relations from X to Y, Id_X is the diagonal relation $\mathsf{Id}_X = \{(a,a) \mid a \in X\}$ and composition is the ordinary composition of relations: given $s \in \mathbf{Rel}(X,Y)$ and $t \in \mathbf{Rel}(Y,Z)$, then

$$t \circ s = \{(a,c) \mid \exists b \in Y \ (a,b) \in s \text{ et } (b,c) \in t\}\,.$$

We denote this composition by simple juxtaposition $t\,s$ as a product, and Id_X as X. An example of categories built in that way is the category whose objects are finite sets and a morphism from I to J is an $I \times J$ matrix with coefficients in some (semi-)ring, composition being defined as the usual product of matrices.

We should think of **Rel** as of an (over)simplification of the categories of vector spaces – or more accurately, vector spaces given with a choice of basis – and linear maps seen as matrices. In this category we can see an element of $\mathbf{Rel}(X,Y)$ as a linear map from the free module generated by X to the free module generated by Y over the semi-ring of coefficients $\{0,1\}$ with $1+1=1$. This model has the virtue of featuring a concrete notion of linearity (composition of relations commutes with their unions) which illustrate in a very simple way the kind of linearity that Linear Logic axiomatizes logically.

An isomorphism is a morphism $f \in \mathcal{C}(X,Y)$ such that there is a morphism $g \in \mathcal{C}(Y,X)$ such that $g \circ f = \mathsf{Id}_X$ and $f \circ g = \mathsf{Id}_Y$. If g and g' satisfy these conditions then $g = g \circ \mathsf{Id}_Y = g \circ (f \circ g') = (g \circ f) \circ g' = \mathsf{Id}_X \circ g' = g'$ by the equations above and hence g is fully determined by f and is denoted as f^{-1}.

The opposite category of $\mathcal{C}$ is the category $\mathcal{C}^{\mathsf{op}}$ given by $\mathsf{Obj}(\mathcal{C}^{\mathsf{op}}) = \mathsf{Obj}(\mathcal{C})$ and $\mathcal{C}^{\mathsf{op}}(X,Y) = \mathcal{C}(Y,X)$. The identities are the same and composition is defined in the obvious way (reversing the order of factors).

The product $\prod_{i\in I} \mathcal{C}_i$ of a family of categories $(\mathcal{C}_i)_{i\in I}$ has the families $\overrightarrow{X} = (X_i \in \mathsf{Obj}(\mathcal{I}))_{i\in I}$ and an element of $\prod_{i\in I} \mathcal{C}_i(\overrightarrow{X},\overrightarrow{Y})$ is a family $(f_i \in \mathcal{C}_i(X_i,Y_i))$. Identities and composition are defined in the obvious componentwise manner.

C.1.1 Functors

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A functor F from $\mathcal{C}$ to $\mathcal{D}$ is an operation which

- maps any object X of $\mathcal{C}$ to an object $F(X)$ of $\mathcal{D}$
- and any morphism $f \in \mathcal{C}(X,Y)$ to a morphism $F(f) \in \mathcal{C}(F(X),F(Y)$

such that, for any $X,Y,Z \in \mathsf{Obj}(\mathcal{C})$ and $f \in \mathcal{C}(X,Y)$ and $g \in \mathcal{C}(Y,Z)$:

$$F(\mathsf{Id}_X) = \mathsf{Id}_{F(X)} \qquad F(g \circ f) = F(g) \circ F(f)\,.$$

Notice that F induces trivially a morphisms $\mathcal{C}^{\mathsf{op}} \to \mathcal{D}^{\mathsf{op}}$ defined exactly as F on object and morphisms, this functor is also denoted F.

A contravariant functor from $\mathcal{C}$ to $\mathcal{D}$ is a functor from $\mathcal{C}^{\mathsf{op}}$ to $\mathcal{D}$ (or, equivalently, from $\mathcal{C}$ to $\mathcal{D}^{\mathsf{op}}$).

A functor $F : \mathcal{C} \to \mathcal{D}$ is *full* if, for any $X, Y \in \mathsf{Obj}(\mathcal{C})$, the function $\mathcal{C}(X,Y) \to \mathcal{D}(F(X), F(Y))$ which maps f to $F(f)$ is surjective. It is *faithful* if this function is injective.

For instance, the functor P from **Rel** to **Set** which maps a set X to $\mathcal{P}(X)$ and a relation $s \in \mathbf{Rel}(X,Y)$ to the function $P(s) : \mathcal{P}(X) \to \mathcal{P}(Y)$ given by $P(s)(u) = \{b \in Y \mid \exists a \in u \ (a,b) \in s\}$ is a functor from **Rel** to **Set**. This functor is faithful but not full.

For any category $\mathcal{C}$, there is a functor $\mathsf{Hom}_{\mathcal{C}} : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$ defined on objects by $\mathsf{Hom}_{\mathcal{C}}(X,Y) = \mathcal{C}(X,Y)$ and on morphisms by

$$\begin{aligned} \mathsf{Hom}_{\mathcal{C}}(f,g) : \mathcal{C}(X,Y) &\to \mathcal{C}(X',Y') \\ h &\mapsto g \circ h \circ f \end{aligned}$$

for $g \in \mathcal{D}(Y,Y')$ and $f \in \mathcal{C}(X',X)$.

C.1.2 Natural transformations

Let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A natural transformation from F to G is a family $T = (T_X)_{X \in \mathsf{Obj}(\mathcal{C})}$ of morphisms such that, for each $X \in \mathsf{Obj}(\mathcal{C})$ one has $T_X \in \mathcal{D}(F(X), G(X))$ and such that, for each $f \in \mathcal{C}(X,Y)$, one has $G(f) \circ T_X = T_Y \circ F(f)$. This is expressed by saying that the following diagram commutes:

$$\begin{array}{ccc} F(X) & \xrightarrow{T_X} & G(X) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(Y) & \xrightarrow{T_Y} & G(Y) \end{array}$$

this means that the composition of morphisms on both sides coincide. One writes $S : F \overset{\bullet}{\to} G$. Let $F, G, H : \mathcal{C} \to \mathcal{D}$ be three functors. If $S : F \overset{\bullet}{\to} G$ and $T : G \overset{\bullet}{\to} H$, one defines $T \circ S : F \overset{\bullet}{\to} H$ par $(T \circ S)_X = T_X \circ S_X$. In that way one defines the category $\mathcal{D}^{\mathcal{C}}$ of functors and natural transformations. This composition is often called the *horizontal composition* of natural transformations.

Exercise C.1.3. Let $F, F' : \mathcal{C} \to \mathcal{D}$ and $G, G' : \mathcal{D} \to \mathcal{E}$ be functors. Let $S : F \overset{\bullet}{\to} F'$ and $T : G \overset{\bullet}{\to} G'$ be natural transformations. Let $X \in \mathsf{Obj}(\mathcal{C})$. Prove that $G'(S_X) \circ T_{F(X)} = T_{F'(X)} \circ G(S_X)$. One denote as $(T * S)_X \in \mathcal{E}(G(F(X)), G'(F'(X)))$ the morphism so defined. Prove that $T * S$ is a natural transformation $G \circ F \overset{\bullet}{\to} G' \circ F'$. Prove that this operation is associative and give its neutral element. It is called *vertical composition* of natural transformations. Let $F'' : \mathcal{C} \to \mathcal{D}$ and $G'' : \mathcal{D} \to \mathcal{E}$ be two other functors and $S' : F' \overset{\bullet}{\to} F''$ and $T' : G' \overset{\bullet}{\to} G''$ be two other natural transformations. Prove that $(T' \circ T) * (S' \circ S) = (T' * S') \circ (T * S)$. This property is called

exchange law. The category of categories, with functors as morphisms and natural transformations as morphisms between morphisms, with these two laws of composition, is a 2-*category*.

C.2 Limits and colimits

C.2.1 Projective limits (limits)

C.2.1.1 Terminal objects.

An object T of a cetagory $\mathcal{C}$ is *terminal* if, for any object X of $\mathcal{C}$, the set $\mathcal{C}(X,T)$ has exactly one element. Let T and T' be terminal objects of $\mathcal{C}$. Let f be the unique element of $\mathcal{C}(T',T)$ and f' the unique element of $\mathcal{C}(T,T')$. Since $\mathcal{C}(T,T) = \{\mathsf{Id}_T\}$, we must have $f \circ f' = \mathsf{Id}_T$ and also $f' \circ f = \mathsf{Id}_{T'}$. In other words, there is exactly one morphism from T to T', and this morphism is an iso. It is a very strong way to say that, if a category has a terminal object, this object is unique up to unique iso.

Terminal objects are a very special case of projective limit as we shall see, but, choosing the suitable category, any projective limit can be seen as a terminal object (this seems to be a general pattern of category theory: any universal notion is more general than any other universal notion).

C.2.1.2 General limits

Let $\mathcal{C}$ be a category and I be a small category (that is, such that $\mathsf{Obj}(I)$ is a set). There is an obvious functor $\Delta : \mathcal{C} \to \mathcal{C}^I$ which maps an object X of $\mathcal{C}$ to the constant functor defined by $\Delta(X)(i) = X$ and $\Delta(X)(u) = \mathsf{Id}_X$. Let $D : I \to \mathcal{C}$ be a functor (such a "small" functor is sometimes called a *diagram*). A *projective cone* based on D is a pair (X,p) where $X \in \mathsf{Obj}(\mathcal{C})$ and $p : \Delta(X) \stackrel{\bullet}{\to} D$. In other words it consists of the following data: the object X, and, for any $i \in \mathsf{Obj}(I)$, a mophism $p_i \in \mathcal{C}(X, D(i))$ such that, for each $\varphi \in I(i,j)$, one has $D(\varphi) \circ p_i = f_j$.

Let (X,p) and (Y,q) be projective cones based on D. A cone morphism from (X,p) to (Y,q) is an $h \in \mathcal{C}(X,Y)$ such that, for each $i \in I$, one has $q_i \circ h = p_i$. In that way we define a category $\mathcal{C}_D$. A *limiting projective cone* on D is a terminal object of the category $\mathcal{C}_D$.

In other words, a limiting projective cone based on D is a projective cone (P,p) based on D such that, for any other cone (X,q) based on D, there is exactly one $h \in \mathcal{C}(P,X)$ such that $\forall i \in I\ p_i \circ h = q_i$. A limiting projective cone based on D is also simply called a (projective, or inverse) limit of D.

Proposition C.2.1. *Let $(Y,(q_i)_{i\in I})$ and $(Y',(q'_i)_{i\in I})$ be projective limits of the diagram D. Then there is exactly one morphism $g \in \mathcal{C}(Y,Y')$ such that $\forall i \in I\ q'_i \circ g = q_i$. Moreover, g is an iso.*

This is a rephrasing of the fact that a projective limit is a terminal object in the category of cones. Because of this strong uniqueness property one often uses the notation $\varprojlim D$ to denote this limit when it exists. Remember that, to

be fully specified, a limit must be given as an object P together with a family of morphisms $(p_i)_{i\in\mathsf{Obj}(I)}$ (the projective cone) which can be seen as some kind of "projections" from P to the objects of the diagram D, whence the adjective "projective".

Proposition C.2.2. *Assume that all diagrams $D \in \mathsf{Obj}(\mathcal{C}^I)$ have a projective limit $(\varprojlim D, (p_i^D)_{i\in\mathsf{Obj}(I)})$. Then there is exactly one functor $L : \mathcal{C}^I \to \mathcal{C}$ such that $L(D) = \varprojlim D$ and, for each $T \in \mathcal{C}^I(D, E)$, the following triangle commutes for each $i \in \mathsf{Obj}(I)$:*

$$\begin{array}{ccc} \varprojlim D & \xrightarrow{L(T)} & \varprojlim E \\ {\scriptstyle p_i^D}\downarrow & & \downarrow{\scriptstyle p_i^E} \\ D(i) & \xrightarrow{T_i} & E(i) \end{array}$$

Proof. Observe that $(\varprojlim D, (T_i \circ p_i^D)_{i\in\mathsf{Obj}(I)})$ is a projective cone on E and apply the universal property of the cone $(\varprojlim E, (p_i^E)_{i\in\mathsf{Obj}(I)})$. □

Here are a few examples of projective limits.

Example C.2.3. If I is a *discrete category*, that is a category whose only morphisms are the identities (and therefore can be considered as a bare set since it is small), then D is just an I-indexed family of objects of $\mathcal{C}$. In that case, when the projective limit $(P, (\mathsf{pr}_i)_{i\in I})$ of D exists, it is called the *cartesian product* of the family D and the morphisms $\mathsf{pr}_i \in \mathcal{C}(P, D_i)$ are called the *projections*. We will often use $\&_{i\in I} D_i$ to denote the object P. Special cases: if $I = \emptyset$, the projective limit consists simply of an object $\top$ characterized by the fact that, for any object X of $\mathcal{C}$, the set $\mathcal{C}(X, \top)$ is a singleton, whose unique element will be denoted ast_X. In other words, $\top$ is a terminal object of $\mathcal{C}$. A category is cartesian if all finite families of objects have a cartesian product.

Assume that $\mathcal{C}$ is cartesian. The operation $(X_1, X_2) \mapsto X_1 \,\&\, X_2$ can be turned into a functor $\mathcal{C}^2 \to \mathcal{C}$ by Proposition C.2.2: let $f_i \in \mathcal{C}(X_i, Y_i)$ for $i = 1, 2$. We have $f_i \circ \mathsf{pr}_i \in \mathcal{C}(X_1 \,\&\, X_2, X_i)$ and hence there is a unique morphism $f_1 \,\&\, f_2 \in \mathcal{C}(X_1 \,\&\, X_2, Y_i)$ such that $\mathsf{pr}_i \circ (f_1 \,\&\, f_2) = f_i \circ \mathsf{pr}_i$ for $i = 1, 2$ and the operation which maps (X_1, X_2) to $X_1 \,\&\, X_2$ and $(f_1, f_2) \in \mathcal{C}(X_1, Y_1) \times \mathcal{C}(X_2, Y_2)$ to $f_1 \,\&\, f_2$ is a functor.

Example C.2.4. Let I be the category such that $\mathsf{Obj}(I) = \{1, 2\}$ and $I(1, 2) = \{\alpha, \beta\}$. A diagram is given by two objects X and Y of $\mathcal{C}$ and two morphisms $f, g \in \mathcal{C}(X, Y)$. A projective limit of this diagram consists of an object E and a morphism $e \in \mathcal{C}(E, X)$ such that $f \circ e = g \circ e$ and, for any object Z of $\mathcal{C}$ and any morphism $h \in \mathcal{C}(Z, X)$ such that $f \circ h = g \circ h$, there is exactly one morphism $h_0 \in \mathcal{C}(Z, E)$ such that $h = e \circ h_0$. Such a limit is called an *equalizer* of f and g.

From now on, we drop the adjective "projective" and simply use the word *limit* and *cone* to refer to projective limits and cones.

C.2.1.3 Cartesian closed categories

Let $\mathcal{C}$ be a cartesian category. Let $X, Y \in \mathsf{Obj}(\mathcal{C})$. An *internal hom* from X to Y is a pair (E, e) where E is an object of $\mathcal{C}$ and $e \in \mathcal{C}(E \,\&\, X, Y)$ are such that, for any $Z \in \mathsf{Obj}(\mathcal{C})$ and for any $f \in \mathcal{C}(Z \,\&\, X, Y)$ there is a unique $f' \in \mathcal{C}(Z, E)$ such that $e \circ (f' \,\&\, \mathsf{Id}_X) = f$. Being given by a universal property, an internal hom is unique up to unique morphism which is an isomorphism.

More precisely let (E', e') be another internal hom from X to Y. Since $e' \in \mathcal{C}(E' \,\&\, X, Y)$ there is a unique $h' \in \mathcal{C}(E', E)$ such that $e \circ (h \,\&\, \mathsf{Id}_X) = e'$ and for the same reason there is a unique $h' \in \mathcal{C}(E, E')$ such that $e' \circ (h' \,\&\, \mathsf{Id}_X) = e$. Therefore we have $e \circ ((h \circ h') \,\&\, \mathsf{Id}_X) = e$, and since $e \circ (\mathsf{Id}_E \,\&\, \mathsf{Id}_X) = e$, we have $h \circ h' = \mathsf{Id}_E$ and for the same reason $h' \circ h = \mathsf{Id}_{E'}$, hence h is an iso with h' as inverse.

So we can introduce notations: this internal hom (or rather, a choice of internal hom) from X to Y will be denoted as $(X \Rightarrow Y, \mathsf{Ev}_{X,Y})$, $X \Rightarrow Y$ is called *internal hom object*, Ev is called the *evaluation map*, or application and if $f \in \mathcal{C}(Z \,\&\, X, Y)$, the unique morphism $h : \mathcal{C}(Z, X \Rightarrow Y)$ such that $\mathsf{Ev} \circ (h \,\&\, \mathsf{Id}_X)$ will be denoted as $\mathsf{Cur}(f)$ and called *curryfication* of f, in reference to Haskell Curry, father of the λ-calculus.

These constructions can be characterized by a system of three equations:

$$\begin{aligned} \mathsf{Ev} \circ (\mathsf{Cur}(f) \,\&\, \mathsf{Id}_X) &= f \\ \mathsf{Cur}(f) \circ g &= \mathsf{Cur}(f \circ (g \,\&\, \mathsf{Id}_X)) \quad \text{where } g \in \mathcal{C}(Z', Z) \\ \mathsf{Cur}(\mathsf{Ev}) &= \mathsf{Id}_{X \Rightarrow Y} \,. \end{aligned} \tag{C.1}$$

It can be easier to check these equations than directly the universal property.

Exercise C.2.5. Let $X, Y \in \mathsf{Obj}(\mathcal{C})$. Let $\mathcal{C}_{X,Y}$ be the following category: an object of $\mathcal{C}_{X,Y}$ is a pair (Z, f) where $Z \in \mathsf{Obj}(\mathcal{C})$ and $f \in \mathcal{C}(Z \,\&\, X, Y)$. The homset $\mathcal{C}_{X,Y}((Z, f), (Z', f'))$ is the set of all $g \in \mathcal{C}(Z, Z')$ such that $f' \circ (g \,\&\, \mathsf{Id}_X) = f$. Prove that we have defined a category in that way, and that an internal hom from X to Y is a terminal object in that category.

Exercise C.2.6. Prove that **Set** is cartesian closed and that **Rel** has all small products but is not cartesian closed.

C.2.1.4 Inductive limits (colimits)

The definition of colimits (or inductive limits) is obtained by reversing all arrows, in other words, a colimit in $\mathcal{C}$ is the same thing as a limit in $\mathcal{C}^{\mathsf{op}}$. Since these are very important concept, we spell out the corresponding definitions.

An *initial object* in $\mathcal{C}$ is an object Z of $\mathcal{C}$ such $\mathcal{C}(Z, X)$ is a singleton for any object X.

As above, I is a small category. Given a diagram $D \in \mathcal{C}^I$, a cocone based on D is a pair (X, e) where $X \in \mathsf{Obj}(\mathcal{C})$ and e is a natural transformation $e : D \xrightarrow{\bullet} \Delta(X)$. In other words, it is a pair $(X, (e_i)_{i \in \mathsf{Obj}(I)})$ where $e_i \in \mathcal{C}(D(i), X)$ for each $i \in \mathsf{Obj}(I)$ and, given $\varphi \in I(i, j)$, one has $\varphi \circ e_i = e_j$. A morphism

from the cocone $(X, (e_i)_{i\in\mathsf{Obj}(I)})$ to the cocone $(Y, (f_i)_{i\in\mathsf{Obj}(I)})$ based on D is a morphism $g \in \mathcal{C}(X, Y)$ such that $f_i \circ g = e_i$ for all $i \in \mathsf{Obj}(I)$.

A cocone $(X, (e_i)_{i\in\mathsf{Obj}(I)})$ is a *colimit cocone* if it is an initial object in the category of cocones, in other words: for any cocone $(Y, (f_i)_{i\in\mathsf{Obj}(I)})$ based on D there is a unique morphism $g \in \mathcal{C}(X, Y)$ such that $f_i \circ g = e_i$ for all $i \in \mathsf{Obj}(I)$.

When I is a discrete category (that is, a set), the colimit of a diagram on I, that is, of a family $(X_i)_{i\in I}$ of objects of $\mathcal{C}$, is an object $\oplus_{i\in I} X_i$ together with injection morphisms $(\mathsf{in}_i \in \mathcal{C}(X_i, \oplus_{j\in I} X_j))_{i\in I}$ such that, for any family of morphisms $(f_i \in \mathcal{C}(X_i, Y))_{i\in I}$ there is exactly one morphism $f = [f_i]_{i\in I} \in \mathcal{C}(\oplus_{i\in I} X_i, Y)$ such that $f \circ \mathsf{in}_i = f_i$ for all $i \in I$. Then $(\oplus_{i\in I} X_i, (\mathsf{in}_i)_{i\in I})$ is the *coproduct* of the X_i's.

When $I = \emptyset$, this coproduct (of the empty family of objects), is denoted as 0, which is the initial object of $\mathcal{C}$. We use z_X or simply z for the unique element $\mathcal{C}(0, X)$.

C.3 Adjunctions

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Let $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$ be functors. Observe that we have two functors

$$\mathsf{Hom}_{\mathcal{D}} \circ (L \times \mathsf{Id}_{\mathcal{D}}), \mathsf{Hom}_{\mathcal{C}} \circ (\mathsf{Id}_{\mathcal{C}} \times R) : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathbf{Set}$$

An adjuction between $\mathcal{C}$ and $\mathcal{D}$ is a triple (L, R, Φ) where $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$ are functors and $\Phi : \mathsf{Hom}_{\mathcal{D}} \circ (L \times \mathsf{Id}_{\mathcal{D}}) \overset{\bullet}{\to} \mathsf{Hom}_{\mathcal{C}} \circ (\mathsf{Id}_{\mathcal{C}} \times R)$ is a natural bijection. In other words, for any $A \in \mathsf{Obj}(\mathcal{C})$ and $X \in \mathsf{Obj}(\mathcal{D})$ we are given a bijection

$$\Phi_{A,X} : \mathcal{D}(L(A), X) \to \mathcal{C}(A, R(X))$$

such that, if $\varphi \in \mathcal{C}(A', A)$ and $f \in \mathcal{D}(X, X')$, one has, for each $g \in \mathcal{D}(L(A), X)$:

$$\Phi_{A',X'}(f \circ g \circ L(\varphi)) = R(f) \circ \Phi_{A,X}(g) \circ \varphi$$

Very often in this situation one writes $L \dashv R$ and keep the natural bijection Φ but one has to keep in mind that it is part of the adjunction.

One defines the morphisms

$$\eta_A = \Phi_{A,L(A)}(\mathsf{Id}_{L(A)}) \in \mathcal{C}(A, RL(A))$$
$$\text{and} \quad \varepsilon_X = \Phi^{-1}_{R(X),X}(\mathsf{Id}_{R(X)}) \in \mathcal{D}(LR(X), X)$$

called respectively *unit* and *counit* of the adjunction. They are natural transformations $\eta : \mathsf{Id}_{\mathcal{C}} \overset{\bullet}{\to} RL$ and $\varepsilon : LR \overset{\bullet}{\to} \mathsf{Id}_{\mathcal{D}}$ and satisfy the following equations:

$$R(\varepsilon_X) \circ \eta_{R(X)} = \mathsf{Id}_{R(X)}$$
$$\varepsilon_{L(A)} \circ L(\eta_A) = \mathsf{Id}_{L(A)} \ .$$

Moreover, the data of two functors $L : \mathcal{C} \to \mathcal{D}$, $R : \mathcal{D} \to \mathcal{C}$ and two natural transformations $\eta : \mathsf{Id}_{\mathcal{C}} \overset{\bullet}{\to} RL$ and $\varepsilon : LR \overset{\bullet}{\to} \mathsf{Id}_{\mathcal{D}}$ satisfying the equations above induce uniquely an adjunction of which these two natural transformations are the unit and counit.

C.4 Monads and comonads

Let $\mathcal{C}$ be a category. A *monad* on $\mathcal{C}$ is a triple (T, ε, μ) where $T : \mathcal{C} \to \mathcal{C}$ is a functor, $\varepsilon : \mathsf{Id}_{\mathcal{C}} \stackrel{\bullet}{\to} T$ and $\mu : T^2 = T \circ T \stackrel{\bullet}{\to} T$ are natural transformations. One requires moreover the following commutations..

$$\begin{array}{ccc} T(X) & \xrightarrow{\varepsilon_{T(X)}} & T^2(X) \\ & \searrow^{\mathsf{Id}_{T(X)}} & \downarrow{\scriptstyle \mu_X} \\ & & T(X) \end{array} \qquad \begin{array}{ccc} T(X) & \xrightarrow{T(\varepsilon_X)} & T^2(X) \\ & \searrow^{\mathsf{Id}_{T(X)}} & \downarrow{\scriptstyle \mu_X} \\ & & T(X) \end{array} \qquad \begin{array}{ccc} T^3(X) & \xrightarrow{T(\mu_X)} & T^2(X) \\ \downarrow{\scriptstyle \mu_{T(X)}} & & \downarrow{\scriptstyle \mu_X} \\ T^2(X) & \xrightarrow{\mu_X} & T(X) \end{array}$$

One defines first the category of T-algebras $\mathcal{C}^T$, also called the *Eilenberg-Moore category of* T: the objects of $\mathcal{C}^T$ are the pairs (X, h) where $X \in \mathsf{Obj}(\mathcal{C})$ and $h \in \mathcal{C}(T(X), X)$ such that the following diagrams commute.

$$\begin{array}{ccc} X & \xrightarrow{\varepsilon_X} & T(X) \\ & \searrow^{\mathsf{Id}_X} & \downarrow{\scriptstyle h} \\ & & X \end{array} \qquad \begin{array}{ccc} T^2(X) & \xrightarrow{T(h)} & T(X) \\ \downarrow{\scriptstyle \mu_X} & & \downarrow{\scriptstyle h} \\ T(X) & \xrightarrow{h} & X \end{array}$$

The elements of $\mathcal{C}^T((X, h), (Y, k))$ are the $f \in \mathcal{C}(X, Y)$ such that the following diagram commutes.

$$\begin{array}{ccc} T(X) & \xrightarrow{h} & X \\ \downarrow{\scriptstyle T(f)} & & \downarrow{\scriptstyle f} \\ T(Y) & \xrightarrow{k} & Y \end{array}$$

One defines next the category of free T-algebras, or *Kleisli cetegory*, denoted as $\mathcal{C}_T$. First one sets $\mathsf{Obj}(\mathcal{C}_T) = \mathsf{Obj}(\mathcal{C})$. Then $\mathcal{C}_T(X, Y) = \mathcal{C}(X, T(Y))$. In this category, the identity at X is $\mathrm{Id}_X^{\mathsf{K}} = \varepsilon_X$ and composition is defined in the following way. Let $f \in \mathcal{C}_T(X, Y) = \mathcal{C}(X, T(Y))$ and $g \in \mathcal{C}_T(Y, Z) = \mathcal{C}(Y, T(Z))$. Then

$$g \circ^{\mathsf{K}} f = \mu_Z \circ T(g) \circ f \,.$$

Exercise C.4.1. Prove that we have defined a category $\mathcal{C}_T$.

Exercise C.4.2. If X is an object of $\mathcal{C}$, check that $(T(X), \mu_X)$ is a T-algebra. It is called the *free T-algebra generated by X* and denoted here as $F(X)$. Let $f \in \mathcal{C}_T(X, Y)$. We set $F(f) = \mu_Y \circ T(f)$. Prove that, in that way, one has defined a functor $F : \mathcal{C}_T \to \mathcal{C}^T$. Prove that this functor is full and faithful.

Exercise C.4.3. Let $M :$ **Set** $\to$ **Set** the functor which, with any set X, associates the set $M(X)$ of all finite sequences $\langle a_1, \ldots, a_n \rangle$ of elements of X and with any function $f : X \to Y$ associates the function $M(f) : M(X) \to M(Y)$ which maps $\langle a_1, \ldots, a_n \rangle$ to $\langle f(a_1), \ldots, f(a_n) \rangle$. If X is a set, one defines $\varepsilon_X : X \to M(X)$ as the functions which maps a to $\langle a \rangle$, and $\mu_X : M(M(X)) \to M(X)$ as the function which maps a sequence $\langle m_1, \ldots, m_n \rangle$ of finite sequences of elements of X to their concatenation $m_1 \cdots m_n$.

- Prove that ε and μ are natural transformations.
- Prove that (M, ε, μ) is a monad.
- Prove that $\mathbf{Set}^M$ is the category of monoids and morphisms of monoids.
- Explain why $\mathbf{Set}_M$ can be considered as the category of free monoids and morphisms of monoids.

Exercise C.4.4. Let $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ be the functor which maps a set X to its powerset $\mathcal{P}(X)$ and $f \in \mathbf{Set}(X, Y)$ to the function $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ such that $\mathcal{P}(f)(x) = \{f(a) \mid a \in X\}$. Check that $\mathcal{P}$ is a functor. Find a structure of monad for this functor such that that category $\mathbf{Set}_{\mathcal{P}}$ is isomorphic to **Rel**.

Prove that the Eilenberg-Moore category of $\mathcal{P}$ is a the category of complete sup-semilattices and functions which commute with all suprema. Give an explicit description of the "inclusion" of the category **Rel** in the category of complete lattices.

Reversing the direction of all arrows, we obtain the notion of comonad and of Eilenberg-Moore and Kleisli categories of a comonad. Since comonads will be quite important in the sequel, we spell out these definitions. A *comonad* on $\mathcal{C}$ is a triple (S, δ, λ) where $S : \mathcal{C} \to \mathcal{C}$ is a functor and $\delta : S \stackrel{\bullet}{\to} \mathsf{Id}_{\mathcal{C}}$ and $\lambda : S \stackrel{\bullet}{\to} S \circ S$ make following diagrams commute:

$$\begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ & \searrow^{\mathsf{Id}_{S(X)}} & \downarrow{\scriptstyle \delta_{S(X)}} \\ & & S(X) \end{array} \qquad \begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ & \searrow^{\mathsf{Id}_{S(X)}} & \downarrow{\scriptstyle S(\delta_X)} \\ & & S(X) \end{array} \qquad \begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ {\scriptstyle \lambda_X}\downarrow & & \downarrow{\scriptstyle \lambda_{S(X)}} \\ S^2(X) & \xrightarrow{S(\lambda_X)} & S^3(X) \end{array}$$

An S-coalgebra is a pair (X, h) where X is an object of $\mathcal{C}$ and $h \in \mathcal{C}(X, S(X))$ satisfies the following commutations

$$\begin{array}{ccc} X & \xrightarrow{h} & S(X) \\ & \searrow^{\mathsf{Id}_X} & \downarrow{\scriptstyle \delta_X} \\ & & X \end{array} \qquad \begin{array}{ccc} X & \xrightarrow{h} & S(X) \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle \lambda_X} \\ S(X) & \xrightarrow{S(h)} & S^2(X) \end{array}$$

and a morphism from a coalgebra (X, h) to a coalgebra (Y, k) is an $f \in \mathcal{C}(X, Y)$ such that the following diagram commutes

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle k} \\ S(X) & \xrightarrow{S(f)} & S(Y) \end{array}$$

This defines a the category of coalgebras of the comonad S, or Eilenberg-Moore category of S, denoted as $\mathcal{C}^S$. We use the notation $P = (\underline{P}, \mathsf{h}_P)$ for an object of $\mathcal{C}^S$.

If X is an object of $\mathcal{C}$ then $(S(X), \lambda_X)$ is a coalgebra of S, it is the *cofree* coalgebra generated by X.

Exercise C.4.5. Consider the category whose objects are the pairs (P, d) where P is an object of $\mathcal{C}^S$ and $d \in \mathcal{C}(\underline{P}, X)$. A morphism $(P, d) \to (Q, e)$ in that category is an $f \in \mathcal{C}^S(P, Q)$ such that $e \circ f = d$. This is the category of *coalgebras above* X. Prove that $((S(X), \lambda_X), \delta_X)$ is the terminal object of that category, justifying the terminology "cofree" for this coalgebra above X.

Lemma C.4.6. *The function*

$$\varphi_{X,Y} : \mathcal{C}^S((S(X), \lambda_X), (S(Y), \lambda_Y)) \to \mathcal{C}(S(X), Y)$$
$$f \mapsto \delta_Y \circ f$$

is natural in $X, Y \in \mathsf{Obj}(\mathcal{C})$ *and is a bijection.*

Proof sketch. Naturality results from that of δ. Given $g \in \mathcal{C}(S(X), Y)$ we define $\psi(g) = S(g) \circ \lambda_X \in \mathcal{C}(S(X), S(Y))$. One proves then that $\psi(g) \in \mathcal{C}^S((S(X), \lambda_X), (S(Y), \lambda_Y))$ and that ψ is the inverse of $\varphi_{X,Y}$ using the definition of a comonad and the definition of the Eilenberg Moore category. □

This lemma motivates the definition of the Klesili category $\mathcal{C}_S$ of the comonad S: its objects are those of $\mathcal{C}$ and $\mathcal{C}_S(X, Y) = \mathcal{C}(S(X), Y)$. In that category the identity at X is δ_X and given $f \in \mathcal{C}_S(X, Y)$ and $g \in \mathcal{C}_S(Y, Z)$, their composition $g \circ f \in \mathcal{C}_S(Y, Z)$ is given by

$$g \circ f = g \circ S(f) \circ \lambda_X$$

and then using Lemma C.4.6 one proves that one defines a functor $E : \mathcal{C}_S \to \mathcal{C}^S$ by $E(X) = ((S(X), \lambda_X)$ and, for $f \in \mathcal{C}_S(X, Y)$, by $E(f) = S(f) \circ \lambda_X$ and that this functor is full and faithful. In other words $\mathcal{C}_S$ can be understood as full subcategory of $\mathcal{C}$, the category of cofree coalgebras.

C.5 Monoidal categories

Whereas a cartesian category is a category which enjoys a certain *property* (existence of limits for a class of diagrams), a monoidal category is not just a category, but a category equipped with an additional structure, exactly as a monoid is not just a set but a set equipped with an additional (algebraic) structure.

A symmetric monoidal category (SMC) is a structure $(\mathcal{L}, \mathsf{I}, \boxtimes, \lambda^{\boxtimes}, \rho^{\boxtimes}, \alpha^{\boxtimes}, \sigma^{\boxtimes})$ where

- $\mathcal{L}$ is a category,
- $\boxtimes : \mathcal{L}^2 \to \mathcal{L}$ is a functor and $\mathsf{I} \in \mathsf{Obj}(\mathcal{L})$,
- $\lambda^{\boxtimes}_X \in \mathcal{L}(\mathsf{I} \boxtimes X, X)$ and $\rho^{\boxtimes}_X \in \mathcal{L}(X \boxtimes \mathsf{I}, X)$ are isos which are natural in X,
- $\alpha^{\boxtimes}_{X_1,X_2,X_3} \in \mathcal{L}((X_1 \boxtimes X_2) \boxtimes X_3, X_1 \boxtimes (X_2 \boxtimes X_3))$ is an iso which is natural in X_1, X_2 and X_3,

- and $\sigma^{\boxtimes}_{X_1,X_2} \in \mathcal{L}(X_1 \boxtimes X_2, X_2 \boxtimes X_1)$ is an iso which is natural in X_1 and X_2.

Moreover, the following properties are required. First the two morphisms $\lambda^{\boxtimes}_{\mathsf{I}}, \rho^{\boxtimes}_{\mathsf{I}} \in \mathcal{L}(\mathsf{I} \boxtimes \mathsf{I}, \mathsf{I})$ must be equal. Next, the following diagrams must commute.

$$\begin{array}{ccc}
(\mathsf{I} \boxtimes X_1) \boxtimes X_2 & \xrightarrow{\alpha^{\boxtimes}_{\mathsf{I},X_1,X_2}} & \mathsf{I} \boxtimes (X_1 \boxtimes X_2) \\
& {\scriptstyle \lambda^{\boxtimes}_{X_1} \boxtimes X_2} \searrow & \downarrow {\scriptstyle \lambda^{\boxtimes}_{X_1 \boxtimes X_2}} \\
& & X_1 \boxtimes X_2
\end{array}$$

$$\begin{array}{ccc}
(X_1 \boxtimes \mathsf{I}) \boxtimes X_2 & \xrightarrow{\alpha^{\boxtimes}_{X_1,\mathsf{I},X_2}} & X_1 \boxtimes (\mathsf{I} \boxtimes X_2) \\
& {\scriptstyle \rho^{\boxtimes}_{X_1} \boxtimes X_2} \searrow & \downarrow {\scriptstyle X_1 \boxtimes \lambda^{\boxtimes}_{X_2}} \\
& & X_1 \boxtimes X_2
\end{array}$$

$$\begin{array}{ccc}
(X_1 \boxtimes X_2) \boxtimes \mathsf{I} & \xrightarrow{\alpha^{\boxtimes}_{X_1,X_2,\mathsf{I}}} & X_1 \boxtimes (X_2 \boxtimes \mathsf{I}) \\
& {\scriptstyle \rho^{\boxtimes}_{X_1 \boxtimes X_2}} \searrow & \downarrow {\scriptstyle X_1 \boxtimes \rho^{\boxtimes}_{X_2}} \\
& & X_1 \boxtimes X_2
\end{array}$$

$$\begin{array}{ccc}
((X_1 \boxtimes X_2) \boxtimes X_3) \boxtimes X_4 & \xrightarrow{\alpha^{\boxtimes}_{X_1 \boxtimes X_2,X_3,X_4}} & (X_1 \boxtimes X_2) \boxtimes (X_3 \boxtimes X_4) \\
{\scriptstyle \alpha^{\boxtimes}_{X_1,X_2,X_3} \boxtimes X_4} \downarrow & & \\
(X_1 \boxtimes (X_2 \boxtimes X_3)) \boxtimes X_4 & & \downarrow {\scriptstyle \alpha^{\boxtimes}_{X_1,X_2,X_3 \boxtimes X_4}} \\
{\scriptstyle \alpha^{\boxtimes}_{X_1,X_2 \boxtimes X_3,X_4}} \downarrow & & \\
X_1 \boxtimes ((X_2 \boxtimes X_3) \boxtimes X_4) & \xrightarrow{X_1 \boxtimes \alpha^{\boxtimes}_{X_2,X_3,X_4}} & X_1 \boxtimes (X_2 \boxtimes (X_3 \boxtimes X_4))
\end{array}$$

$$\begin{array}{ccc}
\mathsf{I} \boxtimes X & \xrightarrow{\sigma^{\boxtimes}_{\mathsf{I},X}} & X \boxtimes \mathsf{I} \\
& {\scriptstyle \lambda^{\boxtimes}_{X}} \searrow & \downarrow {\scriptstyle \rho^{\boxtimes}_{X}} \\
& & X
\end{array}$$

$$\begin{array}{ccc}
(X_1 \boxtimes X_2) \boxtimes X_3 & \xrightarrow{\alpha^{\boxtimes}_{X_1,X_2,X_3}} & X_1 \boxtimes (X_2 \boxtimes X_3) \\
{\scriptstyle \sigma^{\boxtimes}_{X_1,X_2} \boxtimes X_3} \downarrow & & \downarrow {\scriptstyle \sigma^{\boxtimes}_{X_1,X_2 \boxtimes X_3}} \\
(X_2 \boxtimes X_1) \boxtimes X_3 & & (X_2 \boxtimes X_3) \boxtimes X_1 \\
{\scriptstyle \alpha^{\boxtimes}_{X_2,X_1,X_3}} \downarrow & & \downarrow {\scriptstyle \alpha^{\boxtimes}_{X_2,X_3,X_1}} \\
X_2 \boxtimes (X_1 \boxtimes X_3) & \xrightarrow{X_2 \boxtimes \sigma^{\boxtimes}_{X_1,X_3}} & X_2 \boxtimes (X_3 \boxtimes X_1)
\end{array}$$

$$\begin{array}{ccc} X_1 \boxtimes X_2 & \xrightarrow{\sigma^{\boxtimes}_{X_1,X_2}} & X_2 \boxtimes X_1 \\ & \searrow^{X_1 \boxtimes X_2} & \downarrow{\scriptstyle \sigma^{\boxtimes}_{X_2,X_1}} \\ & & X_1 \boxtimes X_2 \end{array}$$

We define a notion of "monoidal tree" by the following syntax:

$$\tau, \tau_1, \cdots := * \mid n \mid \langle \tau_1, \tau_2 \rangle$$

where n is an integer. We define the degree of a monoidal tree as the number of its integer leaves, in other words

$$\begin{aligned} \mathsf{deg}(*) &= 0 \\ \mathsf{deg}(n) &= 1 \\ \mathsf{deg}(\langle \tau_1, \tau_2 \rangle) &= \mathsf{deg}(\tau_1) + \mathsf{deg}(\tau_2)\,. \end{aligned}$$

We say that a monoidal tree of degree n is well-formed if the set of its labels is $\{1, \ldots, n\}$.

Given a well-formed monoidal tree τ of degree n and a sequence $\overrightarrow{X} = (X_1, \ldots, X_k)$ of objects of $\mathcal{L}$ with $k \geq n$, we can define an object $\mathsf{T}^{\boxtimes}_{\tau}(\overrightarrow{X})$ of $\mathcal{L}$ as follows:

$$\begin{aligned} \mathsf{T}^{\boxtimes}_{*}(\overrightarrow{X}) &= \mathsf{I} \\ \mathsf{T}^{\boxtimes}_{i}(\overrightarrow{X}) &= X_i \\ \mathsf{T}^{\boxtimes}_{\langle \tau_1, \tau_2 \rangle}(\overrightarrow{X}) &= \mathsf{T}^{\boxtimes}_{\tau_1}(\overrightarrow{X}) \boxtimes \mathsf{T}^{\boxtimes}_{\tau_2}(\overrightarrow{X})\,. \end{aligned}$$

Then, given two well-formed monoidal trees τ_1 and τ_2 of degree n and a sequence $\overrightarrow{X} = (X_1, \ldots, X_n)$ of objects of $\mathcal{L}$, it is possible, using the natural transformations $\lambda^{\boxtimes}$, $\rho^{\boxtimes}$, $\alpha^{\boxtimes}$ and $\sigma^{\boxtimes}$, to define isomorphisms in $\mathcal{L}(\mathsf{T}^{\boxtimes}_{\tau_1}(\overrightarrow{X}), \mathsf{T}^{\boxtimes}_{\tau_2}(\overrightarrow{X}))$. The coherence diagrams above allow to prove that all these morphisms are actually equal: this is *Mac Lane's coherence theorem*. We use $\varphi^{\boxtimes}_{\tau_1,\tau_2}(\overrightarrow{X})$ to denote this iso. Of course we have $\varphi^{\boxtimes}_{\tau,\tau}(\overrightarrow{X}) = \mathsf{Id}$ and $\varphi^{\boxtimes}_{\tau_2,\tau_3}(\overrightarrow{X})\,\varphi^{\boxtimes}_{\tau_1,\tau_2}(\overrightarrow{X}) = \varphi^{\boxtimes}_{\tau_1,\tau_3}(\overrightarrow{X})$.

Exercise C.5.1. Prove that any cartesian category has a canonical structure of monoidal category, with & as monoidal bifunctor.

C.5.1 Commutative comonoids

Definition C.5.2. In a SMC $\mathcal{L}$ (with the usual notations), a commutative comonoid is a tuple $C = (\underline{C}, \mathsf{w}_C, \mathsf{c}_C)$ where $\underline{C} \in \mathcal{L}$, $\mathsf{w}_C \in \mathcal{L}(\underline{C}, 1)$ and $\mathsf{c}_C \in \mathcal{L}(\underline{C}, \underline{C} \otimes \underline{C})$ are such that the following diagrams commute.

$$\begin{array}{ccc} \underline{C} & \xrightarrow{\mathsf{c}_C} & \underline{C} \otimes \underline{C} \\ & \searrow^{(\lambda_{\underline{C}})^{-1}} & \downarrow{\scriptstyle \mathsf{w}_C \otimes \underline{C}} \\ & & 1 \otimes \underline{C} \end{array} \qquad \begin{array}{ccc} \underline{C} & \xrightarrow{\mathsf{c}_C} & \underline{C} \otimes \underline{C} \\ & \searrow^{\mathsf{c}_C} & \downarrow{\scriptstyle \sigma_{\underline{C},\underline{C}}} \\ & & \underline{C} \otimes \underline{C} \end{array}$$

$$\begin{array}{ccccc} \underline{C} & & \xrightarrow{\quad\mathsf{c}_C\quad} & & \underline{C}\otimes\underline{C} \\ {\scriptstyle \mathsf{c}_C}\downarrow & & & & \downarrow{\scriptstyle \underline{C}\otimes\mathsf{c}_C} \\ \underline{C}\otimes\underline{C} & \xrightarrow{\mathsf{c}_C\otimes\underline{C}} & (\underline{C}\otimes\underline{C})\otimes\underline{C} & \xrightarrow{\alpha_{\underline{C},\underline{C},\underline{C}}} & \underline{C}\otimes(\underline{C}\otimes\underline{C}) \end{array}$$

The category $\mathcal{L}^{\otimes}$ of commutative comonoids has these tuples as objects, and an element of $\mathcal{L}^{\otimes}(C,D)$ is an $f \in \mathcal{L}(\underline{C},\underline{D})$ such that the two following diagrams commute

$$\begin{array}{ccc} \underline{C} & \xrightarrow{f} & \underline{D} \\ & {\scriptstyle \mathsf{w}_C}\searrow & \downarrow{\scriptstyle \mathsf{w}_D} \\ & & 1 \end{array} \qquad \begin{array}{ccc} \underline{C} & \xrightarrow{f} & \underline{D} \\ {\scriptstyle \mathsf{c}_C}\downarrow & & \downarrow{\scriptstyle \mathsf{c}_D} \\ \underline{C}\otimes\underline{C} & \xrightarrow{f\otimes f} & \underline{D}\otimes\underline{D} \end{array}$$

Theorem C.5.3. *For any SMC $\mathcal{L}$ the category $\mathcal{L}^{\otimes}$ is cartesian. The terminal object is $(1, \mathsf{Id}_1, (\lambda_1)^{-1})$ (remember that $\lambda_1 = \rho_1$) simply denoted as 1 and for any object C the unique morphism $C \to 1$ is w_C. The cartesian product of $C_0, C_1 \in \mathcal{L}^{\otimes}$ is the object $C_0 \otimes C_1$ of $\mathcal{L}^{\otimes}$ such that $\underline{C_0 \otimes C_1} = \underline{C_0} \otimes \underline{C_1}$ and the structure maps are defined as*

$$\underline{C_0}\otimes\underline{C_1} \xrightarrow{\mathsf{w}_{C_0}\otimes\mathsf{w}_{C_1}} 1\otimes 1 \xrightarrow{\quad\lambda_1\quad} 1$$

$$\underline{C_0}\otimes\underline{C_1} \xrightarrow{\mathsf{c}_{C_0}\otimes\mathsf{c}_{C_1}} \underline{C_0}\otimes\underline{C_0}\otimes\underline{C_1}\otimes\underline{C_1} \xrightarrow{\sigma_{2,3}} \underline{C_0}\otimes\underline{C_1}\otimes\underline{C_0}\otimes\underline{C_1}$$

The projections $\mathsf{pr}_i^{\otimes} \in \mathcal{L}^{\otimes}(C_0 \otimes C_1, C_i)$ are given by

$$\begin{array}{l} \underline{C_0}\otimes\underline{C_1} \xrightarrow{\mathsf{w}_{C_0}\otimes\underline{C_1}} 1\otimes\underline{C_1} \xrightarrow{\lambda_{\underline{C_1}}} \underline{C_1} \\ \underline{C_0}\otimes\underline{C_1} \xrightarrow{\underline{C_0}\otimes\mathsf{w}_{C_1}} \underline{C_0}\otimes 1 \xrightarrow{\rho_{\underline{C_0}}} \underline{C_0} \end{array} .$$

The proof is straightforward. In a commutative monoid M, multiplication is a monoid morphism $M \times M \to M$. The following is in the vein of this simple observation.

Lemma C.5.4. *If $C \in \mathcal{L}^{\otimes}$ then $\mathsf{w}_C \in \mathcal{L}^{\otimes}(C,1)$ and $\mathsf{c}_C \in \mathcal{L}^{\otimes}(C, C\otimes C)$.*

Proof. The second statement amounts to the following commutation

$$\begin{array}{ccccc} \underline{C} & & \xrightarrow{\quad\mathsf{c}_C\quad} & & \underline{C}\otimes\underline{C} \\ {\scriptstyle \mathsf{c}_C}\downarrow & & & & \downarrow{\scriptstyle \mathsf{c}_C\otimes\mathsf{c}_C} \\ \underline{C}\otimes\underline{C} & \xrightarrow{\mathsf{c}_C\otimes\mathsf{c}_C} & \underline{C}\otimes\underline{C}\otimes\underline{C}\otimes\underline{C} & \xrightarrow{\sigma_{2,3}} & \underline{C}\otimes\underline{C}\otimes\underline{C}\otimes\underline{C} \end{array}$$

which results from the commutativity of C. The first statement is similarly trivial. □

Bibliography

[1] Samson Abramsky, Esfandiar Haghverdi, and Philip J. Scott. "Geometry of Interaction and Linear Combinatory Algebras". In: *Mathematical Structures in Computer Science* 12.5 (2002), pp. 625–665.

[2] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. "Full Abstraction for PCF". In: *Information and Computation* 163.2 (Dec. 2000), pp. 409–470.

[3] Andrea Asperti and Cosimo Laneve. "Paths, Computations and Labels in the Lambda-Calculus". In: *Rewriting Techniques and Applications*. Vol. 690. Lecture Notes in Computer Science. Springer-Verlag, 1993, pp. 152–167.

[4] Andrea Asperti et al. "Paths in the Lambda-Calculus". In: *Proceedings of the ninth Symposium on Logic In Computer Science*. IEEE. Paris: IEEE Computer Society Press, July 1994.

[5] William Averson. *An invitation to* $\mathbf{C}^*$*-algebra*. Springer-Verlag, 1976.

[6] Gianluigi Bellin and Jacques Van de Wiele. "Subnets of Proof-nets in MLL−". In: *Advances in Linear Logic*. Ed. by Jean-Yves Girard, Yves Lafont, and Laurent Regnier. Vol. 222. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995, pp. 249–270.

[7] Gérard Berry. "Stable Models of Typed Lambda-calculi". In: *International Colloquium on Automata, Languages and Programming*. Vol. 62. Lecture Notes in Computer Science. Springer, 1978.

[8] Richard Blute et al. "Natural Deduction and Coherence for Weakly Distributive Categories". In: *Journal of Pure and Applied Algebra* 113 (1996), pp. 229–296.

[9] Vincent Danos. "La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul)". Thèse de Doctorat. U, 1990.

[10] Vincent Danos, Hugo Herbelin, and Laurent Regnier. "Game semantics & abstract machines". In: *Proceedings of the eleventh annual symposium on Logic In Computer Science*. IEEE. New Brunswick: IEEE Computer Society Press, July 1996, pp. 394–405.

[11] Vincent Danos and Laurent Regnier. "Proof-Nets and the Hilbert Space". In: *Advances in Linear Logic*. Ed. by Jean-Yves Girard, Yves Lafont, and Laurent Regnier. Vol. 222. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995, pp. 307–328.

[12] Vincent Danos and Laurent Regnier. "Reversible, Irreversible and Optimal λ-Machines". In: *Theoretical Computer Science* 227 (Sept. 1999), pp. 79–97.

[13] Vincent Danos and Laurent Regnier. "The structure of multiplicatives". In: *Archive for Mathematical Logic* 28 (1989), pp. 181–203.

[14] Roberto Di Cosmo, Delia Kesner, and Emmanuel Polonovski. "Proof nets and explicit substitutions". In: *Mathematical Structures in Computer Science* 13.3 (June 2003), pp. 409–450.

[15] Jean-Yves Girard. "Geometry of Interaction I: an Interpretation of System *F*". In: *Logic Colloquium '88*. Ed. by Ferro et al. North-Holland, 1988.

[16] Jean-Yves Girard. "Geometry of interaction II: Deadlock Free Algorithms". In: *Proceedings of COLOG '88*. Ed. by Martin-Löf and Mints. Vol. 417. Lecture Notes in Computer Science. Springer, 1990, pp. 76–93.

[17] Jean-Yves Girard. "Geometry of interaction II: Deadlock Free Algorithms". In: *Proceedings of COLOG'88*. Ed. by Martin-Löf and Mints. Vol. 417. Lecture Notes in Computer Science. Heidelberg: Springer-Verlag, 1990, pp. 76–93.

[18] Jean-Yves Girard. "Geometry of interaction III: accommodating the additives". In: *Advances in Linear Logic*. Ed. by Jean-Yves Girard, Yves Lafont, and Laurent Regnier. Vol. 222. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995, pp. 329–389.

[19] Jean-Yves Girard. "Geometry of interaction V: Logic in the hyperfinite factor". In: *Theoretical Computer Science* 412 (2011), pp. 1860–1883.

[20] Jean-Yves Girard. "Linear logic". In: *Theoretical Computer Science* 50 (1987), pp. 1–102.

[21] Jean-Yves Girard. "Proof-nets: the parallel syntax for proof-theory". In: *Logic and Algebra*. Ed. by Aldo Ursini and Paolo Agliano. Vol. 180. Lecture Notes In Pure and Applied Mathematics. New York: Marcel Dekker, 1996, pp. 97–124.

[22] Jean-Yves Girard. "Quantifiers in Linear Logic II". In: *Nuovi problemi della logica e della filosofia della scienza*. Ed. by Corsi and Sambin. Bologna: CLUEB, 1991, pp. 79–90.

[23] Jean-Yves Girard. "The system F of variable types, fifteen years later". In: *Theoretical Computer Science* 45 (1986), pp. 159–192.

[24] Jean-Yves Girard. "Towards a geometry of interaction". In: *Categories in Computer Science and Logic*. Proceedings of Symposia in Pure Mathematics n°92. Providence: American Mathematical Society, 1989, pp. 69–108.

[25] Jean-Yves Girard, Yves Lafont, and Laurent Regnier, eds. *Advances in Linear Logic*. Vol. 222. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995.

[26] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. "The Geometry of Optimal Lambda Reduction". In: *Proceedings of the 19th Annual ACM Symposium on Principles of Programming Languages*. Association for Computing Machinery. ACM Press, 1992, pp. 15–26.

[27] Stefano Guerrini. "A linear algorithm for MLL proof net correctness and sequentialization". In: *Theoretical Computer Science* 412.20 (2011), pp. 1958–1978.

[28] Esfandiar Haghverdi and Philip J. Scott. "A Categorical Model for the Geometry of Interaction". In: *Theoretical Computer Science* 350.2-3 (Jan. 2004), pp. 252–274. DOI: 10.1016/j.tcs.2005.10.028.

[29] Willem Heijltjes and Robin Houston. "Proof equivalence in MLL is PSPACE-complete". In: *Logical Methods in Computer Science* 12 (1 Mar. 2016). DOI: 10.2168/LMCS-12(1:2)2016. URL: http://lmcs.episciences.org/1625.

[30] Dominic Hughes. "Simple multiplicative proof nets with units". In: *Annals of Pure and Applied Logic* (2013). To appear. Available at `http://arxiv.org/abs/math/0507003` .

[31] Dominic Hughes and Rob van Glabbeek. "Proof Nets for Unit-free Multiplicative-Additive Linear Logic". In: *ACM Transactions on Computational Logic* 6.4 (2005), pp. 784–842.

[32] André Joyal, Ross Street, and Dominic Verity. "Traced Monoidal Categories". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 119 (1996), pp. 447–468.

[33] J. Lamping. "An algorithm for optimal lambda calculus reductions". In: *Proceedings of the 17th Annual ACM Symposium on Principles of Programming Languages*. Association for Computing Machinery. San Francisco: ACM Press, 1990, pp. 16–30.

[34] Olivier Laurent. "Sequentialization of Multiplicative Proof Nets". Unpublished note. Available at `http://perso.ens-lyon.fr/olivier.laurent/seqmll.pdf` . Apr. 2013.

[35] Jean-Jacques Lévy. "Lambda-Calcul Étiqueté". Thèse de Doctorat. Université Paris 7, 1978.

[36] Jean-Jacques Lévy. "Optimal Reduction in the Lambda-Calculus". In: *To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism*. Ed. by J. Hindley and J. Seldin. Academic Press, 1980.

[37] Pasquale Malacaria and Laurent Regnier. "Some Results on the Interpretation of λ-calculus in Operator Algebras". In: *Proceedings of the sixth Symposium on Logic In Computer Science*. IEEE. IEEE Computer Society Press, 1991.

[38] Paulin Jacobé de Naurois and Virgile Mogbil. "Correctness of linear logic proof structures is NL-complete". In: *Theoretical Computer Science* 412.20 (2011), pp. 1941–1957. DOI: 10.1016/j.tcs.2010.12.020. URL: https://doi.org/10.1016/j.tcs.2010.12.020.

[39] Michele Pagani and Lorenzo Tortora de Falco. "Strong normalization property for second order linear logic". In: *Theoretical Computer Science* 411.2 (2010), pp. 410–444.

[40] Mario Petrich. *Inverse Semigroups*. Pure and Applied Mathematics. John Wiley & sons, 1984.

[41] Laurent Regnier. "Une équivalence sur les lambda-termes". In: *Theoretical Computer Science* 126 (1994), pp. 281–292.

[42] Walter Rudin. *Real and complex analysis*. McGraw-Hill Education, 1987.

[43] Thomas Seiller. "Logique dans le facteur hyperfini, géométrie de l'interaction et complexité". Thèse de Doctorat. Université d'Aix-Marseille, Nov. 2012.

[44] Terese. *Term Rewriting Systems*. Vol. 55. Cambridge tracts in theoretical computer science. Cambridge University Press, 2003.

Index