Handbook of Linear Logic

(very early draft: mostly incomplete, sometimes incorrect)

The LLHandbook project

December 8, 2022

Contents

I Proofs **7**

1 Sequents **9**

1.1 Formulas . . . 9
1.2 Sequents and proofs . . . 10
1.3 Basic properties . . . 13
1.3.1 Multiset-based sequent rules . . . 13
1.3.2 Expansion of identities . . . 14
1.3.3 Linear equivalences . . . 14
1.3.4 Deduction lemma . . . 17
1.3.5 One-sided sequent calculus . . . 19
1.4 Some fragments of interest . . . 23
1.5 Cut elimination and consequences . . . 25
1.5.1 A proof for propositional linear logic . . . 25
1.5.2 Consequences . . . 36
1.6 Reversibility and focusing . . . 37
1.6.1 Reversibility . . . 37
1.6.2 Generalized connectives and rules . . . 40
1.6.3 Focusing . . . 41
1.7 Variations . . . 43
1.7.1 Exponential rules . . . 43
1.7.2 Non-symmetric sequents . . . 43
1.7.3 Mix rules . . . 44
1.7.4 Dyadic sequent calculus . . . 44

2 Proof nets **47**

2.1 Multiplicative Proof Nets . . . 47
2.1.1 Proof structures . . . 47
2.1.2 Proof nets . . . 51
2.1.3 Cut Elimination . . . 55
2.1.4 Sequentialization . . . 59
2.2 Multiplicative units and the mix rules . . . 68
2.2.1 Multiplicative units and jumps . . . 68
2.2.2 The Mix Rules . . . 73
2.3 Multiplicative Exponential Proof Nets . . . 78

2.3.1 Multiplicative Exponential Linear Logic with Mix 78
2.3.2 Proof Structures 79
2.3.3 Correctness Criterion 79
2.3.4 Cut Elimination 79
2.3.5 Generalized ? Nodes 86
2.4 Translation of the Lambda-Calculus 87
2.4.1 The Lambda-Calculus inside Linear Logic 87
2.4.2 Directed Proof Nets 88
2.4.3 The Translation 88
2.4.4 Untyped Lambda-Calculus 91
2.5 Further Reading 91
2.5.1 Historical Papers 91
2.5.2 Sequentialization 92
2.5.3 Rewriting Properties 92
2.5.4 Extensions of the Syntax 92
2.5.5 Relations with the Lambda-Calculus 92
2.5.6 Complexity 92

II Static models 93

3 Coh 95
3.1 Coherent spaces 95
3.1.1 The coherence relation 95
3.1.2 Clique spaces 96
3.2 The cartesian closed structure of coherent spaces 98
3.2.1 Stable functions 98
3.2.2 Cartesian product 100
3.2.3 The coherent space of stable functions 100
3.3 The monoidal structure of coherent spaces 102
3.3.1 Linear functions 102
3.3.2 Tensor product 103
3.3.3 The coherent space of linear functions 103
3.3.4 Duality 105
3.3.5 Additive constructions 105
3.3.6 Multiplicative constructions 107
3.4 Exponentials 107
3.5 Conclusion 110

4 Rel 113
4.1 Relational interpretation of the sequent calculus: resource derivations 113
4.2 The relational model as a category 117
4.3 Enriching the relational model with a (non-uniform) coherence structure 120
4.3.1 Boudes' exponential 122

4.3.2 Another exponential . . . 124

5 Categories 127
5.1 Categories, functors and natural transformations . . . 127
5.1.1 Functors . . . 128
5.1.2 Natural transformations . . . 129
5.2 Limits and colimits . . . 130
5.2.1 Projective limits (limits) . . . 130
5.3 Adjunctions . . . 133
5.4 Monads and comonads . . . 133
5.5 Monoidal categories . . . 136
5.5.1 Commutative comonoids . . . 138
5.6 Seely categories . . . 140
5.6.1 The multiplicative structure . . . 140
5.6.2 The additive structure . . . 142
5.6.3 The exponential structure . . . 143
5.6.4 Derived structures in a Seely category . . . 145
5.6.5 Free exponentials and Lafont categories . . . 148
5.6.6 Interpreting the sequent calculus in a Seely category . . . 148

6 Scott 149
6.1 Scott semantics . . . 149
6.1.1 Multiplicative structure . . . 150
6.1.2 Additive structure . . . 151
6.1.3 Exponential structure . . . 151
6.2 Relation with the relational model . . . 153
6.2.1 A duality on preorders . . . 153
6.2.2 The category of preorders with projections . . . 154

III Dynamic models 159

7 Geometry of Interaction 161
7.1 A brief and partial history of the geometry of interaction . . . 161
7.1.1 Notations and conventions . . . 162
7.2 Paths in proof nets . . . 163
7.2.1 The lifting functor . . . 163
7.2.2 Path residuals . . . 169
7.2.3 Path reductions . . . 170
7.2.4 Straight paths . . . 171
7.2.5 Persistent paths . . . 172
7.2.6 The special cut lemma . . . 173
7.3 An algebraic caracterisation of persistent paths . . . 175
7.3.1 The dynamic algebra Λ^* . . . 175
7.3.2 An easy model . . . 176
7.3.3 ab^* forms . . . 178

7.3.4 Weight of paths 178
7.3.5 Regular paths 179
7.4 Proof-nets as operators 183
7.4.1 Lifting partial permutations to operators on ℓ^2 183
7.4.2 Graphs and matrices 185
7.4.3 Strong topology and strong normalization 189
7.4.4 Weak topology and cycles 190
7.4.5 Conclusion 192
7.5 The Interaction Abstract Machine 192
7.5.1 The interaction model 193
7.5.2 The Interaction Abstract Machine (IAM) 195
7.5.3 Legal paths 198

A Graphs **209**

B Abstract Reduction Systems **213**
B.0.1 Definitions and Notations 213
B.0.2 Confluence 214
B.0.3 Normalization 216
B.0.4 Simulation 219
B.0.5 Commutation 220

Part I

Proofs

Chapter 1

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.

1.1 Formulas

The *formulas* of Linear Logic are defined by Table 1.1. 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 1.1 (except the first one) corresponds to a particular class of connectives, and each class consists in a pair of connectives. Those in the left column are called *positive* and those in the right column are called *negative* (see

	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	$A \mathbin{\&} B$	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 1.1: Formulas of Linear Logic.

Section 1.6 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 $⅋$ and $\oplus$ to $\vee$ turns linear provable formulas into valid classical formulas. The exponential connectives are 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.

The *linear implication* and the *linear equivalence* are presented as defined multiplicative connectives, by $A \multimap B := A^{\perp} ⅋ B$ and $A \multimapboth B := (A^{\perp} ⅋ B) \otimes (A ⅋ B^{\perp})$, 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 $A \multimapboth B$ 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 $B[x_1, \ldots, x_n]$ is a formula with variables among x_i, then the formula $A[B/X]$ is A where every atom $X(t_1, \ldots, t_n)$ is replaced by $B[t_1/x_1, \ldots, t_n/x_n]$. For example, $(\forall y.X(y))[\forall z.p(x,z)/X] = \forall y.\forall z.p(y,z)$.

1.2 Sequents and proofs

A sequent is an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite sequences of formulas. For a sequence $\Gamma = A_1, \ldots, A_n$, the notation $\$\Gamma$, for $\$ \in \{?, !\}$, represents the sequence $\$A_1, \ldots, \$A_n$, and similarly $\Gamma^{\perp}$ represents the sequence $A_1^{\perp}, \ldots, A_n^{\perp}$. Proofs are labelled trees with nodes labelled with inference rules and edges labelled by sequents[1]. Table 1.2 gives a picture of the LL inference rules together with the labelling of the incident edges. 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 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 1.2 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 of the tensor rule, 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), while all occurrences

[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.

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 of course (resp. why not) 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*.

Notation 1.2.1. 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). A open proof with holes $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$ and conclusion $\Gamma \vdash \Delta$ is also called a *derivation* of $\Gamma \vdash \Delta$ from $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$.

Definition 1.2.2 (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:

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

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 $\mathcal{S}$). The inference rule:

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

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 $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$ 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 1.2, as we will prove in the next section, but it is not derivable.

Identity and negation group

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

Multiplicative group

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

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

Additive group

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

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

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.

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

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

Exponential group

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

Structural rules

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

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

Table 1.2: Inference rules for two-sided Linear Logic sequent calculus

Example 1.2.3. The inference rule $\dfrac{\vdash A \otimes B}{\vdash A}$ (dashed line) 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 1.2.4).

Exercise 1.2.4. Prove that each rule in the additive (resp. multiplicative) group in Table 1.2 is derivable from the multiplicative (resp. additive) group and the structural rules free on every formulas, *i.e.*:

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

1.3 Basic properties

1.3.1 Multiset-based sequent rules

Notice that all the active occurrences in the rules of Table 1.2 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.

Exercise 1.3.1. Given a natural number n, let us denote by $\mathcal{P}_n$ the set of all permutations over n. Given a list of formulas $\Gamma = A_1, \ldots, A_n$ and a permutation $\sigma \in \mathcal{P}_n$, we write $\Gamma \cdot \sigma$ for the action of σ over Γ, *i.e.* the list $A_{\sigma(1)}, \ldots, A_{\sigma(n)}$.

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 1.2 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:
$$\frac{(\Gamma, A) \cdot \sigma \vdash \Delta \cdot \rho \qquad (\Gamma', B) \cdot \sigma' \vdash \Delta' \cdot \rho'}{(\Gamma, \Gamma', A \parr B) \cdot \sigma'' \vdash (\Delta, \Delta') \cdot \rho''}$$
(double line) is derivable from the rule $(\parr_L)$ and the 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 considering sequents as made of finite multisets instead of finite sequences of formulas. We will adopt this convention henceforth, keeping the use of the exchange rule implicit, whenever we focus on provability.

1.3.2 Expansion of identities

The axiom rule in Table 1.2 is defined for any formula. However regarding the expressiveness of the system, it is enough to restrict it to atomic formulas. Indeed, Table 1.3 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 5.6. The definition of $\eta(A)$ is also crucial in the notion of syntactic isomorphism.

1.3.3 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 $A \multimapboth B$ 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 1.3.2. By definition, we have $A \dashv\vdash B$ if and only if $A^\perp \dashv\vdash B^\perp$.

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

Example 1.3.3. For any formulas A and B, $A \otimes B$ and $A \mathbin{\&} B$ are equiprovable. However neither $\perp \otimes \perp \vdash \perp \mathbin{\&} \perp$ nor $\perp \mathbin{\&} \perp \vdash \perp \otimes \perp$ are provable.

Exercise 1.3.4 (Beffara's formula). Prove that $A \otimes (A^\perp \parr A)$ is linearly equivalent to A.

1.3.3.1 De Morgan laws

Negation is involutive:

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

Duality between connectives:

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

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

$$\eta(A^\perp) = \dfrac{\dfrac{\eta(A) : A \vdash A}{A^\perp, A \vdash}\ (n_L)}{A^\perp \vdash A^\perp}\ (n_R) \qquad \eta(\mathbf{1}) = \dfrac{\dfrac{}{\vdash \mathbf{1}}\ (\mathbf{1}_R)}{\mathbf{1} \vdash \mathbf{1}}\ (\mathbf{1}_L) \qquad \eta(\perp) = \dfrac{\dfrac{}{\perp \vdash}\ (\perp_L)}{\perp \vdash \perp}\ (\perp_R)$$

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

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

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

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

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

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

Table 1.3: η-expansion

1.3.3.2 Fundamental equivalences

Associativity, commutativity, neutrality:

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

Idempotence of additives:

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

Distributivity of multiplicatives over additives:

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

Defining property of exponentials:

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

Monoidal structure of exponentials:

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

Idempotence of exponentials:

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

Other properties 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 1.1 and Exercise 1.3.5).

Exercise 1.3.5. 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 !.

1. Prove that $\leq$ defines a preorder on iterated exponential modalities.
2. Prove that $\leq$ is not a total preorder.

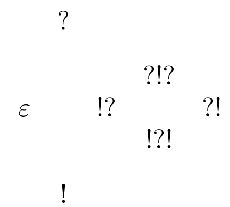


Figure 1.1: The lattice of iterated exponential modalities

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 the order relations pictured on Figure 1.1 hold.
7. Prove that $? \not\leq \varepsilon$, $? \not\leq ?!?$, $?! \not\leq !?$ and $!? \not\leq ?!$.
8. Conclude that Figure 1.1 exactly describes the preorder relation on iterated exponential modalities.

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

$$\begin{array}{llll}
\exists\xi.\exists\psi.A \dashv\vdash \exists\psi.\exists\xi.A & \exists\xi.(A \oplus B) \dashv\vdash \exists\xi.A \oplus \exists\xi.B & \exists\zeta.(A \otimes B) \dashv\vdash A \otimes \exists\zeta.B & \exists\zeta.A \dashv\vdash A \\
\forall\xi.\forall\psi.A \dashv\vdash \forall\psi.\forall\xi.A & \forall\xi.(A \,\&\, B) \dashv\vdash \forall\xi.A \,\&\, \forall\xi.B & \forall\zeta.(A \parr B) \dashv\vdash A \parr \forall\zeta.B & \forall\zeta.A \dashv\vdash A
\end{array}$$

1.3.3.3 Definability

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

$$\begin{aligned}
\mathbf{0} &\dashv\vdash \forall X.X \\
\mathbf{1} &\dashv\vdash \forall X.(X \multimap X) \\
A \oplus B &\dashv\vdash \forall X.(!(A \multimap X) \multimap !(B \multimap X) \multimap X)
\end{aligned}$$

The constants $\top$ and $\bot$ and the connective & can be defined by duality (see Remark 1.3.2).

1.3.4 Deduction lemma

As already mentioned in Section 1.3.3, 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 1.3.6 (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:

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

- $(\otimes_R)$ rule:

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

- $(!_R)$ 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 1.3.7. Prove the rule of example 1.2.3 is not derivable.

Lemma 1.3.8 (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 1.3.6 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:

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

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:

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

1.3.5 One-sided sequent calculus

Notice that Table 1.2 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 1.3.3, the hypothesis and the thesis in a sequent can be exchanged by negation.

Exercise 1.3.9. 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 1.2, while enjoying the following features:

- sequents have the form $\vdash \Gamma$;
- negation is not a connective but a syntactically defined operation on formulas given by the De Morgan laws (see Section 1.3.3);
- the rules are essentially the same as those of the two-sided version, except that the left hand side of sequents is kept empty.

The construction is as follows. We consider *DM-normal* (for De Morgan-normal) formulas, which are the formulas obtained by using the constructs of Table 1.1, except that we restrict linear negation to be applied to atoms only: in the remaining of that section we reserve letters $F, G, \ldots$ such DM-normal formulas. Note that, for each occurrence of an atom α in a DM-normal formula F:

- either this occurrence is under the scope of exactly one linear negation, necessarily inside a subformula $\alpha^\perp$ of F, and we call this subformula a *negated atom*;
- or this occurrence is not under the scope of a linear negation, and we call this occurrence a *positive atom*.

$$
\begin{aligned}
|\alpha|_{DM} &\coloneqq \alpha & \|\alpha\|_{\mathrm{DM}} &\coloneqq \alpha^\perp \\
|A^\perp|_{DM} &\coloneqq \|A\|_{\mathrm{DM}} & \|A^\perp\|_{\mathrm{DM}} &\coloneqq |A|_{DM} \\
|\mathbf{1}|_{DM} &\coloneqq \mathbf{1} & \|\mathbf{1}\|_{\mathrm{DM}} &\coloneqq \bot \\
|\bot|_{DM} &\coloneqq \bot & \|\bot\|_{\mathrm{DM}} &\coloneqq \mathbf{1} \\
|A \otimes B|_{DM} &\coloneqq |A|_{DM} \otimes |B|_{DM} & \|A \otimes B\|_{\mathrm{DM}} &\coloneqq \|A\|_{\mathrm{DM}} \parr \|B\|_{\mathrm{DM}} \\
|A \parr B|_{DM} &\coloneqq |A|_{DM} \parr |B|_{DM} & \|A \parr B\|_{\mathrm{DM}} &\coloneqq \|A\|_{\mathrm{DM}} \otimes \|B\|_{\mathrm{DM}} \\
|\top|_{DM} &\coloneqq \top & \|\top\|_{\mathrm{DM}} &\coloneqq \mathbf{0} \\
|\mathbf{0}|_{DM} &\coloneqq \mathbf{0} & \|\mathbf{0}\|_{\mathrm{DM}} &\coloneqq \top \\
|A \oplus B|_{DM} &\coloneqq |A|_{DM} \oplus |B|_{DM} & \|A \oplus B\|_{\mathrm{DM}} &\coloneqq \|A\|_{\mathrm{DM}} \,\&\, \|B\|_{\mathrm{DM}} \\
|A \,\&\, B|_{DM} &\coloneqq |A|_{DM} \,\&\, |B|_{DM} & \|A \,\&\, B\|_{\mathrm{DM}} &\coloneqq \|A\|_{\mathrm{DM}} \oplus \|B\|_{\mathrm{DM}} \\
|!A|_{DM} &\coloneqq !|A|_{DM} & \|!A\|_{\mathrm{DM}} &\coloneqq ?\|A\|_{\mathrm{DM}} \\
|?A|_{DM} &\coloneqq ?|A|_{DM} & \|?A\|_{\mathrm{DM}} &\coloneqq !\|A\|_{\mathrm{DM}} \\
|\forall\xi.A|_{DM} &\coloneqq \forall\xi.|A|_{DM} & \|\forall\xi.A\|_{\mathrm{DM}} &\coloneqq \exists\xi.\|A\|_{\mathrm{DM}} \\
|\exists\xi.A|_{DM} &\coloneqq \exists\xi.|A|_{DM} & \|\exists\xi.A\|_{\mathrm{DM}} &\coloneqq \forall\xi.\|A\|_{\mathrm{DM}}
\end{aligned}
$$

Table 1.4: Computing a DM-normal formula from a standard LL formula.

Given any formula A of linear logic we associate with it two DM-normal formulas $|A|_{DM}$ and $\|A\|_{\mathrm{DM}}$ inductively as in Table 1.4: informally, $|A|_{DM}$ is obtained from A by pushing negations down to the atoms, using De Morgan laws, while $\|A\|_{\mathrm{DM}}$ is nothing but $|A^\perp|_{DM}$.

Exercise 1.3.10. Check that, for every formula A, $A \dashv\vdash |A|_{DM}$ and $A^\perp \dashv\vdash \|A\|_{\mathrm{DM}}$.

We define the *dual formula* F^* of a DM-normal formula F inductively as follows:

$$
\begin{aligned}
\alpha^* &\coloneqq \alpha^\perp & (\alpha^\perp)^* &\coloneqq \alpha \\
\mathbf{1}^* &\coloneqq \bot & \bot^* &\coloneqq \mathbf{1} \\
(A \otimes B)^* &\coloneqq A^* \parr B^* & (A \parr B)^* &\coloneqq A^* \otimes B^* \\
\top^* &\coloneqq \mathbf{0} & \mathbf{0}^* &\coloneqq \top \\
(A \oplus B)^* &\coloneqq A^* \,\&\, B^* & (A \,\&\, B)^* &\coloneqq A^* \oplus B^* \\
(!A)^* &\coloneqq ?A^* & (?A)^* &\coloneqq !A^* \\
(\forall\xi.A)^* &\coloneqq \exists\xi.A^* & (\exists\xi.A)^* &\coloneqq \forall\xi.A^* \quad .
\end{aligned}
$$

It should be clear from the definitions that:

- for each DM-normal formula F, we have $F^{**} = F$ and $F^* = |F^\perp|_{DM} = \|F\|_{\mathrm{DM}}$;
- for each formula A, we have $|A|^*_{DM} = \|A\|_{\mathrm{DM}}$ and $\|A\|^*_{\mathrm{DM}} = |A|_{DM}$.

Example 1.3.11. Let A be the formula $((\forall X.?(X\otimes X^\perp)\parr!(X^\perp\parr X))^\perp\parr(\mathbf{1}\oplus\mathbf{1}))^\perp$, we have:

$$\begin{aligned}|A|_{DM} &= (\forall X.?(X \otimes X^\perp) \parr !(X^\perp \parr X)) \otimes (\perp \& \perp)\\ |A|^*_{DM} &= |(\forall X.?(X \otimes X^\perp) \parr !(X^\perp \parr X))^\perp \parr (\mathbf{1} \oplus \mathbf{1})|_{DM}\\ &= (\exists X.!(X^\perp \parr X) \otimes ?(X \otimes X^\perp)) \parr (\mathbf{1} \oplus \mathbf{1})\end{aligned}$$

Exercise 1.3.12 (De Morgan normalization). Let us consider the following rewriting system on formulas:

$$\begin{aligned}(A^\perp)^\perp &\to A & & \\ \mathbf{1}^\perp &\to \perp & \perp^\perp &\to \mathbf{1}\\ (A\otimes B)^\perp &\to A^\perp \parr B^\perp & (A \parr B)^\perp &\to A^\perp \otimes B^\perp\\ \top^\perp &\to \mathbf{0} & \mathbf{0}^\perp &\to \top\\ (A\oplus B)^\perp &\to A^\perp \& B^\perp & (A \& B)^\perp &\to A^\perp \oplus B^\perp\\ (!A)^\perp &\to ?A^\perp & (?A)^\perp &\to !A^\perp\\ (\forall\xi.A)^\perp &\to \exists\xi.A^\perp & (\exists\xi.A)^\perp &\to \forall\xi.A^\perp \quad .\end{aligned}$$

1. Prove the strong normalization and the confluence of this system.

2. Prove that $|A|_{DM}$ is the normal form of A for this system.

As in the general case, free and bound variables and first-order substitution are defined as usual, and DM-normal formulas are always considered up to renaming of bound names. If F is a DM-normal formula, X is a second-order variable of arity n and $G[x_1,\dots,x_n]$ is a DM-normal formula with variables among x_i, then the formula $F[G/X]$ is F where every positive atom $X(t_1,\dots,t_n)$ is replaced by $G[t_1/x_1,\dots,t_n/x_n]$, and every negated atom $X(t_1,\dots,t_n)^\perp$ is replaced by $(G[t_1/x_1,\dots,t_n/x_n])^*$.

The rules of the one-sided sequent calculus are presented in Table 1.5, where every formula occurring in each sequent is assumed to be DM-normal. Note that there is no rule for negation, as this is now an involutive operator on formulas rather than a connective.

Theorem 1.3.13. *A two-sided sequent $\Gamma \vdash \Delta$ is provable (resp. provable without (cut)) by rules of Table 1.2 if and only if the sequent $\vdash \|\Gamma\|_{\mathrm{DM}}, |\Delta|_{DM}$ is provable (resp. provable without (cut)) by the rules of Table 1.5.*

Proof (Sketch). The *if*-direction is a consequence of Exercise 1.3.9 and the fact that the rules of Table 1.5 are specific instances of the right rules of Table 1.2, but for the axiom and the cut rules, which can be easily proved admissible from Table 1.2.

The *only-if*-direction can be proven by structural induction on a proof of $\Gamma \vdash \Delta$. □

Identity group

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

Multiplicative group

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

Additive group

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

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

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 2, can be seen as a quotient of one-sided sequent calculus proofs under some commutations of rules.

Beyond that point, unless we explicitly consider a two-sided calculus, we will generally identify any formula A with its DM-normal form $|A|_{DM}$, and no longer distinguish between F^* and $F^\perp$ (as already done on Table 1.5).

1.4 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 1.1):

- 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: $\mathbf{1}$, $\perp$, $\otimes$, $⅋$;
- MLL_v with constructors: X, $X^\perp$, $\otimes$, $⅋$;

- MLL_0 with constructors: $\mathbf{1}$, $\bot$, X, $X^\bot$, $\otimes$, $⅋$;
- MLL_1 with constructors: $p(t_1,\ldots,t_n)$, $p(t_1,\ldots,t_n)^\bot$, $\otimes$, $⅋$, $\forall x$, $\exists x$ — where p ranges over predicate symbols;
- MLL_{01} with constructors: $\mathbf{1}$, $\bot$, $p(t_1,\ldots,t_n)$, $p(t_1,\ldots,t_n)^\bot$, $\otimes$, $⅋$, $\forall x$, $\exists x$ — where p ranges over predicate symbols;
- MLL_2 with constructors: X, $X^\bot$, $\otimes$, $⅋$, $\forall X$, $\exists X$;
- MLL_{02} with constructors: $\mathbf{1}$, $\bot$, X, $X^\bot$, $\otimes$, $⅋$, $\forall X$, $\exists X$;
- MLL_{12} with constructors: $p(t_1,\ldots,t_n)$, $p(t_1,\ldots,t_n)^\bot$, $\otimes$, $⅋$, $\forall x$, $\exists x$, $\forall X$, $\exists X$ — where p ranges over predicate symbols and second order variables;
- MLL_{012} with constructors: $\mathbf{1}$, $\bot$, $p(t_1,\ldots,t_n)$, $p(t_1,\ldots,t_n)^\bot$, $\otimes$, $⅋$, $\forall x$, $\exists x$, $\forall X$, $\exists X$ — 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 1.5 (or Table 1.2 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 $⅋$ connectives; and the only rules are (ax), (cut), $(\otimes)$, $(⅋)$ 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 2.1
- 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 1.5 minus the quantifier group. This is the fragment for which we will provide a full proof of the admissibility of (cut), in Section 1.5.
- MELL_0: formulas are built from propositional variables (and their duals) using multiplicative connectives and units, as well as exponential modalities; and the rules are those of Table 1.5 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 1.5 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},

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):

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

  Note that $\perp \otimes ??\perp$ 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 1.7.2.1 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):

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

  Note that positive and negative formulas are disjoints classes of formulas and that $?\perp \otimes \top$ 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.

1.5 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 1.5.1 (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.

1.5.1 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 prove 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.

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

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

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

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

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

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

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

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

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

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 1.5.2 (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.

$$\dfrac{\pi_1 : \vdash C, \Gamma \quad \dfrac{}{\vdash C^\perp, C}\ (ax)}{\vdash C, \Gamma}\ (cut) \underset{\mathsf{ax}}{\longrightarrow} \pi_1 : \vdash C, \Gamma$$

Figure 1.2: Axiom case.

Definition 1.5.3 (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.

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 $\vdash \Gamma, A, \cdots, A, \Delta$ with n occurrences of A.

Definition 1.5.4 (Structural cut)**.** The following inference is called *structural cut*: $\dfrac{\vdash C^{(k)}, \Gamma \quad \vdash {C^\perp}^{(l)}, \Delta}{\vdash \Gamma, \Delta}\ (scut)$ if 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.

1.5.1.1 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.

Definition 1.5.5 (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 1.5.6 (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

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

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

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

Figure 1.3: Multiplicative key cases.

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

Figure 1.4: Additive key case.

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

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

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

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

Figure 1.5: Exponential key cases.

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

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

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

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

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

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

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

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

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

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

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

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

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

Figure 1.6: Commutation cases.

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 1.5.7. *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 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:

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

□

1.5.1.2 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.

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 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 three types: (i) one of the premises is an axiom inference, this is called an *axiom key case*, (ii) the cut formula is principal in a logical (or structural) rule in both premises of the cut, this is called a *key logical case*, (ii) or there is at least one premise in which the cut formula is not principal, this is called a *commutative case*.

Axiom key cases When a proof has the following shape:

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

- $k = 1$

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

- $k = 0$

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

- $k > 1$

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

Figure 1.7: Axiom key cases.

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, $\underset{\mathsf{ax}}{\longrightarrow}$ is defined in Figure 1.7.

Logical key cases:

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

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

We add the reduction depicted in Figure 1.3, denoted as $\underset{\otimes/⅋}{\longrightarrow}$, 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:

$$\dfrac{\dfrac{}{\vdash \mathbf{1}}\ (\mathbf{1}) \qquad \dfrac{\pi_1 : \ \vdash \Gamma}{\vdash \Gamma, \bot}\ (\bot)}{\vdash \Gamma}\ (scut) \underset{\mathbf{1}/\bot}{\longrightarrow} \pi_1 : \ \vdash \Gamma$$

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):

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

We consider the reduction depicted in Figure 1.4, denoted as $\underset{\oplus/\&}{\longrightarrow}$, 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 $\underset{\&/\oplus}{\longrightarrow}$ 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 one of the following shapes (with $l \geq 0$):

- Promotion versus dereliction:

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

- Promotion versus weakening:

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

- Promotion versus contraction:

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

We add the reduction depicted in Figure 1.8, denoted $\underset{!/?}{\longrightarrow}$, where each of the bottommost cut occurrences has been labelled and we notice the following relations about the ranks and levels of cuts:

- Promotion versus dereliction ($l = 0$):

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

- Promotion versus dereliction ($l \geq 1$):

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

- Promotion versus weakening:

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

- Promotion versus contraction:

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

Figure 1.8: Exponential key cases.

- (!) versus (?) ($l = 0$): $\mathsf{rk}(\beta) < \mathsf{rk}(\alpha)$ and $\mathsf{lvl}(\beta) = \mathsf{lvl}(\alpha) - 2$.
- (!) versus (?) ($l \geq 1$): $\mathsf{rk}(\beta) < \mathsf{rk}(\alpha)$, $\mathsf{rk}(\gamma) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\gamma) = \mathsf{lvl}(\alpha) - 1$ (but $\mathsf{lvl}(\beta)$ may be larger than the $\mathsf{lvl}(\alpha)$).
- (!) versus (w): $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\beta) = \mathsf{lvl}(\alpha) - 1$.
- (!) versus (c): $\mathsf{rk}(\beta) = \mathsf{rk}(\alpha)$, $\mathsf{lvl}(\beta) = \mathsf{lvl}(\alpha) - 1$.

The symmetrical case, $\underset{?/!}{\longrightarrow}$ is naturally considered too.

Commutative cases When none of the previous cases apply, 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:

⅋ commutation step

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

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

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

& commutation step

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

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

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

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

We notice that the resulting proof is cut-free.

Promotion commutation step

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

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

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

Dereliction commutation step

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

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

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

Structural commutation step For contraction:

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

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

For weakening:

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

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

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

1.5.1.3 $\longmapsto_{\mathsf{c}}$ is rank-decreasing

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

Lemma 1.5.8. *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.

1. Exponential key-case: we saw in the previous section that $\pi \longrightarrow \pi'$ by reducing the root cut of π into one or more cuts which are either (i) of the same rank and lower level or (ii) of lower rank and that at most one cut has its rank unchanged. If all cuts are of a smaller rank we are done, otherwise, 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 $\mathsf{rk}(\pi'') < \mathsf{rk}(\pi)$.

2. Linear key-cases: we saw in the previous section that the rank decrease by one step of $\longrightarrow$.

3. 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 $\mathsf{rk}(pi') = \mathsf{rk}(\pi_1) < \mathsf{rk}(\pi)$.

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 $\pi_{01}, \ldots, \pi_{0k}$ rooted in those cuts of rank equating $\mathsf{rk}(\pi)$ leading proofs $\pi'_{01}, \ldots, \pi'_{0k}$ 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 $\mathsf{rk}(\pi') < \mathsf{rk}(\pi)$.

□

1.5.2 Consequences

Cut elimination has several important consequences:

Definition 1.5.9 (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,
- $\mathbf{1}$, $\perp$, $\mathbf{0}$, $\top$ 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).

Theorem 1.5.10 (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.

Theorem 1.5.11 (Consistency)**.** *The empty sequent* $\vdash$ *is not provable. Subsequently, it is impossible to prove both a formula* A *and its negation* $A^\perp$*; it is impossible to prove* $\mathbf{0}$ *or* $\perp$*.*

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 $\vdash A^\perp$ 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, $\perp$ and $\mathbf{0}$ are not, since they are equivalent to $\mathbf{1}^\perp$ and $\top^\perp$ respectively. □

1.6 Reversibility and focusing

As already seen in Section 1.5.2, 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.

1.6.1 Reversibility

Definition 1.6.1 (Reversibility)**.** A connective c is called *reversible* if for every proof $\pi : \ \vdash \Gamma, c(A_1, \ldots, A_n)$, there is a proof π' with the same conclusion in which $c(A_1, \ldots, A_n)$ is introduced by the last rule (*i.e.* principal).

Remark 1.6.2. Now that we have Theorem 1.5.1, we can refine a bit Definition 1.2.2 with the following implications:

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

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

Lemma 1.6.3. *The following* reversed rules *are derivable with cuts:*

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

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

Proof. Derivability results from the following proof schema (which relies on considering proofs from Table 1.3 and removing their last rule):

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

- $$\dfrac{\dfrac{}{\vdash 1}\ (\mathbf{1}) \qquad \vdash \bot, \Gamma}{\vdash \Gamma}\ (cut)$$

- $$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\ (ax)}{\vdash A, A^\perp \oplus B^\perp}\ (\oplus_1) \qquad \vdash A \mathbin{\&} B, \Gamma}{\vdash A, \Gamma}\ (cut)$$

- $$\dfrac{\dfrac{\dfrac{}{\vdash B, B^\perp}\ (ax)}{\vdash B, A^\perp \oplus B^\perp}\ (\oplus_2) \qquad \vdash A \mathbin{\&} B, \Gamma}{\vdash B, \Gamma}\ (cut)$$

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

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

□

Theorem 1.6.4. *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 1.6.3 followed by the introduction rule of c to the proof of $\vdash \Gamma, c(A_1, \ldots, A_n)$:

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

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

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 ?. □

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:

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

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.

1.6.2 Generalized connectives and rules

Definition 1.6.5. 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 $\parr$, &, $\bot$, $\top$.

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 $N[X_1, X_2, X_3] = (X_1 \parr X_2) \parr (X_2 \mathbin{\&} X_3)$, we can derive an introduction rule for N as

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

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:

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

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 1.6.6. 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 1.6.7. 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

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

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

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

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

Theorem 1.6.8. *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.

1.6.3 Focusing

Definition 1.6.9. 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 ⅋, &, $\bot$, $\top$. 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 1.6.10. A proof $\pi : \vdash \Gamma, A$ is said to be *positively focused on* A if it has the shape

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

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.

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 1.6.11. *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 on 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

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

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

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

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.

1.7 Variations

1.7.1 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

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

and a *digging* rule

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

without modifying the provability. Note that digging violates the subformula property.

In presence of the digging rule

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

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.

1.7.2 Non-symmetric sequents

The same remarks that lead to the definition of the one-sided calculus can lead 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).
- Similar restrictions are used in various semantics and proof search formalisms.

1.7.2.1 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 ($\_^{\perp}$, $⅋$, $\perp$ 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 1.6. For each connective of ILL, they are obtained from the corresponding rules of Table 1.2 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 ILL rules can nevertheless be obtained by restricting the following derivations to intuitionistic sequents:

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta}{\Gamma, A^{\perp} \vdash \Delta}\,(n_L) \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A^{\perp} ⅋ B \vdash \Delta, \Delta'}\,(⅋_L) \qquad\qquad \dfrac{\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A^{\perp}, B, \Delta}\,(n_R)}{\Gamma \vdash A^{\perp} ⅋ B, \Delta}\,(⅋_R)$$

1.7.3 Mix rules

It is quite common to consider mix rules:

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

1.7.4 Dyadic sequent calculus

$\vdash \Gamma \mid \Delta \Leftrightarrow \; \vdash ?\Gamma, \Delta$

Identity group

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

Multiplicative group

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

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

Additive group

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

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

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.

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

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

Exponential group

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

Structural rules

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

Table 1.6: Inference rules for Intuitionistic Linear Logic sequent calculus

Chapter 2

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 first two sections, about multiplicative proof nets, are currently in a satisfactory state, although some introductory material and pictures are missing. The treatment of* MELL *is only roughly sketched.*

2.1 Multiplicative Proof Nets

We first consider multiplicative proof nets without units, *i.e.* proof nets for the MLL_v fragment (see Section 1.4).

2.1.1 Proof structures

2.1.1.1 Definition of proof structures

Definitions and abstract properties of graphs we are going to use can be found in Appendix **??**.

An *proof structure!multiplicative* $\mathcal{S}$ is the data of:

- a directed acyclic graph $\mathcal{G}_\mathcal{S}$ — the incoming arrows of a node are called its *premise*, the outgoing arrows are its *conclusion*;
- a labelling of the nodes of $\mathcal{G}_\mathcal{S}$ with labels in $\{ax, cut, \otimes, ⅋, \bullet\}$ — 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 ⅋-node has exactly two premises and one conclusion;
  - each $\bullet$-node has exactly one premise and no conclusion;

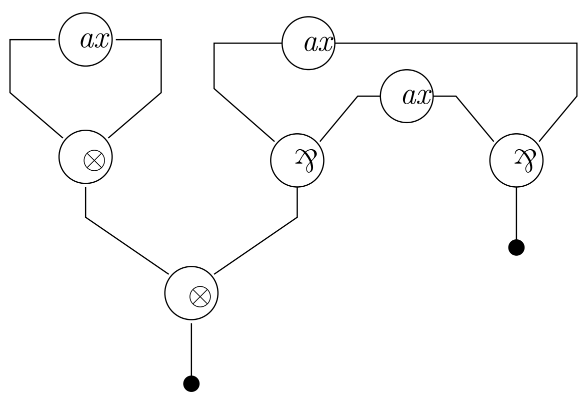
- an ordering of the premises of each $\otimes$-node and of each $⅋$-node — the first premise is called the left premise of the node, and the second one is the right premise.

A premise of a $\bullet$-node is called a *conclusion edge* of the proof structure.

Nodes with all their conclusions connected to $\bullet$-nodes are called *terminal node* (in particular *cut* nodes are always terminal). Nodes which are not $\bullet$-nodes are called *internal node* of the proof structure: note that internal nodes can be terminal. A *premise node* of some node N is any internal node N_0 such that a conclusion of N_0 is also a premise of N. By definition a non-empty proof structure must contain at least one *ax* node and at least one terminal node.

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 in 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 $\bullet$-nodes which are simply represented as bullets.

Example 2.1.1 (Untyped Proof Structure). Consider the proof structure



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

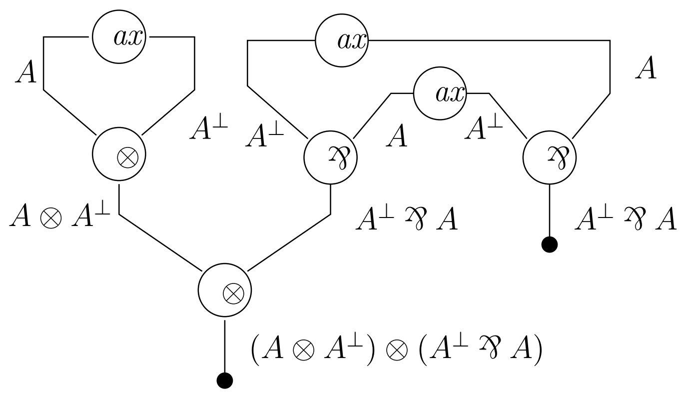
A *typing of a proof structure* of a proof structure is a labelling of its arrows with formulas of MLL_v — the label of an arrow is called its type — such that:

- 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 $⅋$-node) has type A and its right premise has type B then its conclusion has type $A \otimes B$ (resp. $A ⅋ B$).

A *typed proof structure* is the data of an untyped proof structure together with such a typing. Given an enumeration of the conclusions of a typed proof structure $\mathcal{S}$, the *conclusion sequent of a proof structure* of $\mathcal{S}$ is the sequence of the types of its conclusions. Note that the empty graph is a typed proof structure, whose conclusion is the empty sequent $\vdash$.

Example 2.1.2 (Typed Proof Structure). We can type the previous example of proof structure as follows



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

Remark 2.1.3. In presence of typing, the directed acyclicity requirement 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 $\parr$: in this case the definition imposes that premises are typed with an immediate subformula of the conclusion.

In the more general setting 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 to be introduced in the next section.

Given an untyped proof structure, directed acyclicity ensures that types can be inferred from top to bottom: the data of an untyped proof structure $\mathcal{S}$, together with a typing of its axioms with dual formulas, induces at most one proof structure (and if $\mathcal{S}$ is cut-free, such a proof structure always exists).

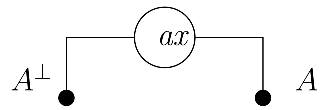
In the following, we will simply call *proof structure* any untyped proof structure, that might or might not be associated with a typed proof structure.

2.1.1.2 Translation of proof trees into typed proof structures

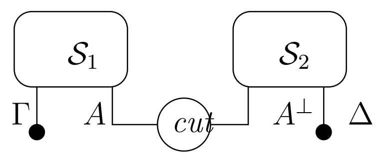
A proof π of the sequent calculus MLL_v can be translated into a typed proof structure $\mathtt{ps}(\pi)$ with the same conclusion. There is a bijection between internal nodes of $\mathtt{ps}(\pi)$ and the rules of π which are not exchange rules, in such a way that each node is labelled with the name of its corresponding rule.

The translation is defined by induction on the structure of the proof:

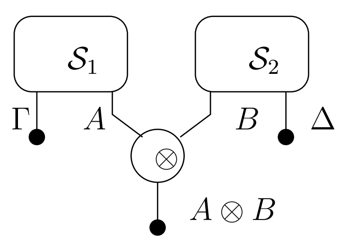
- An (ax) rule $\dfrac{}{\vdash A^{\perp}, A}\ (ax)$ is translated into an ax-node with conclusions labelled $A^{\perp}$ and A which have $\bullet$-nodes as targets.



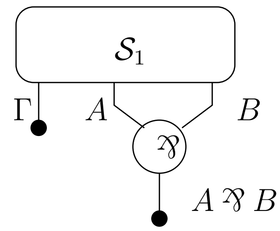
- If π_1 is translated into $\mathcal{S}_1 = \texttt{ps}(\pi_1)$ and π_2 is translated into $\mathcal{S}_2 = \texttt{ps}(\pi_2)$, then with the proof $\dfrac{\overset{\pi_1}{\vdash \Gamma, A} \qquad \overset{\pi_2}{\vdash A^\perp, \Delta}}{\vdash \Gamma, \Delta}\ (cut)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1$ and $\mathcal{S}_2$ 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.



- If π_1 is translated into $\mathcal{S}_1 = \texttt{ps}(\pi_1)$ and π_2 is translated into $\mathcal{S}_2 = \texttt{ps}(\pi_2)$, then with the proof $\dfrac{\overset{\pi_1}{\vdash \Gamma, A} \qquad \overset{\pi_2}{\vdash \Delta, B}}{\vdash \Gamma, \Delta, A \otimes B}\ (\otimes)$ we associate the typed proof structure $\mathcal{S}$ obtained from $\mathcal{S}_1$ and $\mathcal{S}_2$ 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 $\mathcal{S}_1 = \texttt{ps}(\pi_1)$, then to the proof $\dfrac{\overset{\pi_1}{\vdash \Gamma, A, B}}{\vdash \Gamma, A \parr B}\ (\parr)$ 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.



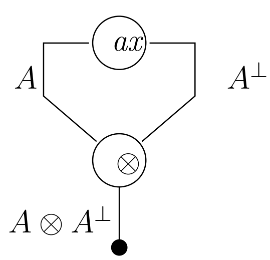
It is then natural to try to analyse the kernel of the translation `ps` by understanding when two different sequent calculus proofs are mapped to the same typed proof structure. One can prove that it is the case if and only if one can transform one of the two proofs to the other by some permutations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

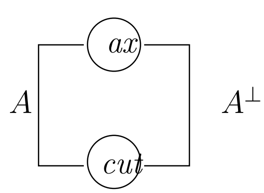
In a cut-free sequent calculus proof or typed proof structure, the formulas used in the *ax* rules or nodes are occurrences of sub-formulas of the conclusions of the proof or the typed proof structure. Two proofs are mapped to the same typed proof structure if and only if the pairing of such occurrences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

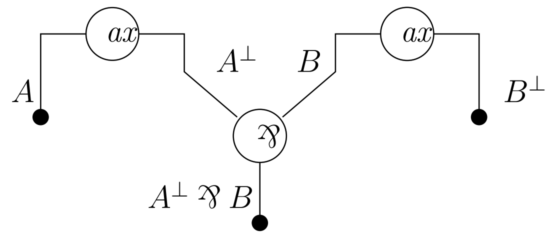
2.1.2 Proof nets

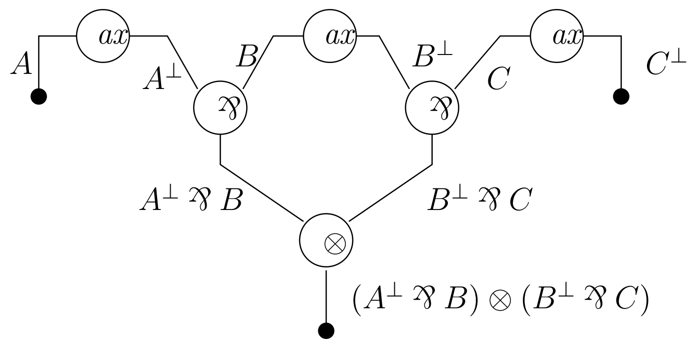
Not all typed proof structures represent (or are the translation of) proofs in the sequent calculus MLL_v. This leads to the study of *correctness criteria* to try to delineate a sub-set of "valid" proof structures which belong to the image of the translation `ps`.

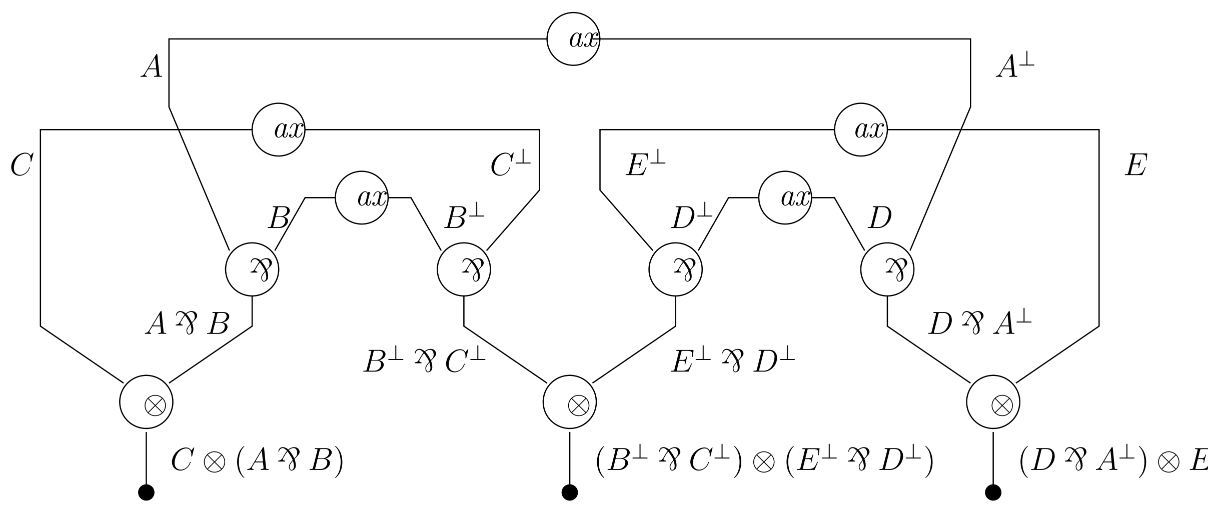
Example 2.1.4. Here are a few examples of typed proof structures which do not correspond to any proof of MLL_v:











2.1.2.1 Switching graphs

Given a proof structure $\mathcal{S}$, let $\mathcal{N}_{\parr}(\mathcal{S})$ be the set of its $\parr$-nodes. A *switching* of $\mathcal{S}$ is a function φ defined on $\mathcal{N}_{\parr}(\mathcal{S})$ and such that, for each $\parr$-node P, $\varphi(P)$ is one of its premises. The *switching graph* $\mathcal{S}^{\varphi}$ associated with φ is the graph obtained from $\mathcal{S}$ by keeping only the premise $\varphi(P)$ for each $\parr$-node P: formally, we modify the target of the other premise into a new node $P^{\bullet}$, as in Fig. 2.1 — where we depict nodes with the labels inherited from $\mathcal{S}$, and the new node $P^{\bullet}$

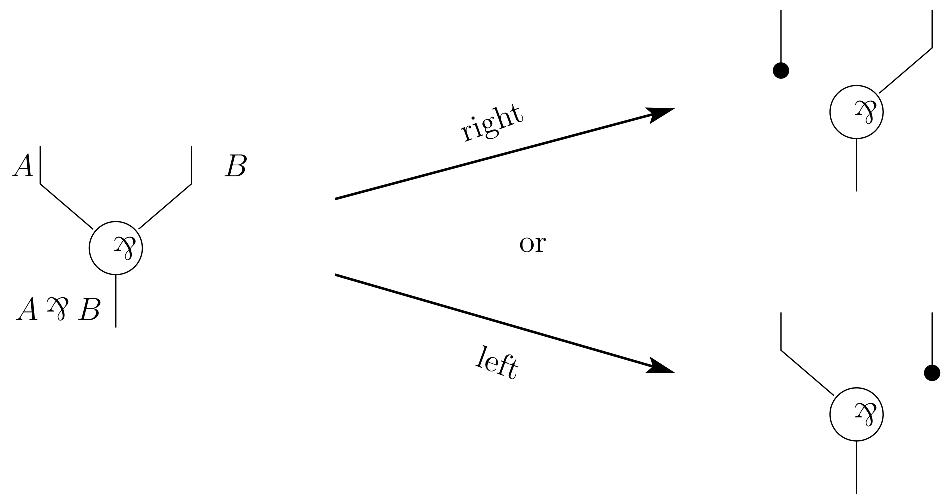


Figure 2.1: Switching a ⅋-node.

is depicted as a •-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 s in $\mathcal{S}^\varphi$ is the same as in $\mathcal{S}$;
- the target function t in $\mathcal{S}^\varphi$ is the same as in $\mathcal{S}$, except each premise a of a ⅋-node P with $a \neq \varphi(P)$ is 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 to define switching graphs. A proof structure with p ⅋-nodes induces 2^p switchings and thus (up to) 2^p switching graphs.

Definition 2.1.5 (Proof nets)**.** A proof structure is *acyclic proof structure* if its switching graphs do not contain any undirected cycle. An acyclic proof structure is called a *proof net!multiplicative*.

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.

Lemma 2.1.6 (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 **??**, 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.

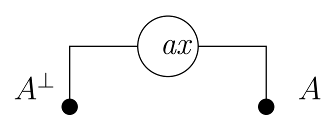
Exercise 2.1.7. Inspect all possible switchings of the proof structures of Example 2.1.4. Which of these structures are proof nets? For the latter, check that no switching graph is connected.

2.1.2.2 Soundness

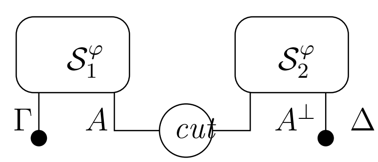
Proposition 2.1.8 (Soundness of Correctness)**.** *The translation* $\texttt{ps}(\pi)$ *of a sequent calculus proof* π *of* MLL_v *is a typed connected proof net.*

Proof. By definition, the translation $\texttt{ps}(\pi)$ of a sequent calculus proof π of MLL_v is a typed connected proof structure: it remains only to check that the underlying proof structure is acyclic (it 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 undirected cycle.

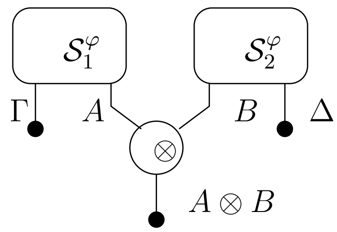


- If π is obtained from π_1 and π_2 with a (cut) rule, every switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ is obtained by connecting through a *cut*-node a switching graph $\mathcal{S}_1^\varphi$ of $\mathcal{S}_1$ and a switching graph $\mathcal{S}_2^\varphi$ of $\mathcal{S}_2$.



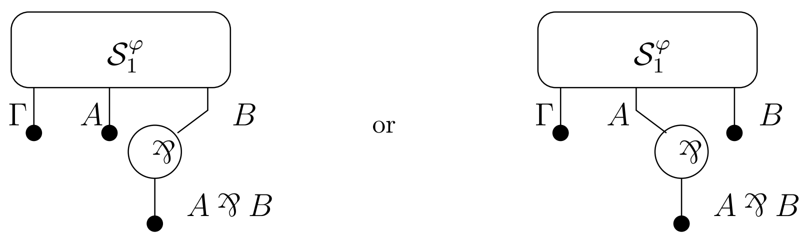
We can deduce that no switching graph of $\mathcal{S}$ contains an undirected cycle.

- 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 an undirected 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 an undirected cycle.

□

We will establish the converse of this property in Section 2.1.4: 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.

2.1.3 Cut Elimination

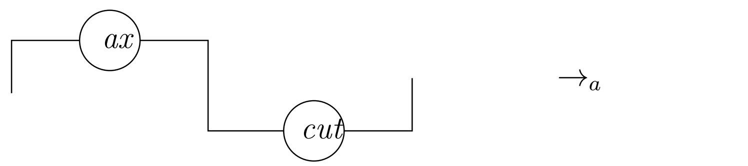
If we propose proof nets as an alternative to sequent calculus to study proofs in (multiplicative) linear logic, we need to be able to deal with cut elimination in this new syntax without referring to the sequent calculus.

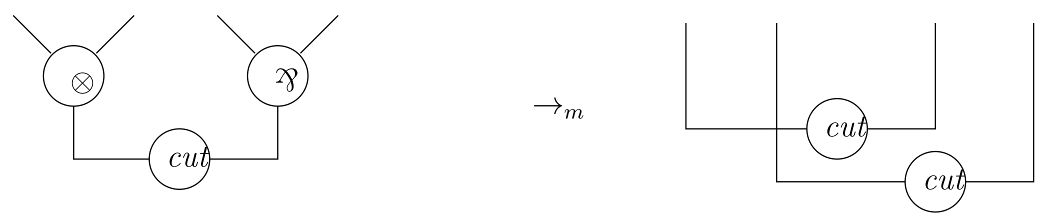
Cut elimination in proof nets is defined as a graph rewriting procedure, which acts through local transformations of the proof net.

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

2.1.3.1 Reductions Steps

We consider two reductions steps:





In the left hand side of the $\to_a$-step, we require that there is no directed path from a to a', where a is the arrow on the left (with an unknown target) and a' is the arrow on the right (with an unknown source): this condition is always verified in the case in a proof net.

More precisely, notice that:

1. if $\mathcal{R}$ is a proof structure which is not a proof net, one might have a cut having as premise a conclusion of an *ax* node which does not belong to any redex (see e.g. the "loop" made of an axiom with a cut between its conclusions, as in Example 2.1.4);

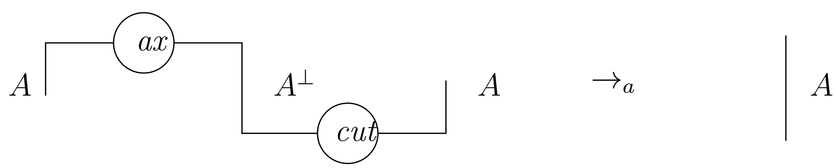
2. if $\mathcal{R}$ is a proof net, any cut having as premise a conclusion of an *ax* node always belongs to a redex;

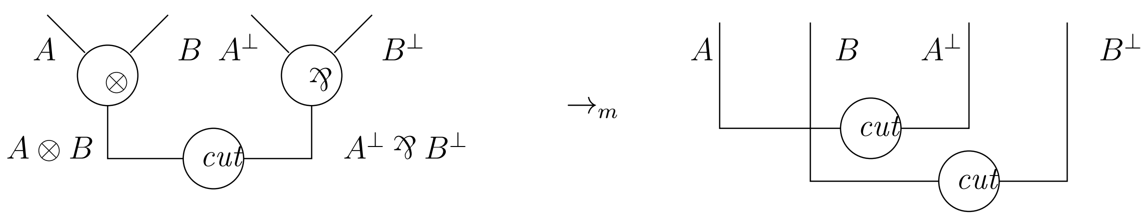
3. if $\mathcal{R}$ is a proof net that cannot be typed, one might have a cut which does not belong to any redex (e.g. a cut between two $\otimes$-nodes);

4. if $\mathcal{R}$ is a proof net that can be typed, every cut belongs to a redex: by Point 2 we only have to check this is the case when both premises of the cut are conclusions of a $\otimes$-node or a $⅋$-node. Just observe that in this

case, due to the typing constraints, the premises of the cut cannot be both conclusions of a $\otimes$-node (resp. $⅋$-node). As a consequence, normal forms for the reduction of typed proof nets are exactly cut-free typed proof nets.

Notice also that applying these reduction steps inside a typed structure preserves typing:





Examples TODO

2.1.3.2 Preservation of Correctness

Lemma 2.1.9 (Preservation of Acyclicity)**.** *If $\mathcal{R}$ is a proof net and $\mathcal{R} \to \mathcal{R}'$ 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 (we use here in a crucial way the condition required to apply the $\to_a$-step). Then one of these two switching graphs is acyclic if and only if the other one is.
- Through an m step, a switching graph $\mathcal{S}$ of the reduct gives rise to two switching graphs $\mathcal{S}_1$ and $\mathcal{S}_2$ in the redex depending on the choice of a

premise a_1 or a_2 of the ⅋ node which disappears through the reduction. Assume there is a cycle in $\mathcal{S}$. It must go through at least one of the cuts otherwise it is a cycle in $\mathcal{S}_1$ and $\mathcal{S}_2$.

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

If it uses both cuts: either the premises of the $\otimes$-node are connected in $\mathcal{S}$ (without using those cuts) and we have a cycle in both $\mathcal{S}_1$ and $\mathcal{S}_2$; or a_1 is connected to a premise of the $\otimes$-node, and we have a cycle in $\mathcal{S}_1$.

□

Remember that, thanks to Lemma 2.1.6, all the switching graphs of a proof net have the same number of connected components.

Lemma 2.1.10 (Preservation of Connected Components)**.** *If $\mathcal{R}$ is a proof net and $\mathcal{R} \to \mathcal{R}'$ 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 2.1.9). We can thus use Lemma **??**. We consider the two reduction steps. In each case, in every switching graph we loose two nodes and two edges thus the number of connected components is not modified. □

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

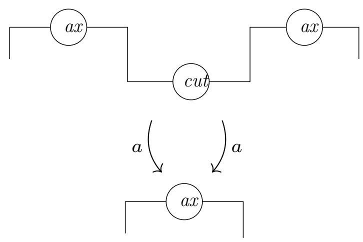
2.1.3.3 Properties

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 few 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.

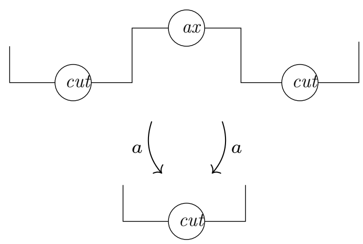
Lemma 2.1.11 (Sub-Confluence)**.** *The reduction of proof structures is sub-confluent.*

Proof. There are two kinds of critical pairs:

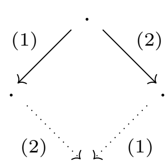
- a/a (shared *cut*)



- a/a (shared *ax*)



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



since they cannot overlap. □

Proposition 2.1.12 (Convergence)**.** *The reduction of proof structures is convergent.*

Proof. Confluence is obtained by Proposition **??** and Lemma 2.1.11. Moreover, the number of nodes is reduced in each reduction step. □

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

Corollary 2.1.13 (Normalisation 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.*

2.1.4 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 tree structure of sequent calculus proofs.

2.1.4.1 Sequential structures

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

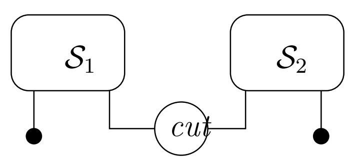
- each node labelled *hyp* has an arbitrary number of conclusions (at least one) and no premise;
- in a typing for a proof structure with *hyp*-nodes, we require that for each *hyp*-node with conclusions $A_1, \ldots, A_n$, the sequent $\vdash A_1, \ldots, A_n$ is derivable in MLL_v.

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.

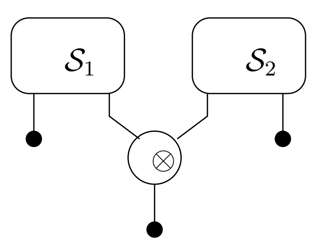
In fact, some form of sequentiality can be recovered from the correctness criterion, even without typing. We say that a proof structure $\mathcal{S}$ is *sequential structure* (resp. *connected sequential structure*) if one of the following holds (resp. if one of (S1) to (S4) holds), assuming inductively that $\mathcal{S}_1$ and $\mathcal{S}_2$ are sequential (resp. connected sequential):

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

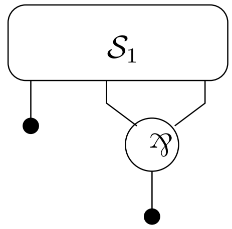
(S2) $\mathcal{S}$ is obtained from $\mathcal{S}_1$ and $\mathcal{S}_2$ by removing one $\bullet$-node in each of $\mathcal{S}_1$ and $\mathcal{S}_2$, with premises a_1 in $\mathcal{S}_1$ and a_2 in $\mathcal{S}_2$, and by introducing a new *cut*-node with premises a_1 and a_2;



(S3) $\mathcal{S}$ is obtained from $\mathcal{S}_1$ and $\mathcal{S}_2$ 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;



(S4) $\mathcal{S}$ is obtained from $\mathcal{S}_1$ by removing two $\bullet$-nodes in $\mathcal{S}_1$ and $\mathcal{S}_2$, with premises a_1 in $\mathcal{S}_1$, and by introducing a new terminal $\parr$-node with premises a_1 and a_2;



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

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

Note that cases (S1) to (S4) match exactly the structure of the proof of Proposition 2.1.8.

Lemma 2.1.14 (Untyped correctness)**.** *Any sequential (resp. connected sequential) proof structure is a proof net (resp. a connected proof net).*

Proof. By induction on the definition of sequential (resp. connected sequential) proof structures: cases (S1) to (S4) match exactly the structure of the proof of Proposition 2.1.8; case (S5) is trivial, and (S6) follow straightforwardly from the induction hypotheses. □

In the following, we will establish the converse, first in the connected case: every connected proof net is connected sequential. We will deduce the sequentialization theorem (Theorem 2.1.23) from the fact that a typed proof structure is the translation of a proof as soon as it is connected sequential. We will also deduce that a proof structure is sequential iff it is a proof net (Theorem 2.2.15), which will be useful later.

Lemma 2.1.15. *Let $\mathcal{S}$ be a sequential proof structure. Then $\mathcal{S}$ is connected sequential iff one of its switching graphs is connected.*

Proof. We have already noted that if $\mathcal{S}$ is connected sequential then $\mathcal{S}$ is a connected proof net. Conversely, assume that $\mathcal{S}$ is sequential and one of its switching graphs is connected. We show that $\mathcal{S}$ is connected sequential by induction on sequential structures. Cases (S1) to (S4) are straightforward by induction hypothesis. Case (S5) does not apply because $\mathcal{S}$ is not empty. Moreover, to apply (S6), one of $\mathcal{S}_1$ or $\mathcal{S}_2$ must be empty, otherwise the switching graphs of $\mathcal{S}$ must have at least two components: then $\mathcal{S} = \mathcal{S}_1$ or $\mathcal{S} = \mathcal{S}_2$ and we conclude directly by induction hypothesis. □

To establish the correspondence between sequential structures and 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 2.1.16. *For any proof structure $\mathcal{S}$:*

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

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

- the $⅋$-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;
- a cycle in $\mathcal{S}[\otimes/cut]^\varphi$ cannot cross the conclusions of the newly introduced terminal $\otimes$-nodes, hence it is also a cycle in $\mathcal{S}^\varphi$.

□

2.1.4.2 Switching paths

Recall that any switching graph $\mathcal{S}^\varphi$ of $\mathcal{S}$ has an additional node $P^\bullet$ for each $⅋$-node P, and the same edges as $\mathcal{S}$, except that the target of the premise $\varphi(P)$ of each $⅋$-node P is changed to $P^\bullet$. In particular, any path γ of $\mathcal{S}^\varphi$ defines a path $\varphi^{-1}(\gamma)$ of $\mathcal{S}$ with the same sequence of edges, and visiting the same nodes, except maybe at endpoints: if the source (resp. target) of γ is $P^\bullet$ for some $⅋$-node P, then the source (resp. target) of $\varphi^{-1}(\gamma)$ is P. Moreover γ is simple iff $\varphi^{-1}(\gamma)$ is. We call *switching path* of $\mathcal{S}$ any such simple path $\varphi^{-1}(\gamma)$, for any switching φ.

Given a path γ, a *blocking* $⅋$ of γ is any $⅋$-node P such that $a_1^+a_2^-$ is a subpath of γ, where a_1 and a_2 are both premises of P. It should be clear that a path with a blocking $⅋$ cannot be a switching path. Conversely:

Lemma 2.1.17 (Non switching path)**.** *A simple path γ of $\mathcal{S}$ is a switching path unless it admits a blocking $⅋$.*

Proof. If there is no blocking $⅋$, we construct a switching φ such that γ is of the shape $\varphi^{-1}(\gamma_0)$. For each $⅋$-node P with premises a_1 and a_2, and conclusion a_0:

- if γ does not cross a_1, or if a_1^- is the first edge of γ, or if a_1^+ is the last edge of γ, then we can set $\varphi(P) \coloneqq a_2$ — and this is sufficient ensure that $\mathsf{s}(a_1)$, $\mathsf{s}(a_2)$ and $\mathsf{t}(a_2)$ are the same in $\mathcal{G}_\mathcal{S}$ and in $\mathcal{S}^\varphi$;
- if γ does not cross a_2, or if a_2^- is the first edge of γ, or if a_2^+ is the last edge of γ, then we can set $\varphi(P) \coloneqq a_1$ — and this is sufficient ensure that $\mathsf{s}(a_2)$, $\mathsf{s}(a_1)$ and $\mathsf{t}(a_1)$ are the same in $\mathcal{G}_\mathcal{S}$ and in $\mathcal{S}^\varphi$.

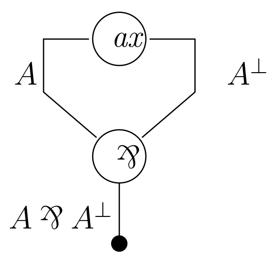
If each $⅋$-node occurring in γ falls in one of these two cases, we obtain γ_0 with the same sequence of edges as γ — but maybe with different endpoints.

It suffices to check that this covers all possible cases. Indeed, if γ crosses both premises of a $⅋$-node P, we write a_1 and a_2 for those premises so that γ crosses a_1 before a_2 and we observe that:

- either γ contains a subpath of the shape $a_1^+\gamma' a_2^-$, which is forbidden because γ' should either cross a_0 twice, or be empty;
- or γ contains a subpath of the shape $a_1^+\gamma' a_2^+$, and then γ' must have a_0^+ as first edge and a_2^+ must be the last edge of γ because γ is simple;
- or γ contains a subpath of the shape $a_1^-\gamma' a_2^-$ and then we can apply the previous case to $\overline{\gamma}$;
- or γ contains a subpath of the shape $a_1^-\gamma' a_2^+$, and then either a_1^- is the first edge of γ or the previous edge in γ is a_0^-, and then a_2^+ is the last edge of γ, because γ is simple.

□

Observe that there might be cyclic switching paths, although there is no cycle in a switching graph: both cyclic paths between the premises of the ⅋-node in the proof net



are indeed switching paths. We generalize this as follows: a *proper cycle* of a ⅋-node P is any switching path of the shape $a_1^-\gamma' a_2^+$ where a_1 and a_2 are both premises of the same ⅋-node.

Notice that the two premises of a ⅋-node which has no proper cycle are disconnected in every switching graph. Hence, every switching graph of a proof net containing a ⅋-node which has no proper cycle has at least two connected components (and thus the proof net is not connected).

Lemma 2.1.18 (Cyclic Switching Paths)**.** *In a proof net, every cyclic switching path is a proper cycle of some ⅋-node.*

Proof. Assume $\gamma = \varphi^{-1}(\gamma_0)$ is a switching path in a proof net $\mathcal{S}$, with $\mathsf{s}(\gamma) = \mathsf{t}(\gamma)$. It will be sufficient to show that the first edge of γ is a^- with a a premise of a ⅋-node: then we can apply the same argument to $\overline{\gamma}$.

If $\mathsf{s}(\gamma)$ is not a ⅋-node, then γ has the same endpoints as γ_0, which is thus a cycle in $\mathcal{S}^\varphi$: this is forbidden since $\mathcal{S}$ is a proof net. So $\mathsf{s}(\gamma)$ must be a ⅋-node P, with conclusion a_0: it suffices to show that a_0^+ is not the first edge of γ.

Indeed, we could otherwise write $\gamma = a_0^+\gamma' a_1^+$ with a_1 a premise of P. Then γ would be a cycle in $\mathcal{S}^{\varphi'}$, defining φ' by $\varphi'(P) := a_1$ and $\varphi'(P') := \varphi(P')$ for each ⅋-node $P' \neq P$. □

Given a cycle γ in a proof net, one of the following must thus hold:

- γ has no blocking $⅋$, hence it is a switching path and it must be a proper cycle of some $⅋$-node;
- γ has exactly one blocking $⅋$-node P, and a cyclic permutation of the edges of γ yields a proper cycle of P: in this case, writing N for the source and target of γ, we say P *clasping* $⅋$*-node* N and write $P \prec N$;
- γ has at least two blocking $⅋$-nodes, or it is of the shape $a_1^- \gamma' a_2^+$ where γ' has at least one blocking $⅋$ and a_1 and a_2 are both premises of a $⅋$-node — *i.e.* no cyclic permutation of the edges of γ yield a switching path.

A *strong switching path* is a switching path whose first edge (if any) is not a^- with a a premise of a $⅋$-node. In particular, the previous Lemma entails that there is no cyclic strong path in a proof net.

Lemma 2.1.19 (Concatenation of Switching Paths)**.** *If γ is a switching path and γ' is a strong switching path with $\mathsf{t}(\gamma) = \mathsf{s}(\gamma')$, and if γ and γ' are disjoint then their concatenation $\gamma\gamma'$ is a switching path. If moreover γ is strong, then $\gamma\gamma'$ is strong as well.*

Proof. By hypotheses, the path $\gamma\gamma'$ is a simple undirected path of $\mathcal{S}$. If it is not a switching path, then the previous lemma gives a $⅋$-node P with premises a_1 and a_2 such that $\gamma\gamma'$ contains $a_1^+ a_2^-$. Since γ and γ' are both switching paths, none can contain such a subpath. Then a_2^- must be the first edge of γ', which contradicts the fact that it is strong.

If moreover γ is strong, then either it is empty and $\gamma\gamma' = \gamma'$ is strong, or the first edge of $\gamma\gamma'$ is the first edge of γ and $\gamma\gamma'$ is strong. □

Note that in general, if a_0 is the conclusion of a $⅋$-node or $\otimes$-node, and a_1 is one of its premises, there may exist non disjoint switching paths $\gamma_0 = a_0^+ \gamma_0'$ and $\gamma_1 = a_1^- \gamma_1'$, even in a proof net: for instance, in the net associated with the cut free proof of $X^\perp, (X \otimes Y) ⅋ Y^\perp$, we can consider the paths from the $\otimes$-node to the conclusion of the $⅋$-node. However, this never occurs if γ_1 is a proper cycle of a $⅋$-node:

Lemma 2.1.20. *Let π be a proper cycle of a $⅋$-node P in a proof net $\mathcal{R}$, and $\gamma = a_0^+ \gamma'$ be a switching path of $\mathcal{R}$, where a_0 is the conclusion of P. Then $\mathsf{t}(\gamma)$ does not occur in π and γ is disjoint from π.*

Proof. Observe that π does not cross a_0: otherwise, we obtain a suffix of π or of $\overline{\pi}$ of the shape $a_0^+ \pi' a_1^+$ for some premise a_1 of P, which is a strong cyclic switching path, thus violating Lemma 2.1.18.

We first prove that $\mathsf{t}(\gamma)$ does not occur in π. Otherwise, we consider the longest prefix δ of γ that is disjoint from π: either $\delta = \gamma$ or there is some edge e such that δe is also a prefix of γ and e or $\overline{e}$ occurs in π. In both cases, $\mathsf{t}(\delta)$ occurs in π, and $\delta = a_0^+ \delta'$ for some prefix δ' of γ'. Then we can write $\pi = \overline{\pi_1} \pi_2$ with $\mathsf{t}(\pi_1) = \mathsf{t}(\pi_2) = P$ and $\mathsf{s}(\pi_1) = \mathsf{s}(\pi_2) = \mathsf{t}(\delta)$. Applying Lemma 2.1.19, we thus obtain two cyclic switching paths $\pi_1\delta$ and $\pi_2\delta$. Applying Lemma 2.1.18 to these two paths, we obtain that $\mathtt{t}(\delta)$ is a ⅋-node, with premises a_0 and a_1, so that both π_1 or π_2 have a_1^- as first edge: this contradicts the fact that π is a simple path.

We deduce that γ is disjoint from π: otherwise, we can apply the previous result to the longest prefix δ of γ that is disjoint from π. □

2.1.4.3 Sequentialization of connected proof nets

For every $\otimes$-node T of a proof net $\mathcal{R}$, exactly one of the following holds:

1. no cycle of $\mathcal{R}$ contains T;
2. T belongs to some switching cycle of $\mathcal{R}$ and thus (by Lemma 2.1.18) T occurs in the proper cycle of some ⅋-node P, hence $P \prec T$;
3. T belongs to some simple cycle of $\mathcal{R}$, but none of the simple cycles of $\mathcal{R}$ containing T is switching. In this case, for every simple cycle γ of $\mathcal{R}$ with source T, by Lemma 2.1.17 there is a blocking ⅋-node of γ and, since no switching cycle of $\mathcal{R}$ contains T, γ has to contain another blocking ⅋-node. Thus T is the source of at least one simple cycle and every simple cycle with source T contains at least two blocking ⅋-nodes.

In case 1, the $\otimes$-node T is called *splitting*, since removing it splits its connected component in three: in other words, writing a_1 and a_2 for the premises of T and a_0 for its conclusion, there is no path $a_1^-\gamma a_2^+$ (and thus no path $a_2^-\gamma a_1^+$), there is no path $a_1^-\gamma a_0^-$ (and thus no path $a_0^+\gamma a_1^+$) and there is no path $a_2^-\gamma a_0^-$ (and thus no path $a_0^+\gamma a_2^+$) in $\mathcal{R}$. In the particular case of a terminal splitting $\otimes$-node, its connected component is actually split in two (if we ignore the conclusion of the node and of the proof net).

Case 3 never occurs in a connected proof net:

Lemma 2.1.21 (Non-splitting $\otimes$ in a connected proof net)**.** *For each $\otimes$-node T of a connected proof net $\mathcal{R}$, if T is not splitting then there is a ⅋-node P with $P \prec T$.*

Proof. Fix a connected proof net $\mathcal{R}$ and a non-splitting $\otimes$-node T in $\mathcal{R}$. We write e_1, e_2 and e_3 for the three edges such that $\mathtt{t}(e_j) = T$ for $1 \leq j \leq 3$. Consider a cyclic path γ with source T: for each ⅋-node P occurring in γ, either γ crosses exactly one premise of P together with its conclusion, or P is blocking for γ. We fix a switching φ of $\mathcal{R}$ selecting the premise of P crossed by γ when there is exactly one, leaving the other values unspecified.

Since γ is a cycle in a proof net, it must have at least one blocking ⅋. Thus we can write $\gamma = \gamma_0 \cdots \gamma_{n+1}$ uniquely so that the following holds:

- the first edge of γ_0 is $\overline{e_{j_1}}$ and the last edge of γ_{n+1} is e_{j_2} for some $j_1 \neq j_2$;
- for $0 \leq i \leq n$, the last edge of γ_i is $a_{i,1}^+$ and the first edge of γ_{i+1} is $a_{i,2}^-$, where $a_{i,1}$ and $a_{i,2}$ are the premises of a blocking ⅋-node P_i;

- each blocking ⅋ of γ is some P_i.

In particular, each γ_i induces a path γ'_i in $\mathcal{R}^\varphi$ with the same edges.

By acyclicity and connectedness of $\mathcal{R}^\varphi$, removing T from $\mathcal{R}^\varphi$ (and replacing it with two new •-nodes as targets for its premises and an *hyp*-node as a source for its conclusion) defines exactly three connected components $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$, each $\mathcal{G}_j$ containing e_j for $1 \leq j \leq 3$. Moreover, for $0 \leq i < n$, $a_{i,2}$ and $a_{i+1,1}$ are connected *via* γ'_{i+1}, hence they belong to the same component; similarly, $a_{0,1}$ is in $\mathcal{G}_{j_1}$ and $a_{n,2}$ is in $\mathcal{G}_{j_2}$.

It follows that at least one blocking ⅋-node P_i is such that $a_{i,1}$ is in $\mathcal{G}_{j'_1}$ and $a_{i,2}$ is in $\mathcal{G}_{j'_2}$, with $j'_1 \neq j'_2$. We thus obtain paths θ_1 and θ_2 in $\mathcal{R}^\varphi$ such that, for $k \in \{1,2\}$, the first edge of θ_k is $\overline{e_{j'_k}}$ and its last edge is $a^+_{i,k}$. Then $\varphi^{-1}(\theta_1)$ and $\varphi^{-1}(\theta_2)$ are disjoint strong switching paths in $\mathcal{R}$ with source T and target P, which makes $\overline{\varphi^{-1}(\theta_1)}\varphi^{-1}(\theta_2)$ a proper cycle of P_i containing T. □

A *descent path* is a directed path from an internal node to a terminal node: each internal node admits a unique descent path, except for *hyp*-nodes which have as many descent paths as conclusions. Moreover, a descent path is always a strong switching path.

Lemma 2.1.22 (Finding a splitting node)**.** *Let $\mathcal{R}$ be a connected proof net, whose terminal nodes are all ⊗-nodes. Then $\mathcal{R}$ must have a splitting ⊗-node.*

Proof. If all the terminal nodes of $\mathcal{R}$ are non-splitting ⊗-nodes, we can construct an infinite path as follows:

Step 0: since $\mathcal{R}$ is not empty, it must have a terminal node T_0, and we can fix an arbitrary premise to enter T_0, and start with the empty path at T_0;

Step 1: since T_0 is a non-splitting ⊗, Lemma 2.1.21 provides some ⅋-node P_0 such that $P_0 \prec T_0$, and then we reach P_0 by following a suffix of some proper cycle of it, starting from the other premise of T_0;

Step 2: now we can follow the descent path from P_0 to reach a new terminal node T_1;

Step 3: and then we iterate from Step 1, with T_1 in place of T_0.

Since this path is infinite, some edge must be repeated: to obtain a contradiction, it is sufficient to deduce the existence of a cycle in a switching graph of $\mathcal{R}$.

Let us formalize the construction as follows. Let T_0 be any non-splitting terminal ⊗-node and γ_0 be the empty path at T_0. Now, assuming the path γ_n has been constructed, with $T_n = \mathsf{t}(\gamma_n)$ a terminal ⊗-node, we define T_{n+1} and γ_{n+1} as follows. We fix a_n to be a premise of T_n such that a_n^+ is not the last edge of γ_n (we have only one choice, unless $n = 0$). Choose P_n to be a ⅋-node such that $P_n \prec T_n$ and let π_n be any proper cycle of P_n containing the edge a_n^-. We can write $\pi_n = \overline{\lambda_n}\rho_n$ where $\mathsf{s}(\lambda_n) = \mathsf{s}(\rho_n) = T_n$ and $\mathsf{t}(\lambda_n) = \mathsf{t}(\rho_n) = P_n$, so that a_n^- is the first edge of ρ_n. Then we define δ_n to be the descent path from P_n and $T_{n+1} = \mathsf{t}(\delta_n)$. Finally, we set $\gamma_{n+1} = \gamma_n \rho_n \delta_n$.

For some sufficiently large n_0, γ_{n_0} crosses some arrow twice. For $0 \leq i < n_0$, for any prefix γ of γ_{n_0} that is also a suffix of γ_i, we write $\gamma = \gamma_i \gamma_i'$; if moreover ρ_i is a prefix of γ_i', we write $\gamma_i' = \rho_i \varphi_i$ so that $\gamma = \gamma_i \rho_i \varphi_i$.

Let γ be the longest simple prefix of γ_{n_0} such that, for $0 \leq i < n_0$, if γ is a suffix of $\gamma_i \rho_i$ then φ_i is disjoint from π_i: in other words, to construct γ, we follow γ_{n_0} and stop the first time we reach an arrow that was crossed before, or that is crossed by the proper cycle π_i of a previous blocking $⅋$-node P_i.

By construction, γ is simple. Moreover, there exists an edge e such that γe is a prefix of γ_{n_0} and:

(a) either $\gamma = \gamma_{i_0} \rho_{i_0} \varphi_{i_0}$ and π_{i_0} crosses e, for some $i_0 < n_0$;

(b) or γ crosses e.

Since γ is simple and ρ_i and δ_i are strong switching paths for each i, Lemma 2.1.19 entails that γ is a strong switching path, as well as each γ_i' and each φ_i. Let us write $V \coloneqq \mathsf{t}(\gamma) = \mathsf{s}(e)$.

In case (a), observe that φ_{i_0} is not empty: otherwise $e = a^+$ where a is the conclusion of P_{i_0}, hence π_{i_0} crosses a, which violates Lemma 2.1.20. Hence φ_{i_0} is a switching path whose first edge is a^+, and such that $\mathsf{t}(\varphi_{i_0}) = V$ occurs in π_{i_0}: again, this violates Lemma 2.1.20.

In case (b), observe that, for some $n < n_0$, either $\gamma_n' e$ is a prefix of ρ_n, or we can write $\gamma_n' = \rho_n \delta_n'$ with $\delta_n' e$ a prefix of δ_n. In both cases, γ_n must cross e: indeed, ρ_n and δ_n are simple, and also disjoint by Lemma 2.1.20, so γ_n' cannot cross e. Hence there, for some $0 \leq i_0 < n$, e is crossed by either ρ_{i_0} or δ_{i_0}. Note that γ_n' is not empty: otherwise, γ_n crosses both premises of T_n, which yields a switching cycle with source T_n, thus violating Lemma 2.1.18.

If ρ_{i_0} crosses e, case (a) applies, and we obtain a contradiction. Hence δ_{i_0} crosses e and then V occurs in δ_{i_0}. Write γ' for the suffix of φ_{i_0} starting at that occurrence: γ' a non empty suffix of γ with $\mathsf{s}(\gamma') = V$, hence it is a switching cycle; moreover, its first edge is either an edge of δ_{i_0} or $a_{i_0+1}^-$ with a_{i_0+1} a premise of T_{i_0+1}, which violates Lemma 2.1.18.

□

Theorem 2.1.23 (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 Lemmas 2.1.15 and 2.1.16, $\mathcal{R}$ is connected sequential as soon as $\mathcal{R}[\otimes/cut]$ is. 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 $\mathcal{R}$.

If $\mathcal{R}$ contains a terminal *hyp*-node then it is reduced to that node and its conclusions: we conclude by (S1).

If $\mathcal{R}$ contains a terminal $⅋$-node, then:

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

Otherwise, all the terminal nodes of $\mathcal{R}$ are $\otimes$-nodes, and Lemma 2.1.21 allows to apply Lemma 2.1.22, and we obtain a splitting terminal $\otimes$-node T with premises a_1 and a_2:

- we consider the proof structure $\mathcal{R}'$ obtained from $\mathcal{R}$ by replacing T and its conclusion with two fresh $\bullet$-nodes, with premises a_1 and a_2;
- $\mathcal{R}'$ 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.
□

2.2 Multiplicative units and the mix rules

2.2.1 Multiplicative units and jumps

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 $(\mathbf{1})$ and $(\bot)$. A natural idea is to extend the definition of proof structures with two new node labels $\mathbf{1}$ and $\bot$, and require that each $\mathbf{1}$-node (resp. each $\bot$-node) has no premise and exactly one conclusion, of type $\mathbf{1}$ (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 $\mathbf{1}$-node;
- the translation of
$$\frac{\begin{array}{c}\pi \\ \Gamma\end{array}}{\vdash \Gamma, \bot}\ (\bot)$$
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 **1**-node forms a connected component: eliminating this cut simply amounts to removing this component.

Proposition 2.2.1 (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 2.1.8, except that we drop the connectedness requirement, which allows to treat the translation of the $(\bot)$ rule. □

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 2.1, one possible fix is the introduction of jumps, restoring the connectivity of $\bot$-nodes.

2.2.1.1 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 $j(B) \in \mathcal{N}(\mathcal{S})$. 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 2.2.2 (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. □

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.

2.2.1.2 Sequentialization of connected proof nets with jumps

As a converse to Proposition 2.2.2, 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 2.2.3. *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 $⅋$-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 2.2.4. *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, for which we chose the source of an incoming arrow of B in $\mathcal{G}(\mathcal{R}, j)$. More explicitly: if $j(B)$ is a $⅋$- 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 2.2.5 (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:

- remove all $\bot$-nodes;
- for each *ax*-node N with conclusions a_1 and a_2, fix an enumeration $B_1, \ldots, B_n$ of $j^{-1}(N)$, and write a'_i for the conclusion arrow of B_i in $\mathcal{R}$, for $1 \leq i \leq n$; then replace N with an *hyp*-node, with $n+2$ conclusions a_1, a_2 and a'_i for $1 \leq i \leq n$, leaving each $\mathsf{t}(a'_i)$ unchanged;

- similarly replace each **1**-node with conclusion a_0, with an *hyp*-node with $n+1$ conclusions a_0 and a'_i for $1 \leq i \leq n$, with $a'_1, \ldots, a'_n$ constructed as above.

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 $\perp$-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 2.1.23: $\mathcal{R}'$ 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 $\perp$-nodes $B_1, \ldots, B_n$, 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 $A^\perp$. Then we can set π to be either:

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

or

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

followed by n applications of the $(\perp)$ rule.

(S2) If $\mathcal{R}'$ is a cut between a conclusion of $\mathcal{R}'_1$ and a conclusion of $\mathcal{R}'_2$, where $\mathcal{R}'_1$ and $\mathcal{R}'_2$ are connected sequential structures, then $\mathcal{R}$ is a cut C between a conclusion of $\mathcal{R}_1$ and a conclusion of $\mathcal{R}_2$, such that each $\mathcal{R}'_i$ is obtained from $\mathcal{R}_i$ by as above. If moreover $\mathcal{R}$ is typed, then $\mathcal{R}_1$ and $\mathcal{R}_2$ are typed with conclusion sequents Γ_1, A and $A^\perp, \Gamma_2$, so that the premises of C have dual types A and $A^\perp$. The induction hypothesis yields π_1 and π_2 such that $\mathtt{ps}(\pi_i) = \mathcal{R}_i$, and we can set

$$\pi := \frac{\overset{\pi_1}{\Gamma_1, A} \qquad \overset{\pi_2}{A^\perp, \Gamma_2}}{\Gamma_1, \Gamma_2}\ (cut)$$

(S3) If $\mathcal{R}'$ is a $\otimes$-node between a conclusion of $\mathcal{R}'_1$ and a conclusion of $\mathcal{R}'_2$, where $\mathcal{R}'_1$ and $\mathcal{R}'_2$ are connected sequential structures, then $\mathcal{R}$ is a $\otimes$-node T between a conclusion of $\mathcal{R}_1$ and a conclusion of $\mathcal{R}_2$, such that each $\mathcal{R}'_i$ is obtained from $\mathcal{R}_i$ by as above. If moreover $\mathcal{R}$ is typed, then $\mathcal{R}_1$ and $\mathcal{R}_2$ are typed with conclusion sequents Γ_1, A and B, Γ_2, so that the premises of T have types A and B. The induction hypothesis yields π_1 and π_2 such that $\mathtt{ps}(\pi_i) = \mathcal{R}_i$, and we can set

$$\pi := \frac{\overset{\pi_1}{\Gamma_1, A} \qquad \overset{\pi_2}{B, \Gamma_2}}{\Gamma_1, A \otimes B, \Gamma_2}\ (\otimes)$$

(S4) If $\mathcal{R}'$ is a $⅋$-node between two conclusions of $\mathcal{R}'_1$ where $\mathcal{R}'_1$ is a connected sequential structure, then $\mathcal{R}$ is a $⅋$-node P between two conclusions of $\mathcal{R}_1$

such that $\mathcal{R}'_1$ is obtained from $\mathcal{R}_1$ by as above. If moreover $\mathcal{R}$ is typed, then $\mathcal{R}_1$ is typed with conclusion sequent Γ_1, A, B so that the premises of P have types A and B. The induction hypothesis yields π_1 such that $\mathtt{ps}(\pi_1) = \mathcal{R}_1$, and we can set

$$\pi \coloneqq \dfrac{\begin{array}{c}\pi_1\\ \Gamma_1, A, B\end{array}}{\Gamma_1, A \parr B}\ (\parr)$$

□

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 $(\mathbf{1})$. For the purposes of sequentialization, it would thus be sufficient to consider initial jump functions only. On the other hand, allowing jumps from arbitrary nodes makes the it easier to describe the preservation of connected proof nets with jumps under cut elimination.

2.2.1.3 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 2.2.6. *If $(\mathcal{S}, j)$ is a connected proof net with jumps, and $\mathcal{S} \to \mathcal{S}'$ then there exists a jump function j' on $\mathcal{S}'$ making $(\mathcal{S}', j')$ a connected proof net with jumps.*

Proof. By Lemma 2.2.4, 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 $\mathcal{S} \to \mathcal{S}'$. 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 $j'(B) \coloneqq N$.

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 2.1.9 and 2.1.10 only rely on transformations of switching graphs that are local to the eliminated cut. They can thus be adapted straightforwardly to the reduction $\mathcal{S} \to \mathcal{S}'$, 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.

2.2.2 The Mix Rules

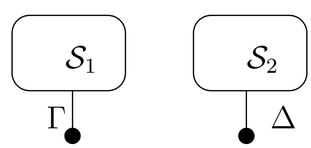
The two mix rules are the nullary mix rule (mix_0) and the binary mix rule (mix).

$$\frac{}{\vdash}\ (mix_0) \qquad\qquad \frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}\ (mix)$$

We can interpret these two rules as proof structure constructions.

The (mix_0) rule is translated into the empty proof structure.

The (mix) 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 $\mathcal{S}_2$.



2.2.2.1 Dealing with the empty proof net

Lemma 2.2.7 (Sociability of (mix_0))**.** *If π is a proof in any fragment of* LL *with (mix_0) and (mix) rules, by applying (possibly many times) the transformation:*

$$\frac{\dfrac{}{\vdash}\ (mix_0) \qquad \begin{array}{c}\pi\\ \vdash \Gamma\end{array}}{\vdash \Gamma}\ (mix) \qquad \mapsto \qquad \begin{array}{c}\pi\\ \vdash \Gamma\end{array}$$

we obtain either the proof $\frac{}{\vdash}\ (mix_0)$ *or a proof without the (mix_0) rule.*

Proof. The transformation described can only be applied a finite number of times (the number of rules strictly decreases). Assume we apply it as many times as possible. If the obtained proof contains an occurrence of the (mix_0) rule, it is the only rule of the proof since the only possible rule below it is (mix) (it must admit the empty sequent $\vdash$ as a premise) but then the transformation can be applied one more time, a contradiction. □

Note that the transformation considered in Lemma 2.2.7 does not modify the associated proof structure.

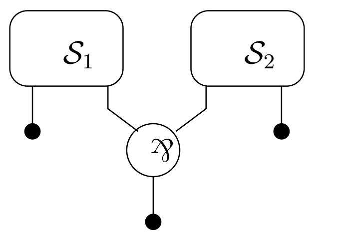
Proposition 2.2.8 (Soundness and Sequentialization with (mix_0))**.** *A typed* MLL_v*-proof structure is the translation of a sequent calculus proof of* $\mathsf{MLL}_v + (mix_0)$ *if and only if it is acyclic and its switching graphs have at most one connected component.*

Proof. By Lemma 2.2.7, for soundness it is enough to apply Proposition 2.1.8 and to see that the empty proof structure (obtained from the (mix_0) rule) has empty switching graphs thus is acyclic and with no connected component.

Concerning sequentialization, since the only multigraph with no connected component is the empty one, the only multiplicative proof structure with switching graphs with no connected component is the empty one which is the translation of the (mix_0) rule. For acyclic and connected multiplicative proof structures, we apply Theorem 2.1.23. □

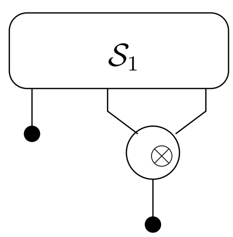
2.2.2.2 Clasping chains and the almost connected case

The treatment of proof structures whose switching graphs have more than one connected component is more delicate. The difficulty lies in the fact that this does not necessarily delineate two connected components in the proof structure itself. Consider for instance a structure $\mathcal{S}$ of the shape:



In a switching graph for $\mathcal{S}$, the nodes of $\mathcal{S}_1$ are disconnected from those of $\mathcal{S}_2$, but the terminal ⅋-node might be connected to nodes of $\mathcal{S}_1$ or of $\mathcal{S}_2$ depending on the switching.

In fact, this is exactly the problem we faced in the proof of sequentialization for connected proof structures: in a structure of the shape



we know that the premises of the ⊗-node are disconnected in every switching graph of $\mathcal{S}_1$, but this does not necessarily entail that this node is splitting.

In Theorem 2.1.23, we relied on the existence of some proper cycles to find a splitting ⊗-node in a proof net whose terminal nodes are all ⊗-nodes, thanks to Lemma 2.1.22. However, this hypothesis is no longer available in a non connected proof net.

We will nonetheless establish that sequentialization can be deduced from connected sequentialization, in case every ⅋-node admits a proper cycle: we say a proof net with this property is *almost connected*. Indeed, we will show that an almost connected proof net is always a sum of connected proof nets — that we can thus sequentialize component by component.

The proof relies on the fact that the length of clasping chains is bounded: a *clasping chain* in a proof structure is a sequence $(P_i)_{0\leq i\leq n}$ such that each P_i is a $⅋$-node and $P_i \prec P_{i+1}$ for $0 \leq i < n$.

Lemma 2.2.9 (Clasping chains are bounded)**.** *In a proof net $\mathcal{R}$, there is no clasping chain $P_0 \prec \cdots \prec P_{n+1}$ with $P_0 = P_{n+1}$. Moreover, the length of clasping chains is bounded.*

Proof. Assume, towards a contradiction, that there is a chain $P_0 \prec \cdots \prec P_{n+1}$ with $P_0 = P_{n+1}$ in a proof net $\mathcal{R}$. Moreover assume that this chain is of minimal length: in particular the P_i's are pairwise distinct.

For $0 \leq i \leq n$, write π_i for a proper cycle of P_i such that P_{i+1} occurs in π_i. Since π_i is proper, it must cross the conclusion $a_{i+1,0}$ of P_{i+1}: w.l.o.g., we can write $\pi_i = \lambda_i\rho_i$, where the first edge of ρ_i is $a^+_{i+1,0}$ — otherwise, we consider $\overline{\pi_i}$ instead. We moreover write $\rho_{n+1} \coloneqq \rho_0$ and $\rho \coloneqq \rho_{n+1}\rho_n \cdots \rho_0$.

By Lemma 2.1.20, ρ_{n+1} and ρ_n are disjoint; and by definition, ρ_{n+1} and ρ_0 are not disjoint. Consider the longest simple prefix γ of ρ: we can write $\gamma = \rho_{n+1} \cdots \rho_{i+1}\rho'_i$ for some i with $0 < i \leq n$, where ρ'_i is a non empty prefix of ρ_i. Moreover, there is an edge e such that γe is a prefix of ρ and e is crossed by γ: if $\rho'_i = \rho_i$, $e = a^+_{i,0}$ and Lemma 2.1.20 entails that ρ'_i does not cross e; and if ρ'_i is a proper prefix of ρ_i then e is an edge of ρ_i which is a simple path, hence ρ'_i does not cross e.

It follow that e is crossed by some ρ_j with $i < j \leq n+1$; in particular, $V \coloneqq \mathsf{t}(\rho'_i)$ occurs in ρ_j. Since each ρ_k is a strong switching path, Lemma 2.1.19 entails that γ is a switching path, hence its suffix $\gamma' \coloneqq \rho_{j-1} \cdots \rho_{i+1}\rho'_i$ is a switching path with first edge $a^+_{j,0}$ and such that $\mathsf{t}(\gamma') = V$ occurs in π_j: this yields a contradiction with Lemma 2.1.20.

We have thus obtained the first result. Now, assuming that there are clasping chains of unbounded length, it is sufficient to consider a clasping chain whose length exceeds the number of $⅋$-nodes of $\mathcal{R}$ to obtain a cyclic clasping chain, hence a contraction by the first result. □

Given a $⅋$-node P in a proof net $\mathcal{R}$, we write $cr(P)$ for the greatest number n of nodes of a clasping chain $P = P_1 \prec \cdots \prec P_n$, and call $cr(P)$ the *clasping rank* of P. Note that $cr(P) \geq 1$. Then, if γ is a path in $\mathcal{R}$, the clasping rank $cr(\gamma)$ of γ is the maximum of the clasping ranks of the $⅋$-nodes having at least one premise edge crossed by γ — we set $cr(\gamma) = 0$ if there is no such $⅋$-node.

We are now ready to show that, in an almost connected proof net $\mathcal{R}$, each connected component of the graph of $\mathcal{R}$ defines a connected proof net: in other words, all the switching graphs of a connected component are connected.

Lemma 2.2.10 (Switching paths in an almost connected proof net)**.** *Let $\mathcal{R}$ be an almost connected proof net. Then for every switching φ of $\mathcal{R}$, and every path γ in $\mathcal{R}$, there exists a path γ' in $\mathcal{R}^\varphi$ having the same endpoints as γ.*

Proof. Note that we do not require γ' to be a simple path nor to be non-empty: anyway, we will only use this result to establish a connectedness property. Also note that γ' and $\gamma'^{-1}(\varphi)$ need not have the same endpoints: to obtain a connectedness property in $\mathcal{R}^\varphi$ it is thus important to focus on γ' rather than $\varphi^{-1}(\gamma')$. We reason by well-founded induction on $cr(\gamma)$.

Assume $cr(\gamma) = n$ and the induction hypothesis applies to all paths of clasping rank strictly less than n. Write $a_1, \ldots, a_k$ for the premises of $\parr$-nodes crossed by γ such that $\varphi(\mathsf{t}(a_i)) \neq a_i$. Observe that the edges of γ have the same endpoints in $\mathcal{R}^\varphi$ as in $\mathcal{R}$, except for the edges supported by the a_i's. To obtain a path γ' in $\mathcal{R}^\varphi$ with the same endpoints as γ in $\mathcal{R}$ it is thus sufficient to find a path γ_i in $\mathcal{R}^\varphi$ with the same endpoints as a_i in $\mathcal{R}$, for $1 \leq i \leq k$.

Write $P_i \coloneqq \mathsf{t}(a_i)$ and $a'_i \coloneqq \varphi(P_i)$ for the other premise of P_i. Let π_i be a proper cycle of P_i:[1] w.l.o.g. (otherwise consider $\overline{\pi_i}$), we can write $\pi_i = a_i^- \rho_i {a'_i}^+$. Every $\parr$-node P' occurring in ρ_i is clasped by P_i, hence $cr(P') < cr(P_i) \leq cr(\gamma)$: it follows that $cr(\rho_i) < cr(\gamma)$, and the induction hypothesis yields a path ρ'_i in $\mathcal{R}^\varphi$ with $\mathsf{s}(\rho'_i) = \mathsf{s}(a_i)$ and $\mathsf{t}(\rho'_i) = \mathsf{s}(a'_i)$. It is then sufficient to set $\gamma_i \coloneqq \rho'_i {a'_i}^+$. □

Corollary 2.2.11 (Sequentialization of almost connected proof nets)**.** *Every almost connected proof net is a sequential proof structure.*

Proof. First observe that, for any switching φ of a proof structure $\mathcal{R}$, each node n of $\mathcal{R}^\varphi$ is connected to a node of $\mathcal{R}$ in $\mathcal{R}^\varphi$:

- either n is already a node of $\mathcal{R}$;
- or n is the newly introduced target $P^\bullet$ of a premise a of a $\parr$-node P, and then a^- connects n to $\mathsf{s}(a)$ — and $\mathsf{s}(a)$ is already in $\mathcal{R}$, because the source of arrows is unchanged in $\mathcal{R}^\varphi$.

Assume that $\mathcal{R}$ is an almost connected proof net, whose graph $\mathcal{G}_\mathcal{R}$ is connected, and fix a switching φ of $\mathcal{R}$. We know $\mathcal{R}^\varphi$ is acyclic. Given two nodes in $\mathcal{R}^\varphi$, we have just observed that these nodes are connected to nodes of $\mathcal{R}$; since $\mathcal{G}_\mathcal{R}$ is connected, the two latter nodes are connected by a path γ in $\mathcal{R}$; and the previous Lemma yields a path in $\mathcal{R}^\varphi$ with the same endpoints. It follows that $\mathcal{R}$ is a connected proof net, and Theorem 2.1.23 ensures that $\mathcal{R}$ is connected sequential, hence sequential.

Now assume that $\mathcal{R}$ is an almost connected proof net: applying the previous reasoning to the connected components of $\mathcal{R}$, we obtain a sequential proof structure for each component. Then we conclude by induction on the number of connected components: if there is none, we apply Item (S5); in the inductive case, we apply Item (S6). □

[1] Here it is important to note that π_i is given by a switching φ_i, that need not be compatible with φ' on the edges of π_i.

2.2.2.3 Nonconnected sequentialization

It only remains to tackle the case of proof nets having $\parr$-nodes without proper cycles. The key observation is that such a node can be replaced with a $\otimes$-node without introducing switching cycles. Moreover, this operation decreases the number of connected components of switching graphs.

If $\mathcal{S}$ is a proof structure and P a $\parr$-node of $\mathcal{S}$, we write $\mathcal{S}[\otimes/P]$ for the structure obtained by changing P into a $\otimes$-node.

Lemma 2.2.12. *If $\mathcal{S}[\otimes/P]$ is a proof net, then so is $\mathcal{S}$. Moreover in this case, $\#_{cc}(\mathcal{S}) = \#_{cc}(\mathcal{S}[\otimes/P]) + 1$.*

Proof. Write $\mathcal{S}' \coloneqq \mathcal{S}[\otimes/P]$. Given a switching φ of $\mathcal{S}$, we obtain a switching φ' of $\mathcal{S}'$ just by forgetting the value of φ on P. Given a cycle γ in $\mathcal{S}^{\varphi}$, γ does not cross the premise of P that is rejected by φ; hence γ is also a path $\mathcal{S}'^{\varphi'}$, which is a cycle. It follows that if $\mathcal{S}'$ is a proof net then so is $\mathcal{S}$.

The switching graphs of $\mathcal{S}'$ have the same number of arrows as those of $\mathcal{S}$, but one node less: in case both are proof nets, **??** entails the required identity. $\square$

Lemma 2.2.13. *If P is a $\parr$-node of a proof structure $\mathcal{S}$, then $\mathcal{S}$ is sequential as soon as $\mathcal{S}[\otimes/P]$.*

Proof. The proof is by a straightforward induction on the sequentiality of $\mathcal{R}[\otimes/P]$: one simply replaces the application of (S3) to P, with the application of first (S6) then (S4). $\square$

Lemma 2.2.14 ($\parr$-node without proper cycle)**.** *Let $\mathcal{R}$ be a proof net and assume P is a $\parr$-node of $\mathcal{R}$, which does not have any proper cycle. Then $\mathcal{R}[\otimes/P]$ is a proof net,*

Proof. Assume there is a cycle π in some switching graph $\mathcal{R}[\otimes/P]^{\varphi}$ of $\mathcal{R}[\otimes/P]$: then π must cross P, as otherwise it is also a cycle in $\mathcal{R}^{\varphi'}$, where φ' is any switching of $\mathcal{R}$ that agrees with φ on the $\parr$-nodes of $\mathcal{R}[\otimes/P]$. We can thus assume w.l.o.g. that $\mathsf{t}(\pi) = \mathsf{s}(\pi) = P$, which yields a proper cycle of P, and a contradiction with the hypothesis. $\square$

The previous Lemmas ensure that if we want to sequentialize a proof net $\mathcal{R}$ containing a $\parr$-node P without proper cycle, then it is sufficient to sequentialize $\mathcal{R}[\otimes/P]$. We are now ready to treat the general case.

Theorem 2.2.15 (Sequentialization of proof nets)**.** *Every proof net is a sequential proof structure.*

Proof. By Lemma 2.1.16, it is sufficient to treat the case of *cut*-free proof nets. We reason by induction on $\#_{cc}(\mathcal{R})$.

If $\mathcal{R}$ is almost sequential, we conclude directly by Corollary 2.2.11. Otherwise $\mathcal{R}$ contains a $\parr$-node P without proper cycle. By Lemmas 2.2.14 and 2.2.12, we can apply the induction hypothesis to $\mathcal{R}[\otimes/P]$, hence $\mathcal{R}[\otimes/P]$ is sequential, and so is $\mathcal{R}$, by Lemma 2.2.13. $\square$

Observe that the previous theorem actually applies to proof structures with *hyp*-nodes, hence to proof structures with **1**- and $\bot$-nodes.

Proposition 2.2.16 (Soundness and Sequentialization with (mix_0) and (mix))**.** *A typed multiplicative proof structure $\mathcal{S}$ is the translation of a sequent calculus proof of $\mathsf{MLL}_0 + (mix_0) + (mix)$ (resp. of $\mathsf{MLL}_0 + (mix)$) if and only if it is a proof net (resp. a non empty proof net).*

Proof. By a straightforward induction on sequent calculus proofs, we obtain that the translation $\mathtt{ps}(\pi)$ of a proof π in $\mathsf{MLL}_0 + (mix_0) + (mix)$ (resp. in $\mathsf{MLL}_0 + (mix)$) is sequential (resp. is sequential and non empty): it is thus a proof net by Lemma 2.1.14.

Conversely, given a typed multiplicative proof net $\mathcal{R}$, Theorem 2.2.15 ensures $\mathcal{R}$ is sequential: we obtain a proof π of $\mathsf{MLL}_0 + (mix_0) + (mix)$ such that $\mathcal{R} = \mathtt{ps}(\pi)$ by induction on sequential structures. Moreover, by Lemma 2.2.7, if $\mathcal{R}$ is not empty, we can assume π does not use (mix_0). ☐

2.3 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.

2.3.1 Multiplicative Exponential Linear Logic with Mix

The formulas of multiplicative exponential linear logic (MELL) are defined as:

$$A, B ::= X \mid X^\bot \mid A \otimes B \mid A \parr B \mid !A \mid ?A$$

The connective $(.)^\bot$ is extended into an involution on all formulas by:

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

For MELL, we consider the rules of MLL as well as the two 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 more traditional one $\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ (w)$. The two rules are inter-derivable:

$$\frac{\vdash \Gamma \qquad \frac{}{\vdash ?A}\ (w_0)}{\vdash \Gamma, ?A}\ (mix) \qquad\qquad \frac{\frac{}{}\ (mix_0)}{\vdash ?A}\ (w)$$

2.3.2 Proof Structures

boxes $\mathcal{B}$, *main door*, with explicit $?p$ nodes (*auxiliary doors*)
content of a box
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)

2.3.3 Correctness Criterion

acyclicity
sequentialization

2.3.4 Cut Elimination

2.3.4.1 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 2.3.1 (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 □

2.3.4.2 Properties

The goal of this section is to prove the convergence of the reduction of proof nets.

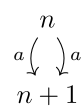
Lemma 2.3.2 (Numbered Congruence)**.** *If $\mathcal{R}$ is a proof net containing $\mathcal{R}_0$ as a sub proof net a depth* 0*, if $\mathcal{R}_0$ equipped with label m reduces to $\mathcal{R}_0'$ with label m' then $\mathcal{R}$ reduces to $\mathcal{R}'$ where $\mathcal{R}'$ is obtained from $\mathcal{R}$ 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** ☐

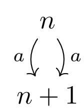
Proposition 2.3.3 (Local Confluence)**.** *The reduction of numbered proof nets is locally confluent.*

Proof.

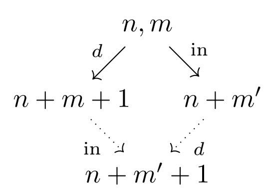
- a/a (shared *cut*)



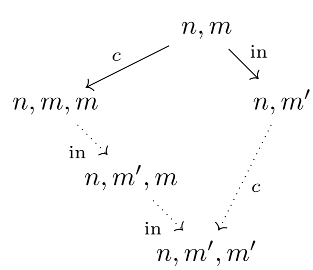
- a/a (shared *ax*)



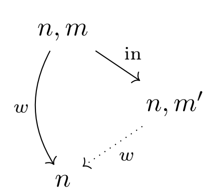
- d/in



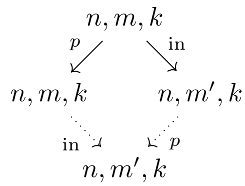
- c/in



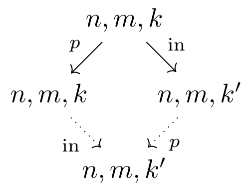
- w/in



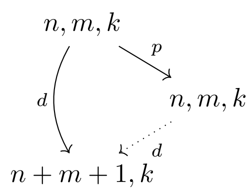
- p/in (left side)



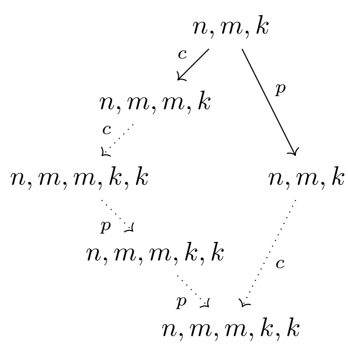
- p/in (right side)



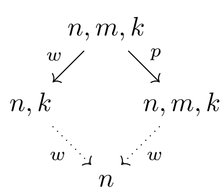
- d/p



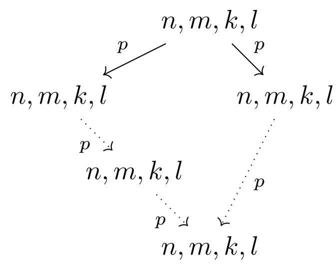
- c/p



- w/p



- p/p



- TODO p/p both on auxiliary doors of the same box

TODO □

Proposition 2.3.4 (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 **??**, 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 undirected 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 $\to_{am}$ reduction is the reduction of proof nets obtained by considering only $\to_a$ and $\to_m$ steps.
- The $\to_{\not w}$ reduction, also called *strict reduction*, is the reduction of proof nets restricted to non w steps.
- The $\to_{\not c}$ reduction is the reduction of proof nets restricted to non c steps.

Lemma 2.3.5 (Strong am Normalization)**.** *The $\rightarrow_{am}$ reduction of proof nets is strongly normalizing.*

Proof. We use Proposition **??**, since the number of nodes of proof nets is strictly decreasing along an a or m reduction step. □

Lemma 2.3.6 (Strong w Normalization)**.** *The $\rightarrow_{w}$ reduction of proof nets is strongly normalizing.*

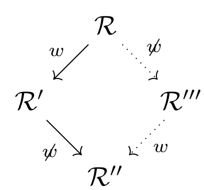
Proof. We use Proposition **??**, since the number of nodes of proof nets is strictly decreasing along a w reduction step. □

Lemma 2.3.7 (Sub-Commutation of am and non c)**.** *The reduction relations $\rightarrow_{am}$ and $\rightarrow_{\not{c}}$ sub-commute.*

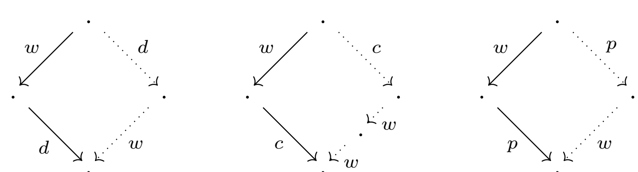
Proof. This easily comes by looking at the proof of Proposition 2.3.3. □

Lemma 2.3.8 (Quasi-Commutation of w over non w)**.** *The $\rightarrow_{w}$ reduction of proof nets quasi-commutes over the $\rightarrow_{\not{w}}$ reduction.*

Proof. Assume we have $\mathcal{R} \rightarrow_{w} \mathcal{R}' \rightarrow_{\not{w}} \mathcal{R}''$. If the $\rightarrow_{w}$ and the $\rightarrow_{\not{w}}$ steps do not overlap, we directly have commutation and by first applying the $\rightarrow_{\not{w}}$ step, one obtains $\mathcal{R} \rightarrow_{\not{w}} \mathcal{R}''' \rightarrow_{w} \mathcal{R}''$.



The only possible overlapping is when the $\rightarrow_{\not{w}}$ step acts on a box containing the $\rightarrow_{w}$ step, but then by looking at the _/in cases of the proof of Proposition 2.3.3, we can see we can close the diagrams in an appropriate way:



□

Lemma 2.3.9 (Weak non w Normalization)**.** *The $\rightarrow_{\not{w}}$ reduction of numbered proof nets is weakly normalizing.*

Proof. We can use the same proof as for Proposition 2.3.4, 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 2.3.4) 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 2.3.4 is that the box above c might contain some w cuts. We then see that the non w cut size strictly decreases. □

Lemma 2.3.10 (Increasing non w Reduction)**.** *The reduction $\rightarrow_{\not w}$ 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 $\mathcal{R} \rightarrow_{\not w} \mathcal{R}'$, 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 2.3.11 (Convergence)**.** *The reduction of proof nets is convergent.*

Proof. We first prove the strong normalization of the $\rightarrow_{\not w}$ reduction by means of Proposition **??**: we have Lemmas 2.3.10 and 2.3.9, and we can check in the proof of Proposition 2.3.3 that diagrams with $a \rightarrow_{\not w} b$ and $a \rightarrow_{\not w} c$ can be closed into $b \rightarrow_{\not w}^{*} d$ and $c \rightarrow_{\not w}^{*} d$ (that is there is no need for w steps in closing the diagram).

We now apply Proposition **??** to $\rightarrow_w$ and $\rightarrow_{\not w}$, using Lemmas 2.3.8 and 2.3.6 to obtain strong normalization.

We conclude with confluence by Newman's Lemma (Proposition **??**) using Proposition 2.3.3. ☐

2.3.5 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:

$$\frac{?A \qquad \dfrac{?A \quad ?A}{?A}}{?A} \qquad \text{vs} \qquad \frac{\dfrac{?A \quad ?A}{?A} \qquad ?A}{?A}$$

$$?A \qquad \text{vs} \qquad \frac{?A \quad \overline{?A}}{?A} \qquad \text{vs} \qquad \frac{\overline{?A} \quad ?A}{?A}$$

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:

$$\frac{}{\vdash !A, ?A^{\perp}}\,(ax) \qquad \mapsto \qquad \dfrac{\dfrac{\dfrac{}{\vdash A, A^{\perp}}\,(ax)}{\vdash A, ?A^{\perp}}\,(?)}{\vdash !A, ?A^{\perp}}\,(!)$$

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 2.3.12 (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 2.3.13 (Simulation). *The translation $(.)^{cw}$ is an injective strict simulation which preserves normal forms from proof nets with ? nodes into proof nets.*

Proof. TODO □

Lemma 2.3.14 (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 2.3.12, $\mathcal{R}^{cw}$ is acyclic. By Proposition 2.3.13, $\mathcal{R}^{cw} \to^+ \mathcal{R}'^{cw}$, thus by Lemma 2.3.1 $\mathcal{R}'^{cw}$ is acyclic. By Lemma 2.3.12 again, $\mathcal{R}'$ is acyclic. □

Proposition 2.3.15 (Convergence). *The reduction of proof nets with ? nodes is convergent.*

Proof. We have strong normalization by Propositions **??** and 2.3.13 and Theorem 2.3.11.

Concerning confluence, by Proposition **??** and Theorem 2.3.11, proof nets have the unique normal form property. By Propositions **??** and 2.3.13, proof nets with ? nodes have the unique normal form property. By Propositions **??**, **??**, proof nets with ? nodes are confluent thanks to strong normalization. □

2.4 Translation of the Lambda-Calculus

2.4.1 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:

$$(\tau \to \sigma)^\bullet = ?\tau^{\bullet\perp} \parr \sigma^\bullet$$

$$L, M ::= X \mid ?L^\perp \parr M$$

2.4.2 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 2.4.1
mention cut-free correctness

2.4.3 The Translation

into directed proof nets using only sub-formulas of L (or dual) and only D conclusions

2.4.3.1 Definition

Pre-translation $(.)^\circ$

$$\cfrac{\cfrac{\cfrac{}{\vdash L^\perp, L}\ (ax)}{\vdash ?L^\perp, L}\ (?)}{\vdash ?\Gamma^\perp, ?L^\perp, L}\ (w)$$

$$\frac{\vdash ?\Gamma^\perp, ?L^\perp, M}{\vdash ?\Gamma^\perp, ?L^\perp \parr M}\ (\parr)$$

$$\dfrac{\vdash ?\Gamma^{\perp}, ?L^{\perp} \parr M \qquad \dfrac{\dfrac{\vdash ?\Gamma^{\perp}, L}{\vdash ?\Gamma^{\perp}, !L}\,(!) \qquad \dfrac{}{\vdash M^{\perp}, M}\,(ax)}{\vdash ?\Gamma^{\perp}, !L \otimes M^{\perp}, M}\,(\otimes)}{\dfrac{\vdash ?\Gamma^{\perp}, ?\Gamma^{\perp}, M}{\vdash ?\Gamma^{\perp}, M}\,(c)}\,(cut)$$

By looking at the proof of Lemma 2.3.7, one can see $\rightarrow_{am}$ is sub-confluent thus it satisfies the unique normal form property (Proposition **??**). Moreover $\rightarrow_{am}$ is strongly normalizing (Lemma 2.3.5), thus we can define the *multiplicative normal form* $\mathrm{NF}_{am}(\mathcal{R})$ of a proof net $\mathcal{R}$ as its unique $\rightarrow_{am}$ normal form.

We define the translation $t^{\bullet}$ of λ-term t by $t^{\bullet} = \mathrm{NF}_{am}(t^{\circ})$.

2.4.3.2 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

2.4.3.3 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 2.4.1 (Sequentialization)**.** *Any λ-proof net is the image of a λ-term through the translation $(.)^{\bullet}$.*

Proof. TODO ☐

2.4.3.4 Kernel

The *σ-reduction* is the congruence on λ-terms generated by:

$$
\begin{array}{lr}
((\lambda y.t)\,u)\,v \to_\sigma (\lambda y.(t\,v))\,u & y \notin v \\
(\lambda y.\lambda x.t)\,u \to_\sigma \lambda x.((\lambda y.t)\,u) & x \notin u
\end{array}
$$

The *σ-equivalence* is the equivalence relation generated by the σ-reduction.

Lemma 2.4.2 (Strong Normalization)**.** *The σ-reduction is strongly normalizing.*

The σ-reduction is not locally confluent, as one can see with the following example:

$$
\begin{array}{ccc}
 & (\lambda y.\lambda z.x)\,u\,v & \\
\swarrow_\sigma & & \searrow^\sigma \\
(\lambda y.((\lambda z.x)\,v))\,u & & (\lambda y.((\lambda z.x)\,v))\,u \\
\sigma \not\downarrow & & \not\downarrow \sigma
\end{array}
$$

with $y \notin v$ and $z \notin u$.

A λ-term is called a *canonical form* if it is of the shape:

$$\overrightarrow{\lambda z}.\overrightarrow{\beta(y,u)}.(x\,\vec{v})$$

where $\beta(y,u).t = (\lambda y.t)\,u$ and all the $\vec{u}$s and $\vec{v}$s are themselves canonical forms.

Note that β-normal forms are exactly canonical forms without β-redex.

Lemma 2.4.3 (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 $t = \overrightarrow{\lambda z}.(\_\ \vec{v})$ where $\_ = x$ or $\_ = \beta(y,u).t'$. In the first case, t is a canonical form (the $\vec{v}$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 $\vec{v}$ 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 2.4.4 (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 2.4.2 and 2.4.3, 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

2.4.4 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 $o = ?o^\perp ⅋ o$. This entails that the set of the sub-formulas of formulas generated from the atom o by the unique construction $?o^\perp ⅋ o$ 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]`

2.5 Further Reading

We suggest an incomplete list of related papers.

2.5.1 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 [39].

2.5.2 Sequentialization

- A sequentialization proof based on the acyclic-connected criterion and using empires [22].
- [9]
- [6]
- The sequentialization proof we used here [33].

2.5.3 Rewriting Properties

- [42]
- [9]
- [37]

2.5.4 Extensions of the Syntax

- Modules
- Units [8, 29]
- Quantifiers [22]
- Additive connectives [21, 30]

2.5.5 Relations with the Lambda-Calculus

- [39]
- [14]

2.5.6 Complexity

- [27]

Part II

Static models

Chapter 3

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 5 for the the basic definitions and properties.

3.1 Coherent spaces

3.1.1 The coherence relation

A *coherent space* E is a structure

$$E = (|E|, \mathrel{\overset{\frown}{\smile}}_E)$$

where $|E|$ is a set (which can be assumed to be at most countable) and $\mathrel{\overset{\frown}{\smile}}_E$ is a binary reflexive and symmetric relation on $|E|$ called *coherence*.

We use the following definitions and notations:

Strict coherence: $\frown_E = (\mathrel{\overset{\frown}{\smile}}_E \cap \neq)$, that is $a \frown_E a'$ iff ($a \mathrel{\overset{\frown}{\smile}}_E a'$ and $a \neq a'$);

Incoherence: $\asymp_E = \neg \frown_E$, that is $a \asymp_E a'$ iff ($a \not\mathrel{\overset{\frown}{\smile}}_E a'$ or $a = a'$);

Strict incoherence: $\smile_E = \neg \mathrel{\overset{\frown}{\smile}}_E$, that is $a \smile_E a'$ iff $a \not\mathrel{\overset{\frown}{\smile}}_E a'$.

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 $\mathrel{\overset{\frown}{\smile}}_E$ and $\asymp_E$ is equality:

$$\forall a, a' \in |E|,\ a = a' \text{ iff } a \mathrel{\overset{\frown}{\smile}}_E a' \text{ and } a \asymp_E a'$$

A *clique* of E is a set of pairwise coherent points of $|E|$; we denote by $\mathrm{Cl}(E)$ the set of cliques:

$$\mathrm{Cl}(E) = \{u \subset |E|, \forall a, a' \in u, a \coh_E a'\}$$

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; in some sense it states that the set $\mathrm{Cl}(E)$ ordered by inclusion is a Scott domain.

Proposition 3.1.1 (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:*

$$u = \bigcup\{u_0 \in \mathrm{Cl}(E),\ u_0 \subset_{\mathrm{fin}} u\}$$

- *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:

$$E^{\perp\perp} = E$$

The last property exhibiting a canonical duality in coherent spaces is proper, we will see that it is the very reason why the category of coherent spaces is $*$-autonomous.

3.1.2 Clique spaces

Clique spaces are 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 $a \coh_E a'$ 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 3.1.2. *Let E be a coherent space; a subset $u \subset |E|$ is a clique iff it is orthogonal to all anticliques:*

$$u \sqsubset E \textit{ iff } \forall u' \sqsubset E^\perp, u \perp u'$$

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 3.1.3. *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 3.1.4. *If E is a coherent space then* 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* 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,* Coh(Clique$(E)) = E$ *and for any clique space E,* Clique(Coh$(E)) = E$.

Proof. The fact that Clique(E) is a clique space is immediate consequence of the proposition 3.1.2. 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 3.1.3.

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 Coh(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 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 $C_E = \mathrm{Cl}(\mathrm{Coh}(E))$.

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: $u \in \mathrm{Cl}(\mathrm{Coh}(E))$. Conversely let $u \in \mathrm{Cl}(\mathrm{Coh}(E))$, $u' \in C_E^{\perp}$ 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 $u \in {C_E}^{\perp\perp} = C_E$. □

Example 3.1.5 (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.

3.2 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:

$$X \to Y \simeq\ !X \multimap Y$$

3.2.1 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 3.2.1 (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 3.1.5) 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 3.2.2 (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 **V**, **F** respectively. It is a kind of ternary parallel-or but contrarily to the parallel-or it is stable.

3.2.2 Cartesian product

If X_0 and X_1 are coherent spaces we define $X_0 \,\&\, X_1 = (|X_0 \,\&\, X_1|, \asymp_{X_0 \& X_1})$ by:

Web: $|X_0 \,\&\, X_1| = \{0\} \times |X_0| \;\cup\; \{1\} \times |X_1|$.

Coherence: $(i, a) \asymp_{X_0 \& X_1} (j, b)$ iff $\begin{cases} i \neq j & \text{or} \\ i = j \text{ and } a \asymp_{X_i} b \end{cases}$

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:

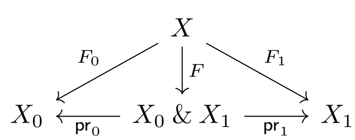
Lemma 3.2.3. *A map $F : \mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1) \to \mathrm{Cl}(X)$ is bi-stable iff the map $F \circ \varphi : X_0 \,\&\, X_1 \to X$ is stable where φ is the bijective function defined by:*

$$\begin{aligned} \varphi : \mathrm{Cl}(X_0 \,\&\, X_1) &\to \mathrm{Cl}(X_0) \times \mathrm{Cl}(X_1) \\ x &\mapsto (\mathsf{pr}_0(x), \mathsf{pr}_1(x)) \end{aligned}$$

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 3.2.4. *The space $X_0 \,\&\, X_1$ is a cartesian product (see section 5.2) 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.

3.2.3 The coherent space of stable functions

The fundamental lemma on stable functions is:

Lemma 3.2.5. *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 $(x_0, b) \coh_{X \Rightarrow Y} (x'_0, b')$ 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 3.2.6. *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 3.2.7. *The category of coherent spaces and stable functions is cartesian closed.*

Proof. In view of the definition of cartesian closed categories (see section 5.2.1.3) we just have to define the evaluation map and the curryfication operation and check equations 5.1:

$$\begin{array}{lrcl} \mathsf{Ev}: & (X \Rightarrow Y) \,\&\, X & \to & Y \\ & (f, x) & \mapsto & \mathsf{Fun}(f)(x) \end{array}$$

$$\begin{array}{rcl} \mathsf{Cur} : \mathsf{Stable}(Z \,\&\, X, Y) & \to & \mathsf{Stable}(Z, X \Rightarrow Y) \\ F : Z \,\&\, X \to Y & \mapsto & \begin{array}{rcl} \mathsf{Cur}(F) : Z & \to & X \Rightarrow Y \\ z & \mapsto & \mathsf{Tr}(\lambda x F(z, x)) \end{array} \end{array}$$

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. □

3.3 The monoidal structure of coherent spaces

3.3.1 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 3.3.1. 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 3.3.2 (Identity). The identity function $\mathsf{Id}_X : X \to X$ is linear.

Example 3.3.3 (Projections). The projections $\mathsf{pr}_i : X_0 \,\&\, X_1 \to X_i$ are linear.

Example 3.3.4. 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 3.3.5. 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 $G(\mathbf{V}, \bot, \bot) = G(\bot, \mathbf{F}, \bot) = \bot$ but $(\mathbf{V}, \bot, \bot) \cup (\bot, \mathbf{F}, \bot) = (\mathbf{V}, \mathbf{F}, \bot)$ and $G(\mathbf{V}, \mathbf{F}, \bot) = \mathbf{V}$ whereas $\bot \cup \bot = \bot$.

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 $F \in \mathsf{Lin}(X, Y)$.

3.3.2 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 3.3.6. *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 3.3.7. *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:

$$\begin{array}{ccc} X_0 \,\&\, X_1 & \xrightarrow{F} & X \\ {\scriptstyle\psi}\big\downarrow & \nearrow_{\hat{F}} & \\ X_0 \otimes X_1 & & \end{array}$$

Remark 3.3.8 (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.

3.3.3 The coherent space of linear functions

As with stable functions there is a fundamental lemma for linear functions:

Lemma 3.3.9. *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 3.2.5 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 3.3.10. *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 3.3.11. *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 5.5):

$$\begin{array}{rrcl} \mathsf{Ev}_\ell : & (X \multimap Y) \otimes X & \to_\ell & Y \\ & f \times x & \mapsto & \mathsf{Fun}_\ell(f)(x) \end{array}$$

$$\begin{array}{rrcl} \mathsf{Cur}_\ell : & \mathsf{Lin}(Z \otimes X, Y) & \to & \mathsf{Lin}(Z, X \multimap Y) \\ & F : Z \otimes X \to_\ell Y & \mapsto & \begin{array}{rcl} \mathsf{Cur}_\ell(F) : Z & \to & X \multimap Y \\ z & \mapsto & \mathsf{Tr}_\ell(\lambda x F(z \times x)) \end{array} \end{array}$$

where Ev_ℓ is defined by extension by linearity (see remark 3.3.8 above). □

3.3.4 Duality

Recall that $X^\perp$ is the dual of X defined by $|X^\perp| = |X|$ and $\overset{\frown}{\smile}_{X^\perp} = \asymp_X$. 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:

$$\begin{array}{l} X \multimap Y \to_\ell Y^\perp \multimap X^\perp \\ \qquad f \mapsto f^\perp = \{(b,a) \in |Y| \times |X|,\ (a,b) \in f\} \end{array}$$

Theorem 3.3.12. *The category of coherent spaces and linear function is ∗-autonomous (see 5.6).*

In particular the space **1** is the dualizing object that we will also denote $\perp$ to emphasize this fact. We also have the linear isomophism:

$$X^\perp \simeq X \multimap \perp$$

Note that the fact that the dualizing object is the same as the unit of the tensor is a peculiarity of coherent spaces.

3.3.5 Additive constructions

Applying duality to the cartesian product & gives rise to a new construction: the $\oplus$ thus defined by:

$$X_0 \oplus X_1 = (X_0^\perp \mathbin{\&} X_1^\perp)^\perp$$

which can be explicited:

Web: $|X_0 \oplus X_1| = \{0\} \times X_0 \ \cup\ \{1\} \times X_1$.

Coherence: $(i,a) \coh_{X_0 \oplus X_1} (j,b)$ iff $\begin{cases} i = j & \text{and} \\ a \coh_{X_i} b \end{cases}$

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 3.3.13. *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 3.3.14. 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 leaded him to the discovery of linear functions.

Remark 3.3.15. 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 $\oplus$.

3.3.6 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 3.3.16. 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 is only 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 3.3.17. The distributivity isomorphisms were another reason Girard invoked for the terminology additive/multiplicative.

3.4 Exponentials

Recall the definition of the space $X \Rightarrow Y$ in section 3.2.3: $(x_0, b) \frown_{X \Rightarrow Y} (x_0', b')$ iff $x_0 \cup x_0' \not\sqsubset X$ or $b \frown_Y b'$ where $(x_0, b), (x_0', b') \in |X \Rightarrow Y| = \mathrm{Cl_{fin}}(X) \times |Y|$. 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: $x_0 \coh_{!X} x'_0$ 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 3.4.1. *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 3.4.2. 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 $!X \multimap !Y$).

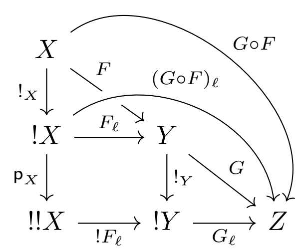
Theorem 3.4.3. *The $!$ operation is functorial on the category of coherent spaces and linear functions. Moreover it is a comonad the counit and comultiplication of which are $\mathsf{d} : ! \to_\ell \mathsf{Id}$ (dereliction) and $\mathsf{p} : ! \to_\ell !!$ (digging) defined by:*

$$\mathsf{Tr}_\ell(\mathsf{d}_X : !X \to_\ell X) = \{(\{a\}, a),\ a \in |X|\}$$
$$\mathsf{Tr}_\ell(\mathsf{p}_X : !X \to_\ell !!X) = \{(u, U),\ u = \bigcup U\}$$

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 5.6.3 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 3.4.4. 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.

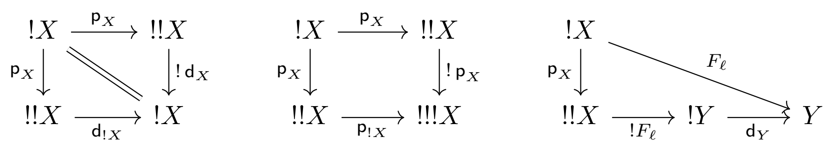
De Morgan: $(!X)^\perp = ?X^\perp$, $(?X)^\perp = !X^\perp$.

Weakening, contraction: There are two natural transformation giving $!X$ the structure of a commutative $\otimes$-comonoid:

$$\begin{array}{ll} \mathsf{w}_X : !X \to_\ell 1 & \mathsf{Tr}_\ell(\mathsf{w}_X) = \{(\emptyset, *)\} \\ \mathsf{c}_X : !X \to_\ell !X \otimes !X & \mathsf{Tr}_\ell(\mathsf{c}_X) = \{(x, (x_0, x_1)),\, x_0 \cup x_1 = x \sqsubset_{\mathrm{fin}} X\} \end{array}$$

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:

$$!(X \,\&\, Y) \simeq !X \otimes !Y$$
$$!\top \simeq \mathbf{1}$$

$$?(X \oplus Y) \simeq ?X \parr ?Y$$
$$?0 \simeq \bot$$

3.5 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 make 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 4.3), that allowed among other things the discovery of differential lambda-calculus.

Chapter 4

Rel

4.1 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.

$$[\mathbf{1}] = [\bot] = \{*\} \qquad [A \otimes B] = [A \parr B] = [A] \times [B]$$
$$[0] = [\top] = \emptyset \qquad [A \oplus B] = [A \mathbin{\&} B] = \{1\} \times [A] \cup \{2\} \times [B]$$
$$[!A] = [?A] = \mathcal{M}_{\mathrm{fin}}([A])$$

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.

$$\frac{a \in [A]}{\vdash_{\mathsf{r}} a : A^\bot, a : A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A \qquad \vdash_{\mathsf{r}} a : A^\bot, \Psi}{\vdash_{\mathsf{r}} \Phi, \Psi}$$

$$\frac{}{\vdash_{\mathsf{r}} * : 1} \qquad \frac{\vdash_{\mathsf{r}} \Phi}{\vdash_{\mathsf{r}} \Phi, * : \bot}$$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A \qquad \vdash_{\mathsf{r}} \Psi, b : B}{\vdash_{\mathsf{r}} \Phi, \Psi, (a, b) : A \otimes B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A, b : B}{\vdash_{\mathsf{r}} \Phi, (a, b) : A \parr B}$$

No rule for 0 No rule for $\top$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, (1, a) : A \oplus B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, b : B}{\vdash_{\mathsf{r}} \Phi, (2, b) : A \oplus B}$$

$$\frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, (1, a) : A \,\&\, B} \qquad \frac{\vdash_{\mathsf{r}} \Phi, b : B}{\vdash_{\mathsf{r}} \Phi, (2, b) : A \,\&\, B}$$

$$\frac{\vdash_{\mathsf{r}} \Phi}{\vdash_{\mathsf{r}} \Phi, [] : ?A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, l : ?A, r : ?A}{\vdash_{\mathsf{r}} \Phi, l + r : ?A} \qquad \frac{\vdash_{\mathsf{r}} \Phi, a : A}{\vdash_{\mathsf{r}} \Phi, [a] : ?A}$$

$$\frac{\left(\vdash_{\mathsf{r}} m_1^j : ?A_1, \ldots m_k^j : ?A_k, b^j : B\right)_{j=1}^n}{\vdash_{\mathsf{r}} \sum_{j=1}^n m_1^j : ?A_1, \ldots, \sum_{j=1}^n m_k^j : ?A_k, [b^1, \ldots, b^n] : !B}$$

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

$$\frac{\begin{array}{c}\vdots\, \pi_1 \\ \vdash \Gamma_1, A\end{array} \qquad \begin{array}{c}\vdots\, \pi_2 \\ \vdash A^\perp, \Gamma_2\end{array}}{\vdash \Gamma_1, \Gamma_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\frac{\begin{array}{c}\vdots\, \theta_1 \\ \vdash_{\mathsf{r}} \Phi_1, a : A\end{array} \qquad \begin{array}{c}\vdots\, \theta_2 \\ \vdash_{\mathsf{r}} a : A^\perp, \Phi_2\end{array}}{\vdash_{\mathsf{r}} \Phi_1, \Phi_2}$$

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

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma\end{array}}{\vdash \Gamma, \bot}$$

then $\mathcal{T}(\pi)$ is set et of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi\end{array}}{\vdash_{\mathsf{r}} \Phi, * : \bot}$$

such that $\theta_1 \in \mathcal{T}(\pi_1)$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma_1, A_1\end{array} \qquad \begin{array}{c}\vdots\,\pi_2\\ \vdash \Gamma_2, A_2\end{array}}{\vdash \Gamma_1, \Gamma_2, A_1 \otimes A_2}$$

then $\mathcal{T}(\pi)$ is the set of all proofs

$$\dfrac{\begin{array}{c}\vdots\,\theta_1\\ \vdash_{\mathsf{r}} \Phi_1, a_1 : A_1\end{array} \qquad \begin{array}{c}\vdots\,\theta_2\\ \vdash_{\mathsf{r}} \Phi_2, a_2 : A_2\end{array}}{\vdash_{\mathsf{r}} \Phi_1, \Phi_2, (a_1, a_2) : A_1 \otimes A_2}$$

with $\theta_i \in \mathcal{T}(\pi_i)$ for $i = 1, 2$.

If π is the proof

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \vdash \Gamma, A_1, A_2\end{array}}{\vdash \Gamma, A_1 \parr A_2}$$

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

$$\dfrac{\begin{array}{c}\vdots\ \pi_1\\ \vdash \Gamma, A_1\end{array} \qquad \begin{array}{c}\vdots\ \pi_2\\ \vdash \Gamma, A_2\end{array}}{\vdash \Gamma, A_1 \ \&\ A_2}$$

then $\mathcal{T}(\pi)$ is the set of all resource derivations

$$\dfrac{\begin{array}{c}\vdots\ \theta\\ \vdash_{\mathsf{r}} \Phi, a : A_i\end{array}}{\vdash_{\mathsf{r}} \Phi, (i, a) : A_1 \ \&\ A_2}$$

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

$$\dfrac{\begin{array}{c}\vdots\ \theta_1\\ \vdash_{\mathsf{r}} m_1^1 : ?A_1, \ldots, m_k^1 : ?A_k, b^1 : B\end{array} \quad \cdots \quad \begin{array}{c}\vdots\ \theta_n\\ \vdash_{\mathsf{r}} m_1^n : ?A_1, \ldots, m_k^n : ?A_k, b^n : B\end{array}}{\vdash_{\mathsf{r}} \sum_{j=1}^n m_1^j : ?A_1, \ldots, \sum_{j=1}^n m_k^j : ?A_k, [b^1, \ldots, b^n] : !B}$$

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 of π reduces to π' by cut-elimination then $[\pi] = [\pi']$. We will present now the same relational semantics, but in a categorical way.

4.2 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

$$(R \cdot x) \cap y' = \mathsf{pr}_2(R \cap (x \times y')) \quad \text{and} \quad (R^\perp \cdot y') \cap x = \mathsf{pr}_1(R \cap (x \times y')) \tag{4.1}$$

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 4.2.1. *An isomorphism in* **Rel** *is a relation which is a bijection.*

The symmetric monoidal structure of **Rel** is 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 $X_1 \mathbin{\&} X_2$ 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 $\langle R_1, R_2 \rangle \in \mathbf{Rel}(Y, X_1 \mathbin{\&} X_2)$ 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 5.6), 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 $\mathsf{d}_X = \{([a],a) \mid a \in X\}$;
- $\mathsf{p}_X = \{(m_1 + \cdots + m_n, [m_1,\dots,m_n]) \mid n \in \mathbf{N} \text{ and } m_1,\dots,m_n \in !X\}$.

The monoidality isomorphism $\mathsf{m}^2_{X,Y} \in \mathbf{Rel}(!X \otimes !Y, !(X \mathbin{\&} Y))$ is the bijection which maps $([a_1,\dots,a_l],[b_1,\dots,b_r])$ to $[(1,a_1),\dots,(1,a_l),(2,b_1),\dots,(2,b_r)]$. 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

$$\begin{aligned}\mu^0 &= \{(*, k[*]) \mid k \in \mathbf{N}\}\\ \mu^2_{X,Y} &= \{(([a_1, \dots, a_n], [b_1, \dots, b_n]), [(a_1, b_1), \dots, (a_n, b_n)])\\ &\qquad \mid a_1, \dots, a_n \in X \text{ and } b_1, \dots, b_n \in Y\}\,.\end{aligned}$$

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

$$\begin{aligned}R^! = \{(m^1_1 + \cdots + m^1_n, \dots, m^k_1 + \cdots + m^k_n, [b_1, \dots, b_n]) \mid\\ \forall i \in \{1, \dots, n\}\ (m^1_i, \dots, m^k_i, b_i) \in R\}\,.\end{aligned}$$

Applying the general interpretation of Section **??** and **??**, we can interpret any proof structure p such that $\vdash p : A_1, \dots, A_k$ 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 $[!A] = [?A] = \mathcal{M}_{\mathrm{fin}}([A])$.

Remark 4.2.2. 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 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

[1] Of which many excellent presentation can be found in the litterature, startingof course with [**Girard**].

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 $a \coh_{E^\perp} a'$ if $a = a'$ or $\neg(a \coh_E a')$; 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.

4.3 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 and antireflexive relations on $|E|$ called *strict coherence* and *strict incoherence* respectively. The binary relation ν_E on $|E|$ 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: $\coh_E = \frown_E \cup \nu_E$ (large coherence) and $\asymp_E = \smile_E \cup \nu_E$ (large incoherence) which are symmetric relations on $|E|$. Notice that

[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 5.6.5.

[3] We drop the "non-uniform" in the sequel.

[4] This uncoupling of equality and coherence on $|E|$ is the main feature of these *non-uniform* coherence spaces.

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 $\frown_E \cap \smile_E = \emptyset$;
- $\coh_E$ and $\frown_E$, two symmetric relations such that $\frown_E \subseteq \coh_E$ and then $\smile_E = |E|^2 \setminus \coh_E$;
- $\coh_E$ and ν_E with $\nu_E \subseteq \coh_E$ and then $\frown_E = \coh_E \setminus \nu_E$ and $\smile_E = |E|^2 \setminus \coh_E$;
- $\frown_E$ 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 $\forall a, a' \in u\ a \coh_E a'$ and we use $\mathrm{Cl}(E)$ 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 $u \in \mathrm{Cl}(E)$ and $u' \in \mathrm{Cl}(E^\perp)$, 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 $|E_1 \otimes E_2| = |E_1| \times |E_2|$, $(a_1, a_2) \coh_{E_1 \otimes E_2} (a'_1, a'_2)$ if $a_i \coh_{E_i} a'_i$ 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:

- $(a, b) \coh_{E \multimap F} (a', b')$ if $a \coh_E a' \Rightarrow b \coh_F b'$ 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 $(a, b) \coh_{E \multimap F} (a', b')$ is equivalent to $b\ \nu_F\ b' \Rightarrow a \in E a'$ and $b \smile_F b' \Rightarrow a \smile_F a'$, a characterization of $\coh_{E \multimap F}$ which will be quite useful when dealing with Boudes' the exponential.

The category **NCoh** has coherence spaces as objects, and $\mathbf{NCoh}(E, F) = \mathrm{Cl}(E \multimap F)$. 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 well as linear curryfication, defined as in **Rel**). It is also *-autonomous with dualizing object $\perp = 1$, the dual of E being $E^\perp$ (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 $(i, a) \coh_{E_1 \& E_2} (j, b)$ 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 $|!E| = \mathcal{M}_{\mathrm{fin}}(|E|) = !|E|$ (this latter exponential being taken that of **Rel**.

4.3.1 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 $!_{\mathsf{b}} E = \mathcal{M}_{\mathrm{fin}}(|E|)$. Given $m, m' \in \mathcal{M}_{\mathrm{fin}}(|E|)$:

- $m \, \nu_{!_{\mathsf{b}} E} \, m'$ if $m = [a_1, \ldots, a_n]$ and $m' = [a'_1, \ldots, a'_n]$ with $\forall i \; a_i \, \nu_E \, a'_i$ and $\forall i, j \; a_i \coh_E a'_j$
- 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 $(a^1_{j^1}, b^1_{j^1}) \coh_{E \multimap F} (a^2_{j^2}, b^2_{j^2})$ since $R \in \mathrm{Cl}(E \multimap F)$, 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 $b^1_{j^1} \coh_F b^2_{j^2}$ 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 $R \in \mathrm{Cl}(E \multimap F)$). 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 $\forall j^1, j^2 \in \{1, \ldots, n\} \; a^1_{j^1} \coh_E a^2_{j^2}$. 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 $j = 1,\dots,n$. So we can write $m^i_j = [a^i_{j,1},\dots,a^i_{j,k_j}]$ with $a^1_{j,l} \; \nu_E \; a^2_{j,l}$ for all $j = 1,\dots,n$ and $l = 1,\dots,k_j$ (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 $m^i = [a^i_{j,l} \mid j \in \{1,\dots n\}$ and $l \in \{1,\dots,k_j\}]$). 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 4.2 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 $\mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \mathbin{\&} E_2))$. Given $m = [a_1,\dots,a_n] \in |!_{\mathsf{b}}E_i| = \mathcal{M}_{\mathrm{fin}}(|E_i|)$, let $i \cdot m = [(i,a_1),\dots,(i,a_n)] \in |!_{\mathsf{b}}(E_1 \mathbin{\&} E_2)|$. 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 $E_1 \mathbin{\&} E_2$, 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 $\mathsf{m}^2_{E_1,E_2} \in \mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \mathbin{\&} E_2))$.

We also need to prove that $\left(\mathsf{m}^2_{|E_1|,|E_1|}\right)^{-1} \in \mathbf{NCoh}(!_{\mathsf{b}}(E_1 \mathbin{\&} E_2), !_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2)$ where $\left(\mathsf{m}^2_{|E_1|,|E_1|}\right)^{-1}$ 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 4.3.1. *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 $E_1 \mathbin{\&} E_2$. 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 $(\mathbf{NCoh}, !_{\mathsf{b}}\_)$ (and probably also $(\mathbf{NCohB}, !_{\mathsf{b}}\_)$) is a Lafont model of LL; it has been discovered when looking for such a free exponential in **NCoh**).

4.3.2 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 $|\mathcal{M}_{\mathrm{fin}}(|E|)|$:

- $m \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} m'$ if $\forall i, j\ i \neq j \Rightarrow a_i \frown\!\!\!\!\smile_E a_j$ (we say that $m + m'$ is a *multiclique*)
- $m \frown_{!_{\mathsf{be}}E} m'$ if $m \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} m'$ 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).

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 $m \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} m'$, that is, $m + m'$ is a multiclique (that is $i \neq j \Rightarrow a_i \frown\!\!\!\!\smile_E a_j$). Since $\forall i, j\ (a_i, b_i) \frown\!\!\!\!\smile_{E \multimap F} (a_j, b_j)$, 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

$$\forall a, a' \in |E| \quad [a] \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} [a'] \Leftrightarrow a \frown\!\!\!\!\smile_E a' \text{ and } [a] \frown_{!_{\mathsf{be}}E} [a'] \Leftrightarrow a \frown_E a' \,.$$

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 $m \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} m'$, 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 $\forall i < j\ m_i \frown\!\!\!\!\smile_{!_{\mathsf{be}}E} m_j$. 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 $\mathsf{m}^2_{|E_1|,|E_2|} \in \mathbf{NCoh}(!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2, !_{\mathsf{b}}(E_1 \mathbin{\&} E_2))$ (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

$$(m_1, m_2) \mathrel{\frown\!\!\!\!\smile}_{!_{\mathsf{be}}E_1 \otimes !_{\mathsf{be}}E_2} (m'_1, m'_2)\,,$$

meaning that $m_i + m'_i$ is a multiclique in E_i, for $i = 1, 2$. Let us write $m_i = [a^i_1, \dots, a^i_{k^i}]$, $m'_i = [a^i_{k^i+1}, \dots, a^i_{n^i}]$ for $i = 1, 2$. Remember that $m = 1{\cdot}m_1 + 2{\cdot}m_2$ and similarly for m'. Then

$$\begin{aligned} m + m' &= 1 \cdot (m_1 + m'_1) + 2 \cdot (m_2 + m'_2) \\ &= [(1, a^1_1), \dots, (1, a^1_{n^1}), (2, a^2_1), \dots, (2, a^2_{n^2})]\,. \end{aligned}$$

For $l \in \{1, n^1 + n^2\}$ let us set $b_l = (1, a^1_l)$ if $1 \le l \le n^1$ and $b_l = (2, a^2_{l-n^1})$ if $n^1 + 1 \le l \le n^2$ so that $m + m' = [b_1, \dots, b_{n^1+n^2}]$. It is clear that $m + m'$ a multiclique because the $m_i + m'_i$'s are and by definition of $E_1 \mathbin{\&} E_2$. 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, $l \in \{1, \dots, k^1\}$. Then we have $b_l \frown_{E_1 \& E_2} b_{l'}$ for $l' \in \{1, \dots, n^1\}$ because $m_1 + m'_1$ is star-shaped, and $b_l \frown_{E_1 \& E_2} b_{l'}$ for $l' \in \{n^1 + 1, \dots, n^1 + n^2\}$ by definition of $E_1 \mathbin{\&} E_2$.

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 $m \mathrel{\frown\!\!\!\!\smile}_{!_{\mathsf{be}}(E_1 \& E_2)} m'$, that is $m + m'$ is a clique in $E_1 \mathbin{\&} E_2$ which implies that $m_i + m'_i$ is a multiclique in E_i for $i = 1, 2$ and hence $(m_1, m_2) \mathrel{\frown\!\!\!\!\smile}_{!_{\mathsf{b}}E_1 \otimes !_{\mathsf{b}}E_2} (m'_1, m'_2)$. Assume moreover that $m + m'$ is star-shaped with $l \in \{1, \dots, n^1 + n^2\}$ as center. By symmetry we can assume that $l \in \{1, \dots, k^1\}$. 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.

4.3.2.1 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 $[*, \dots, *]$ 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 $n, n' \in \mathbf{N}$:

- $\{((n, *), (1, *)), ((n', *), (2, *))\} \in \mathrm{Cl}((!_{\mathsf{b}}1 \multimap 1) \multimap 1 \oplus 1)$ 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 $\{((n,*),(1,*)),((n',*),(2,*))\} \in \mathrm{Cl}((!_{\mathsf{be}}1 \multimap 1) \multimap 1 \oplus 1)$ 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

$$\{([(1,*),(2,*)],(1,*)),([(1,*),(2,*)],(2,*))\} \in \mathbf{NCoh}(!_{\mathsf{b}}(1 \oplus 1), 1 \oplus 1)$$

Chapter 5

Categories

5.1 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 5.1.1. 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 5.1.2. 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.

5.1.1 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 $f \in \mathcal{C}(X',X)$.

5.1.2 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 5.1.3. 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*.

5.2 Limits and colimits

5.2.1 Projective limits (limits)

5.2.1.1 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).

5.2.1.2 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 5.2.1. *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 5.2.2. *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 5.2.3. 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 $(X_1, X_2) \mapsto X_1 \mathbin{\&} X_2$ can be turned into a functor $\mathcal{C}^2 \to \mathcal{C}$ by Proposition 5.2.2: let $f_i \in \mathcal{C}(X_i, Y_i)$ for $i = 1, 2$. We have $f_i \circ \mathsf{pr}_i \in \mathcal{C}(X_1 \mathbin{\&} X_2, X_i)$ and hence there is a unique morphism $f_1 \mathbin{\&} f_2 \in \mathcal{C}(X_1 \mathbin{\&} X_2, Y_i)$ such that $\mathsf{pr}_i \circ (f_1 \mathbin{\&} f_2) = f_i \circ \mathsf{pr}_i$ for $i = 1, 2$ and the operation which maps (X_1, X_2) to $X_1 \mathbin{\&} X_2$ and $(f_1, f_2) \in \mathcal{C}(X_1, Y_1) \times \mathcal{C}(X_2, Y_2)$ to $f_1 \mathbin{\&} f_2$ is a functor.

Example 5.2.4. 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.

5.2.1.3 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 $e \in \mathcal{C}(E \mathbin{\&} X, Y)$ are such that, for any $Z \in \mathsf{Obj}(\mathcal{C})$ and for any $f \in \mathcal{C}(Z \mathbin{\&} X, Y)$ 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:

$$\begin{aligned} \mathsf{Ev} \circ (\mathsf{Cur}(f) \,\&\, \mathsf{Id}_X) &= f \\ \mathsf{Cur}(f) \circ g &= \mathsf{Cur}(f \circ (g \,\&\, \mathsf{Id}_X)) \quad \text{where } g \in \mathcal{C}(Z', Z) \\ \mathsf{Cur}(\mathsf{Ev}) &= \mathsf{Id}_{X \Rightarrow Y}\,. \end{aligned} \tag{5.1}$$

It can be easier to check these equations than directly the universal property.

Exercise 5.2.5. 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 5.2.6. Prove that **Set** is cartesian closed and that **Rel** has all small products but is not cartesian closed.

5.2.1.4 Inductive limits (colimits)

The definition of colimit (or inductive limits, or direct 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 $f \in \mathcal{C}(\oplus_{i\in I} X_i, X)$ 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.

5.3 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 $\Phi : \mathsf{Hom}_{\mathcal{D}} \circ (L \times \mathsf{Id}_{\mathcal{D}}) \stackrel{\bullet}{\to} \mathsf{Hom}_{\mathcal{C}} \circ (\mathsf{Id}_{\mathcal{C}} \times R)$ 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

$$\begin{aligned} \eta_A &= \Phi_{A,L(A)}(\mathsf{Id}_{L(A)}) \in \mathcal{C}(A, RL(A)) \\ \text{and} \quad \varepsilon_X &= \Phi^{-1}_{R(X),X}(\mathsf{Id}_{R(X)}) \in \mathcal{D}(LR(X), X) \end{aligned}$$

called respectively *unit* and *counit* of the adjunction. They are natural transformations $\eta : \mathsf{Id}_{\mathcal{C}} \stackrel{\bullet}{\to} RL$ and $\varepsilon : LR \stackrel{\bullet}{\to} \mathsf{Id}_{\mathcal{D}}$ and satisfy the following equations:

$$\begin{aligned} R(\varepsilon_X) \circ \eta_{R(X)} &= \mathsf{Id}_{R(X)} \\ \varepsilon_{L(A)} \circ L(\eta_A) &= \mathsf{Id}_{L(A)} \ . \end{aligned}$$

Moreover, the data of two functors $L : \mathcal{C} \to \mathcal{D}$, $R : \mathcal{D} \to \mathcal{C}$ and two natural transformations $\eta : \mathsf{Id}_{\mathcal{C}} \stackrel{\bullet}{\to} RL$ and $\varepsilon : LR \stackrel{\bullet}{\to} \mathsf{Id}_{\mathcal{D}}$ satisfying the equations above induce uniquely an adjunction of which these two natural transformations are the unit and counit.

5.4 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..

$$
\begin{array}{ccc}
T(X) & \xrightarrow{\varepsilon_{T(X)}} & T^2(X) \\
& \searrow^{\mathsf{Id}_{T(X)}} & \downarrow \mu_X \\
& & T(X)
\end{array}
\qquad
\begin{array}{ccc}
T(X) & \xrightarrow{T(\varepsilon_X)} & T^2(X) \\
& \searrow^{\mathsf{Id}_{T(X)}} & \downarrow \mu_X \\
& & T(X)
\end{array}
\qquad
\begin{array}{ccc}
T^3(X) & \xrightarrow{T(\mu_X)} & T^2(X) \\
\mu_{T(X)} \downarrow & & \downarrow \mu_X \\
T^2(X) & \xrightarrow{\mu_X} & T(X)
\end{array}
$$

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.

$$
\begin{array}{ccc}
X & \xrightarrow{\varepsilon_X} & T(X) \\
& \searrow^{\mathsf{Id}_X} & \downarrow h \\
& & X
\end{array}
\qquad
\begin{array}{ccc}
T^2(X) & \xrightarrow{T(h)} & T(X) \\
\mu_X \downarrow & & \downarrow h \\
T(X) & \xrightarrow{h} & X
\end{array}
$$

The elements of $\mathcal{C}^T((X, h), (Y, k))$ are the $f \in \mathcal{C}(X, Y)$ such that the following diagram commutes.

$$
\begin{array}{ccc}
T(X) & \xrightarrow{h} & X \\
T(f) \downarrow & & \downarrow f \\
T(Y) & \xrightarrow{k} & Y
\end{array}
$$

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 $\mathrm{Id}^{\mathsf{K}}_X = \varepsilon_X$ 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 5.4.1. Prove that we have defined a category $\mathcal{C}_T$.

Exercise 5.4.2. 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 5.4.3. Let $M : \mathbf{Set} \to \mathbf{Set}$ 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 5.4.4. 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 $\mathbf{Rel}$.

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 $\mathbf{Rel}$ 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:

$$\begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ & {\scriptstyle \mathsf{Id}_{S(X)}} \searrow & \downarrow {\scriptstyle \delta_{S(X)}} \\ & & S(X) \end{array} \qquad \begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ & {\scriptstyle \mathsf{Id}_{S(X)}} \searrow & \downarrow {\scriptstyle S(\delta_X)} \\ & & S(X) \end{array} \qquad \begin{array}{ccc} S(X) & \xrightarrow{\lambda_X} & S^2(X) \\ {\scriptstyle \lambda_X} \downarrow & & \downarrow {\scriptstyle \lambda_{S(X)}} \\ S^2(X) & \xrightarrow{S(\lambda_X)} & S^3(X) \end{array}$$

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

$$\begin{array}{ccc} X & \xrightarrow{h} & S(X) \\ & {\scriptstyle \mathsf{Id}_X} \searrow & \downarrow {\scriptstyle \delta_X} \\ & & X \end{array} \qquad \begin{array}{ccc} X & \xrightarrow{h} & S(X) \\ {\scriptstyle h} \downarrow & & \downarrow {\scriptstyle \lambda_X} \\ S(X) & \xrightarrow{S(h)} & S^2(X) \end{array}$$

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

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \downarrow {\scriptstyle h} & & \downarrow {\scriptstyle k} \\ S(X) & \xrightarrow{S(f)} & S(Y) \end{array}$$

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 5.4.5. 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 5.4.6. *The function*

$$\begin{aligned} \varphi_{X,Y} : \mathcal{C}^S((S(X), \lambda_X), (S(Y), \lambda_Y)) &\to \mathcal{C}(S(X), Y) \\ f &\mapsto \delta_Y \circ f \end{aligned}$$

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. $\square$

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 5.4.6 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.

5.5 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.

$$\begin{array}{ccc} (\mathsf{I} \boxtimes X_1) \boxtimes X_2 & \xrightarrow{\alpha^{\boxtimes}_{\mathsf{I},X_1,X_2}} & \mathsf{I} \boxtimes (X_1 \boxtimes X_2) \\ & \searrow^{\lambda^{\boxtimes}_{X_1} \boxtimes X_2} & \downarrow{\scriptstyle \lambda^{\boxtimes}_{X_1 \boxtimes X_2}} \\ & & X_1 \boxtimes X_2 \end{array}$$

$$\begin{array}{ccc} (X_1 \boxtimes \mathsf{I}) \boxtimes X_2 & \xrightarrow{\alpha^{\boxtimes}_{X_1,\mathsf{I},X_2}} & X_1 \boxtimes (\mathsf{I} \boxtimes X_2) \\ & \searrow^{\rho^{\boxtimes}_{X_1} \boxtimes X_2} & \downarrow{\scriptstyle X_1 \boxtimes \lambda^{\boxtimes}_{X_2}} \\ & & X_1 \boxtimes X_2 \end{array}$$

$$\begin{array}{ccc} (X_1 \boxtimes X_2) \boxtimes \mathsf{I} & \xrightarrow{\alpha^{\boxtimes}_{X_1,X_2,\mathsf{I}}} & X_1 \boxtimes (X_2 \boxtimes \mathsf{I}) \\ & \searrow^{\rho^{\boxtimes}_{X_1 \boxtimes X_2}} & \downarrow{\scriptstyle X_1 \boxtimes \rho^{\boxtimes}_{X_2}} \\ & & X_1 \boxtimes X_2 \end{array}$$

$$\begin{array}{ccc} ((X_1 \boxtimes X_2) \boxtimes X_3) \boxtimes X_4 & \xrightarrow{\alpha^{\boxtimes}_{X_1 \boxtimes X_2, X_3, X_4}} & (X_1 \boxtimes X_2) \boxtimes (X_3 \boxtimes X_4) \\ {\scriptstyle \alpha^{\boxtimes}_{X_1,X_2,X_3} \boxtimes X_4} \downarrow & & \\ (X_1 \boxtimes (X_2 \boxtimes X_3)) \boxtimes X_4 & & \downarrow {\scriptstyle \alpha^{\boxtimes}_{X_1,X_2,X_3 \boxtimes X_4}} \\ {\scriptstyle \alpha^{\boxtimes}_{X_1,X_2 \boxtimes X_3,X_4}} \downarrow & & \\ X_1 \boxtimes ((X_2 \boxtimes X_3) \boxtimes X_4) & \xrightarrow{X_1 \boxtimes \alpha^{\boxtimes}_{X_2,X_3,X_4}} & X_1 \boxtimes (X_2 \boxtimes (X_3 \boxtimes X_4)) \end{array}$$

$$\begin{array}{ccc} \mathsf{I} \boxtimes X & \xrightarrow{\sigma^{\boxtimes}_{\mathsf{I},X}} & X \boxtimes \mathsf{I} \\ & \searrow^{\lambda^{\boxtimes}_{X}} & \downarrow{\scriptstyle \rho^{\boxtimes}_{X}} \\ & & X \end{array}$$

$$\begin{array}{ccc} (X_1 \boxtimes X_2) \boxtimes X_3 & \xrightarrow{\alpha^{\boxtimes}_{X_1,X_2,X_3}} & X_1 \boxtimes (X_2 \boxtimes X_3) \\ {\scriptstyle \sigma^{\boxtimes}_{X_1,X_2} \boxtimes X_3} \downarrow & & \downarrow {\scriptstyle \sigma^{\boxtimes}_{X_1,X_2 \boxtimes X_3}} \\ (X_2 \boxtimes X_1) \boxtimes X_3 & & (X_2 \boxtimes X_3) \boxtimes X_1 \\ {\scriptstyle \alpha^{\boxtimes}_{X_2,X_1,X_3}} \downarrow & & \downarrow {\scriptstyle \alpha^{\boxtimes}_{X_2,X_3,X_1}} \\ X_2 \boxtimes (X_1 \boxtimes X_3) & \xrightarrow{X_2 \boxtimes \sigma^{\boxtimes}_{X_1,X_3}} & X_2 \boxtimes (X_3 \boxtimes X_1) \end{array}$$

$$\begin{array}{ccc} X_1 \boxtimes X_2 & \xrightarrow{\sigma^{\boxtimes}_{X_1,X_2}} & X_2 \boxtimes X_1 \\ & \searrow^{X_1 \boxtimes X_2} & \downarrow{\scriptstyle \sigma^{\boxtimes}_{X_2,X_1}} \\ & & X_1 \boxtimes X_2 \end{array}$$

We define a notion of "monoidal tree" by the following syntax:

$$\tau, \tau_1, \dots := * \mid n \mid \langle \tau_1, \tau_2 \rangle$$

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 $\{1, \dots, n\}$.

Given a well-formed monoidal tree τ of degree n and a sequence $\overrightarrow{X} = (X_1, \dots, X_k)$ 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 $\overrightarrow{X} = (X_1, \dots, X_n)$ 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*.

Exercise 5.5.1. Prove that any cartesian category has a canonical structure of monoidal category, with $\&$ as monoidal bifunctor.

5.5.1 Commutative comonoids

Definition 5.5.2. 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

$$\begin{array}{ccc}
\underline{C} & \xrightarrow{f} & \underline{D} \\
& {\scriptstyle \mathsf{w}_C}\searrow & \downarrow{\scriptstyle \mathsf{w}_D} \\
& & 1
\end{array}
\qquad
\begin{array}{ccc}
\underline{C} & \xrightarrow{\quad f\quad} & \underline{D} \\
{\scriptstyle \mathsf{c}_C}\downarrow & & \downarrow{\scriptstyle \mathsf{c}_D} \\
\underline{C}\otimes\underline{C} & \xrightarrow{f\otimes f} & \underline{D}\otimes\underline{D}
\end{array}$$

Theorem 5.5.3. *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*

$$\underline{C_0}\otimes\underline{C_1} \xrightarrow{\quad\mathsf{w}_{C_0}\otimes\mathsf{w}_{C_1}\quad} 1\otimes 1 \xrightarrow{\quad\quad\lambda_1\quad\quad} 1$$

$$\underline{C_0}\otimes\underline{C_1} \xrightarrow{\mathsf{c}_{C_0}\otimes\mathsf{c}_{C_1}} \underline{C_0}\otimes\underline{C_0}\otimes\underline{C_1}\otimes\underline{C_1} \xrightarrow{\sigma_{2,3}} \underline{C_0}\otimes\underline{C_1}\otimes\underline{C_0}\otimes\underline{C_1}$$

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 5.5.4. *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. □

5.6 Seely categories

We define now the basic notion of categorical model of Linear Logic. A Seely category is a tuple

$$(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \perp, !, \mathsf{d}, \mathsf{p}, \mathsf{m}^0, \mathsf{m}^2)$$

where $\mathcal{L}$ is a category and the additional components will be explained now.

5.6.1 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 5.5.

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

$$\begin{array}{ccc} Z \otimes X & \xrightarrow{\mathsf{cur} f \otimes X} & (X \multimap Y) \otimes X \\ & \searrow^{f} & \downarrow \mathsf{ev} \\ & & Y \end{array}$$

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

$$\begin{array}{ccc} Z \otimes X & \xrightarrow{g \otimes X} & Z' \otimes X \\ & \searrow^{f} & \downarrow f' \\ & & Y \end{array}$$

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

$$\begin{array}{ccccc}
(X \multimap Y) \otimes X' & \xrightarrow{(f \multimap g) \otimes X'} & & & (X' \multimap Y') \otimes X' \\
\downarrow{\scriptstyle (X \multimap Y) \otimes f} & & & & \downarrow{\scriptstyle \mathsf{ev}} \\
(X \multimap Y) \otimes X & \xrightarrow{\mathsf{ev}} & Y & \xrightarrow{g} & Y'
\end{array}$$

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

$$\begin{array}{ccccc}
(X \multimap Y) \otimes X'' & \xrightarrow{h \otimes X''} & & & (X'' \multimap Y'') \otimes X'' \\
\downarrow{\scriptstyle (X \multimap Y) \otimes (f'\,f)} & & & & \downarrow{\scriptstyle \mathsf{ev}} \\
(X \multimap Y) \otimes X & \xrightarrow{\mathsf{ev}} & Y & \xrightarrow{g'\,g} & Y'
\end{array}$$

by functoriality of $\otimes$ and hence they are equal.

Theorem 5.6.1. *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 $X \multimap \_$.

5.6.1.1 $*$-autonomy

Let Z be an object of $\mathcal{L}$. For any object X, we have 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 5.6.2. *The following composition of morphisms*

$$X \multimap Z \xrightarrow{\eta_{X \multimap Z}^{(Z)}} ((X \multimap Z) \multimap Z) \multimap Z \xrightarrow{\eta_X^{(Z)} \multimap Z} X \multimap Z$$

coincides with the identity at $X \multimap Z$.

Proof. We have

$$\begin{aligned}(\eta_X^{(Z)} \multimap Z)\, \eta_{X\multimap Z}^{(Z)} &= \mathsf{cur}(\mathsf{ev}\,(\mathsf{Id} \otimes \eta_X^{(Z)}))\, \eta_{X\multimap Z}^{(Z)} \\ &= \mathsf{cur}(\mathsf{ev}(\eta_{X\multimap Z}^{(Z)} \otimes \eta_X^{(Z)})) \\ &= \mathsf{cur}(\mathsf{ev}\,(\eta_{X\multimap Z}^{(Z)} \otimes \mathsf{Id})\,(\mathsf{Id} \otimes \eta_X^{(Z)})) \\ &= \mathsf{cur}(\mathsf{ev}\;\sigma\,(\mathsf{Id} \otimes \eta_X^{(Z)})) \\ &= \mathsf{cur}(\mathsf{ev}\,(\eta_X^{(Z)} \otimes \mathsf{Id})\,\sigma) \\ &= \mathsf{cur}(\mathsf{ev}\;\sigma\,\sigma) = \mathsf{Id}\ .\end{aligned}$$

□

Definition 5.6.3. 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 $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \perp)$ where $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma)$ is an SMCC and $\perp \in \mathsf{Obj}(\mathcal{L})$ is a dualizing object.

In the definition of a Seely category, the structure $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \perp)$ is assumed to be a $*$-autonomous category. We use then the notation $\_^{\perp}$ for the functor $\_ \multimap \perp : \mathcal{L}^{\mathsf{op}} \to \mathcal{L}$. We set $\eta_X = \eta_X^{(\perp)}$ so that for each object X of $\mathcal{L}$, $\eta_X \in \mathcal{L}(X, X^{\perp\perp})$ is an iso which is natural in X.

Lemma 5.6.4. *Let X be an object of $\mathcal{L}$, we have $\eta_X^{\perp}\, \eta_{X^{\perp}} = \mathsf{Id}_{X^{\perp}}$.*

This is just a special case of Lemma 5.6.2.

5.6.2 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 $(X_1 \,\&\, X_2, \mathsf{pr}_1, \mathsf{pr}_2)$. In most known models countable cartesian products are also available and we denote 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 5.6.5. *If a $*$-autonomous category $(\mathcal{L}, 1, \otimes, \lambda, \rho, \alpha, \sigma, \perp)$ 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

$$(\bigoplus_{i\in I} X_i, (\mathsf{in}_i)_{i\in I}) = ((\mathop{\&}_{i\in I} X_i^\perp)^\perp, (\mathsf{pr}_i^\perp\ \eta_{X_i})_{i\in I})$$

is the coproduct of the X_i's; notice indeed that $\mathsf{in}_j = \mathsf{pr}_j^\perp\ \eta_{X_j} \in \mathcal{L}(X_j, \oplus_{i\in I} X_i)$.

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 $f_i^\perp \in \mathcal{L}(Y^\perp, Y_i^\perp)$ and hence $\langle f_i^\perp\rangle_{i\in I} \in \mathcal{L}(Y^\perp, \&_{i\in I} X_i^\perp)$. Therefore we set

$$[f_i]_{i\in I} = {\eta_Y}^{-1}\ \langle f_i^\perp\rangle_{i\in I}^\perp \in \mathcal{L}(\bigoplus_{i\in I} X_i, Y)\,.$$

Given $j \in I$, we have

$$\begin{aligned}
[f_i]_{i\in I}\ \mathsf{in}_j &= {\eta_Y}^{-1}\ \langle f_i^\perp\rangle_{i\in I}^\perp\ \mathsf{pr}_j^\perp\ \eta_{X_j} \\
&= {\eta_Y}^{-1}\ (\mathsf{pr}_j\ \langle f_i^\perp\rangle_{i\in I})^\perp\ \eta_{X_j} \\
&= {\eta_Y}^{-1}\ f_j^{\perp\perp}\ \eta_{X_j} \\
&= f_j
\end{aligned}$$

by naturality of η. Notice that for all $i \in I$ we have

$$\begin{aligned}
\mathsf{in}_i^\perp &= \eta_{X_i}^\perp\ \mathsf{pr}_i^{\perp\perp} \\
&= {\eta_{X_i^\perp}}^{-1}\ \mathsf{pr}_i^{\perp\perp} && \text{by Lemma 5.6.4} \\
&= \mathsf{pr}_i\ (\eta_{\&_{i\in I} X_i^\perp})^{-1} && \text{by naturality of } \eta^{-1}
\end{aligned}$$

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 $f^{\perp\perp} = \langle f_j^\perp\rangle_{j\in I}^\perp\ \eta_{\&_{i\in I} X_i^\perp}^\perp$. By Lemma 5.6.4 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 $f = {\eta_Y}^{-1}\ \langle f_j^\perp\rangle_{j\in I}^\perp = [f_i]_{i\in I}$. Therefore $(\oplus_{i\in I} X_i, (\mathsf{in}_i)_{i\in I})$ is the coproduct of the X_i's in $\mathcal{L}$.

5.6.3 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

$$
\begin{array}{ccc}
!X & \xrightarrow{\mathsf{p}_X} & !!X \\
 & \searrow^{X} & \downarrow{\scriptstyle \mathsf{d}_{!X}} \\
 & & !X
\end{array}
\qquad
\begin{array}{ccc}
!X & \xrightarrow{\mathsf{p}_X} & !!X \\
 & \searrow^{X} & \downarrow{\scriptstyle !\mathsf{d}_X} \\
 & & !X
\end{array}
\qquad
\begin{array}{ccc}
!X & \xrightarrow{\mathsf{p}_X} & !!X \\
{\scriptstyle \mathsf{p}_X}\downarrow & & \downarrow{\scriptstyle !\mathsf{p}_X} \\
!!X & \xrightarrow{\mathsf{p}_{!X}} & !!!X
\end{array}
$$

The two last morphisms are the *Seely isomorphisms*: $\mathsf{m}^0 \in \mathcal{L}(1, !\top)$ is an isomorphism and $\mathsf{m}^2_{X_1,X_2} \in \mathcal{L}(!X_1 \otimes !X_2, !(X_1 \mathbin{\&} X_2))$ 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:

$$
\begin{array}{ccc}
(!X_1 \otimes !X_2) \otimes !X_3 & \xrightarrow{\alpha_{!X_1,!X_2,!X_3}} & !X_1 \otimes (!X_2 \otimes !X_3) \\
{\scriptstyle \mathsf{m}^2_{X_1,X_2} \otimes !X_3}\downarrow & & \downarrow{\scriptstyle !X_1 \otimes \mathsf{m}^2_{X_2,X_3}} \\
!(X_1 \mathbin{\&} X_2) \otimes !X_3 & & !X_1 \otimes !(X_2 \mathbin{\&} X_3) \\
{\scriptstyle \mathsf{m}^2_{X_1 \& X_2, X_3}}\downarrow & & \downarrow{\scriptstyle \mathsf{m}^2_{X_1, X_2 \& X_3}} \\
!((X_1 \mathbin{\&} X_2) \mathbin{\&} X_3) & \xrightarrow{\langle \mathsf{pr}_1\ \mathsf{pr}_1, \langle \mathsf{pr}_2\ \mathsf{pr}_1, \mathsf{pr}_2 \rangle\rangle} & !(X_1 \mathbin{\&} (X_2 \mathbin{\&} X_3))
\end{array}
$$

The second one deals with the commutators:

$$
\begin{array}{ccc}
!X_1 \otimes !X_2 & \xrightarrow{\sigma_{!X_1,!X_2}} & !X_2 \otimes !X_1 \\
{\scriptstyle \mathsf{m}^2_{X_1,X_2}}\downarrow & & \downarrow{\scriptstyle \mathsf{m}^2_{X_2,X_1}} \\
!(X_1 \mathbin{\&} X_2) & \xrightarrow{!\langle \mathsf{pr}_2, \mathsf{pr}_1 \rangle} & !(X_2 \mathbin{\&} X_1)
\end{array}
$$

And the last one deals with the neutrality isos:

$$
\begin{array}{ccc}
!X \otimes 1 & \xrightarrow{\rho_X} & !X \\
{\scriptstyle !X \otimes \mathsf{m}^0}\downarrow & & \downarrow{\scriptstyle !\langle X, \mathsf{t}_X \rangle} \\
!X \otimes !\top & \xrightarrow{\mathsf{m}^2_{X,\top}} & !(X \mathbin{\&} \top)
\end{array}
\qquad
\begin{array}{ccc}
1 \otimes !X & \xrightarrow{\lambda_X} & !X \\
{\scriptstyle \mathsf{m}^0 \otimes !X}\downarrow & & \downarrow{\scriptstyle !\langle \mathsf{t}_X, X \rangle} \\
!\top \otimes !X & \xrightarrow{\mathsf{m}^2_{\top,X}} & !(\top \mathbin{\&} X)
\end{array}
$$

The compatibility of this symmetric monoidal structure with the comonad structure of $!\_$ is expressed by the following diagram

$$
\begin{array}{ccc}
!X_1 \otimes !X_2 & \xrightarrow{\mathsf{m}^2_{X,Y}} & !(X_1 \mathbin{\&} Y) \\
 & & \downarrow{\scriptstyle \mathsf{p}_{X_1 \& X_2}} \\
{\scriptstyle \mathsf{p}_{X_1} \otimes \mathsf{p}_{X_2}}\downarrow & & !!(X_1 \mathbin{\&} X_2) \\
 & & \downarrow{\scriptstyle !\langle !\mathsf{pr}_1, !\mathsf{pr}_2 \rangle} \\
!!X_1 \otimes !!X_2 & \xrightarrow{\mathsf{m}^2_{!X_1,!X_2}} & !(!X_1 \mathbin{\&} !X_2)
\end{array}
$$

Definition 5.6.6. A *Seely category* is a cartesian $*$-autonomous category equipped with an exponential structure.

5.6.4 Derived structures in a Seely category

5.6.4.1 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 $f^! \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, !Y)$.

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:

$$!X_1 \otimes !X_2 \xrightarrow{\mathsf{p}_{X_1} \otimes \mathsf{p}_{X_2}} !!X_1 \otimes !!X_2 \xrightarrow{\mathsf{m}^2_{!X_1,!X_2}} !(!X_1 \mathbin{\&} !X_2) \xrightarrow{!(\mathsf{d}_{X_1} \& \mathsf{d}_{X_2})} !(X_1 \mathbin{\&} X_2)$$

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

$$\begin{array}{ccccc}
1 \otimes !X & \xrightarrow{\lambda_X} & !1 \otimes !X & \xrightarrow{\mu^2_{1,X}} & !(1 \otimes X) \\
& \searrow^{\lambda_{!X}} & & & \downarrow{!\lambda_X} \\
& & & & !X
\end{array}$$

$$\begin{array}{ccccc}
(!X_1 \otimes !X_2) \otimes !X_3 & \xrightarrow{\mu^2_{X_1,X_2} \otimes !X_3} & !(X_1 \otimes X_2) \otimes !X_3 & \xrightarrow{\mu^2_{X_1 \otimes X_2, X_3}} & !((X_1 \otimes X_2) \otimes X_3) \\
\downarrow{\alpha_{!A_1,!A_2,!A_3}} & & & & \downarrow{!\alpha_{X_1,X_2,X_3}} \\
!X_1 \otimes (!X_2 \otimes !X_3) & \xrightarrow{!X_1 \otimes \mu^2_{X_2,X_3}} & !X_1 \otimes !(X_2 \otimes X_3) & \xrightarrow{\mu^2_{X_1, X_2 \otimes X_3}} & !(X_1 \otimes (X_2 \otimes X_3))
\end{array}$$

$$\begin{array}{ccc}
!X_1 \otimes !X_2 & \xrightarrow{\mu^2_{X_1,X_2}} & !(X_1 \otimes X_2) \\
\downarrow{\sigma_{!X_1,!X_2}} & & \downarrow{!\sigma_{X_1,X_2}} \\
!X_2 \otimes !X_1 & \xrightarrow{\mu^2_{X_2,X_1}} & !(X_2 \otimes X_1)
\end{array}$$

Exercise 5.6.7. Prove that the three diagrams above commute.

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

[1]"lax" means that the associated natural transformations are not isos in general.

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).

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 $\mathcal{L}$

$$\begin{array}{c}
!X_1 \otimes \cdots \otimes !X_n \\
\downarrow {\scriptstyle \mathsf{p}_{X_1} \otimes \cdots \otimes \mathsf{p}_{X_n}} \\
!!X_1 \otimes \cdots \otimes !!X_n \\
\downarrow {\scriptstyle \mu^n_{!X_1, \dots, !X_n}} \\
!(!X_1 \otimes \cdots \otimes !X_n) \\
\downarrow {\scriptstyle !f} \\
!Y
\end{array}$$

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 $f^! \in \mathcal{L}(1, !Y)$.

Let $f \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, Y)$. The two following diagrams commute.

$$\begin{array}{ccc}
!X_1 \otimes \cdots \otimes !X_n & \xrightarrow{f^!} & !Y \\
& {\scriptstyle f} \searrow & \downarrow {\scriptstyle \mathsf{d}_Y} \\
& & Y
\end{array}
\qquad
\begin{array}{ccc}
!X_1 \otimes \cdots \otimes !X_n & \xrightarrow{f^!} & !Y \\
& {\scriptstyle f^{!!}} \searrow & \downarrow {\scriptstyle \mathsf{p}_Y} \\
& & !!Y
\end{array}$$

Exercise 5.6.8. Prove these commutations.

Exercise 5.6.9. 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

$$\begin{array}{ccc}
!X_1 \otimes \cdots \otimes !X_p & \xrightarrow{!X_1 \otimes \cdots \otimes !X_n \otimes f^!} & !X_1 \otimes \cdots \otimes !X_n \otimes !Y \\
& {\scriptstyle (g\,(!X_1 \otimes \cdots \otimes !X_n \otimes f^!))^!} \searrow & \downarrow {\scriptstyle g^!} \\
& & !Z
\end{array}$$

5.6.4.2 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

$$!X \xrightarrow{!\langle \mathsf{pr}_1, \mathsf{pr}_2 \rangle} !(X \mathbin{\&} X) \xrightarrow{(\mathsf{m}^2_{X,X})^{-1}} !X \otimes !X$$

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 5.5.1.

5.6.4.3 The Kleisli category

We have defined in Section 5.4 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 thoes 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 5.6.10. *If $f \in \mathcal{L}_!(X, Y)$ and $g \in \mathcal{C}(Y, Z)$ then $\mathsf{Der}(g) \circ f = g\, f$.*

Theorem 5.6.11. *The category $\mathcal{L}_!$ is cartesian closed.*

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 $\mathcal{L}$.

Given objects X, Y of $\mathcal{L}$ we set

$$(X \Rightarrow Y) = (!X \multimap Y)$$

and define $\mathsf{Ev} \in \mathcal{L}_!((X \Rightarrow Y) \mathbin{\&} X, Y)$ as the following composition of morphisms in $\mathcal{L}$:

$$
\begin{array}{c}
!((!X \multimap Y) \mathbin{\&} X) \\
\Big\downarrow {\scriptstyle (\mathsf{m}^2_{!X \multimap Y, X})^{-1}} \\
!(!X \multimap Y) \otimes !X \\
\Big\downarrow {\scriptstyle \mathsf{d}_{!X \multimap Y} \otimes \mathsf{Id}} \\
(!X \multimap Y) \otimes !X \\
\Big\downarrow {\scriptstyle \mathsf{ev}} \\
Y
\end{array}
$$

The pair $(X \Rightarrow Y, \mathsf{Ev})$ is the internal hom of X, Y in $\mathcal{L}_!$. Let indeed $f \in \mathcal{L}_!(Z \mathbin{\&} X, Y)$, 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

$$\mathsf{Ev} \circ \langle \mathsf{Cur}(f) \circ \mathsf{pr}_1, \mathsf{pr}_2 \rangle = f \in \mathcal{L}_!(Z \mathbin{\&} X, Y)\,.$$

We have

$$\mathsf{Ev} \circ \langle \mathsf{Cur}(f) \circ \mathsf{pr}_1, \mathsf{pr}_2 \rangle =$$

□

5.6.4.4 The Eilenberg-Moore category

5.6.4.5 The models of free comodules: Girard's construction

5.6.5 Free exponentials and Lafont categories

5.6.5.1 The Melliès-Tabareau-Tasson formula

5.6.6 Interpreting the sequent calculus in a Seely category

Chapter 6

Scott

6.1 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 4.2, fails by lack of naturality of 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 4.2. This connection will be made more explicit in Section 6.2.

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

$$\begin{aligned}\mathsf{Id}_S &= \{(a, a') \in |S| \times |S| \mid a' \leq a\} \\ &= \mathop{\downarrow}_{S \multimap S} \{(a, a) \mid a \in |S|\}\end{aligned}$$

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 6.1.1. *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 $\widehat{\theta^{-1}}$.

6.1.1 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 6.1.2. *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 $\perp = 1$ and dual of S isomorphic to $S^\perp$ (the opposit of S). Given $s \in \mathbf{ScottL}(S, T)$, $s^\perp \in \mathbf{ScottL}(T^\perp, S^\perp)$ is just the usual relational transpose of s, $s^\perp = \{(b,a) \mid (a,b) \in s\}$.

6.1.2 Additive structure

The category **ScottL** is cartesian with terminal object $\top = (\emptyset, \emptyset)$ and cartesian product of S_1 and S_2 the object $S_1 \mathbin{\&} S_2$ such that $|S_1 \mathbin{\&} S_2| = \{1\} \times |S_1| \cup \{2\} \times |S_2|$ 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 $\langle s_1, s_2 \rangle \in \mathbf{ScottL}(T, S_1 \mathbin{\&} S_2)$ 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 $S_1 \oplus S_2 = S_1 \mathbin{\&} S_2$. The associated injections and co-pairing are easily retreived from the projections and the pairing of &.

6.1.3 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 6.2. 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 6.1.3. *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* $s = \{([a_1, \dots, a_n], b) \mid b \in f(\downarrow_S \{a_1, \dots, a_n\})\}$. *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 6.1.3 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 6.1.3, allows to prove easily that d^{s} is natural. The comultiplication of the comonad is $\mathsf{p}^{\mathsf{s}}_S = \{(m, [m_1, \dots, m_n]) \mid \forall i\ m_i \leq_{!_{\mathsf{s}} S} m\} \in \mathbf{ScottL}(!_{\mathsf{s}} S, !_{\mathsf{s}} !_{\mathsf{s}} S)$ 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 6.1.3.

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 $\mathsf{m}^{\mathsf{s},2}_{S_1,S_2} = \{((m_1, m_2), 1 \cdot m'_1 + 2 \cdot m'_2) \mid m'_i \leq_{!_{\mathsf{s}} S} m_i \text{ for } i = 1, 2\} = \widehat{\theta} \in \mathbf{ScottL}(!_{\mathsf{s}} S_1 \otimes !_{\mathsf{s}} S_2, !_{\mathsf{s}}(S_1 \mathbin{\&} S_2))$ where $\theta : |!_{\mathsf{s}} S_1 \otimes !_{\mathsf{s}} S_2| \to |!_{\mathsf{s}}(S_1 \mathbin{\&} S_2)|$ 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 6.1.3 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 $\mathcal{I}(\top) = \{\emptyset\}$), $S_1 \mathbin{\&} S_2$ as cartesian product of S_1 and S_2 (and indeed $\mathcal{I}(S_1 \mathbin{\&} S_2) \simeq \mathcal{I}(S_1) \times \mathcal{I}(S_1)$) 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 6.1.4. *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* $S_1 \mathbin{\&} S_2$. *In particular, if* S *and* T *are finite, so are* $S \multimap T$ *and* $S \Rightarrow T$.

Proof. Since $\mathcal{I}(S_1 \mathbin{\&} S_2) \simeq \mathcal{I}(S_1) \times \mathcal{I}(S_2)$, the finiteness of the S_i's implies that of $S_1 \mathbin{\&} S_2$. Next, we know by Lemma 6.1.1 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 $T = \perp$, 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 6.1.3, 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 $(S \Rightarrow \perp)^{\perp}$ 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.

6.2 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 6.1.3) 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.

6.2.1 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

$$(\mathop{\downarrow}_{S} u) \cap u' \neq \emptyset \Rightarrow u \cap u' \neq \emptyset$$

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)::

$$u \perp[S]\ u' \Leftrightarrow ((\mathop{\downarrow}_{S} u) \cap (\mathop{\downarrow}_{S^{\perp}} u') \neq \emptyset \Rightarrow u \cap u' \neq \emptyset) \Leftrightarrow u' \perp[S^{\perp}]\ u\,.$$

Given $D \subseteq \mathcal{P}(|S|)$, we define

$$D^{\perp[S]} = \{u' \subseteq |S| \mid \forall u \in D \quad u \perp[S]\ u'\}\,.$$

Observe that the following usual properties hold:

- $D \subseteq D' \Rightarrow D'^{\perp[S]} \subseteq D^{\perp[S]}$

[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 6.2.1. *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 $\cap\mathcal{U} \in D$. □

6.2.2 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 $\underline{E}$):

$$u\ \pi_E\ r \quad \text{if} \quad u \in \mathsf{D}(E) \text{ and } \downarrow_{\underline{E}} u = r$$

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

$$u\ \varepsilon_E\ v \quad \text{if} \quad u, v \in \mathsf{D}(E) \text{ and } \downarrow_{\underline{E}} u = \downarrow_{\underline{E}} v\,.$$

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 6.2.2. *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 $\downarrow_{\underline{E \multimap F}} w \cdot \downarrow_{\underline{E}} u \subseteq \downarrow_{\underline{F}} (w \cdot u)$*
4. *for all $u \in \mathsf{D}(E)$ one has $w \cdot u \in \mathsf{D}(F)$ and $\downarrow_{\underline{E \multimap F}} w \cdot \downarrow_{\underline{E}} u = \downarrow_{\underline{F}} (w \cdot u)$*
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 $\downarrow_{\underline{E \multimap F}} w \cdot \downarrow_{\underline{E}} u \supseteq \downarrow_{\underline{F}} (w \cdot u)$ always holds. The implication (2)⇒(3) is due to the fact that $\downarrow_{\underline{E \multimap F}} w \cdot \downarrow_{\underline{E}} u = \downarrow_{\underline{F}} (w \cdot \downarrow_{\underline{E}} u)$. 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 $\downarrow_{\underline{E \multimap F}} w$.

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 $\downarrow_{\underline{E \multimap F}} w \cap (u \otimes v') \neq \emptyset$, 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 $w \cap (u \otimes v') \neq \emptyset$. □

6.2.2.1 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 6.2.3. *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 6.2.4. *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 $\theta \in$ **PoProj**(E, F).*

This is an immediate consequence of Lemma 6.2.2 and of the fact that θ is a strong isomorphism.

Lemma 6.2.5. *Let E, F and G be PP's, then the bijection $\alpha : |\underline{G} \otimes \underline{E} \multimap \underline{F}|$ belongs to* **PoProj**$((G \otimes E \multimap F) \to (G \multimap (E \multimap F)))$ *and* $\alpha^{-1} \in$ **PoProj**$((G \multimap (E \multimap F)) \to (G \otimes E \multimap F))$.

Proof. The idea is to apply (several times) Lemma 6.2.2. 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

$$\begin{aligned} t \cdot (w \otimes \downarrow_{\underline{E}} u) &\subseteq t \cdot (\downarrow_{\underline{G}} w \otimes \downarrow_{\underline{E}} u) \\ &= t \cdot \downarrow_{\underline{G} \otimes \underline{E}} (w \otimes u) \\ &\subseteq \downarrow_{\underline{F}} (t \cdot (w \otimes u)) . \end{aligned}$$

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

$$((\alpha \cdot t) \cdot \downarrow_{\underline{G}} w) \cdot u \subseteq (\downarrow_{\underline{E} \multimap \underline{F}} ((\alpha \cdot t) \cdot w)) \cdot u . \tag{6.1}$$

Indeed, assume (6.1) 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 6.1.1 implies $(a, b) \in s$, proving our contention. So we prove (6.1):

$$\begin{aligned}
((\alpha \cdot t) \cdot \downarrow_{\underline{G}} w) \cdot u &= t \cdot (\downarrow_{\underline{G}} w \otimes u) \\
&= t \cdot \downarrow_{\underline{G} \otimes \underline{E}} (w \otimes u) \\
&\subseteq \downarrow_{\underline{F}} (t \cdot (w \otimes u)) \quad \text{by our assumption about } t \\
&= \downarrow_{\underline{F}} (((\alpha \cdot t) \cdot w) \cdot u) \\
&= \downarrow_{\underline{F}} (\downarrow_{\underline{E} \multimap \underline{F}} ((\alpha \cdot t) \cdot w) \cdot u) \\
&\subseteq \downarrow_{\underline{E} \multimap \underline{F}} ((\alpha \cdot t) \cdot w) \cdot u
\end{aligned}$$

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 6.2.4.

6.2.2.2 Additive structure

Part III

Dynamic models

Chapter 7

Geometry of Interaction

7.1 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 [36]; 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 [32] as a first concrete realisation of Lévy's optimal reduction [34, 35]. The correctness of the sharing reduction w.r.t. optimal reduction was shown using a notion of *consistent paths* that were shown to be an invariant, and which were a reformulation in the sharing graph formalism of execution paths.

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 [31], 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 [41].

7.1.1 Notations and conventions

We relie on the definitions of graphs, paths, proof nets of chapter 2, but briefly recall here some maybe nonstandard conventions.

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 2.4.4.

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: $\delta \circ \gamma : \mathsf{n} \to \mathsf{n}''$.

7.2 Paths in proof nets

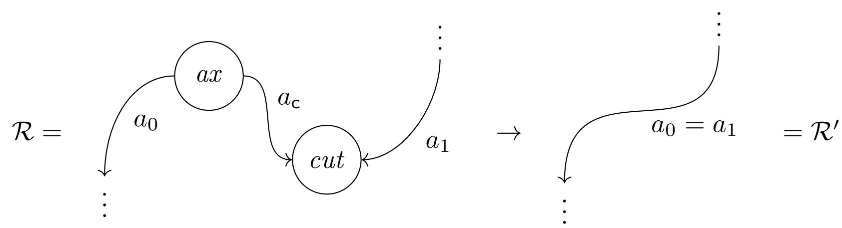
7.2.1 The lifting functor

Let $\mathcal{R}$ be a proof net, c a *cut*-node in $\mathcal{R}$, and $\mathcal{R}'$ be the proof net obtained by applying the one step reduction associated to c, that we view as a rewriting of $\mathcal{R}$. We define 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 $\mathcal{R}$ by describing its action and arrows of $\mathcal{R}'$: for each arrow a in $\mathcal{R}'$ we define by case on the nature of the cut c a path $\mathcal{L}_{\mathsf{c}}(a)$ as a path in $\mathcal{R}$.

This 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 if n is the target source of an edge e and the source node of an edge e' in $\mathcal{R}'$, then $\mathcal{L}_{\mathsf{c}}(e)$ and $\mathcal{L}_{\mathsf{c}}(e')$ are composable paths. This allows to define $\mathcal{L}_{\mathsf{c}}(\mathsf{n})$ as the target of $\mathcal{L}_{\mathsf{c}}(e)$, *i.e.*, the source of $\mathcal{L}_{\mathsf{c}}(e')$, and thus to turn $\mathcal{L}_{\mathsf{c}}$ into a functor from $\mathcal{R}'^*$ to $\mathcal{R}^*$ 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 $\mathcal{R}'^*$.

It remains to define $\mathcal{L}$ on arrows, which we do for each case of cut.

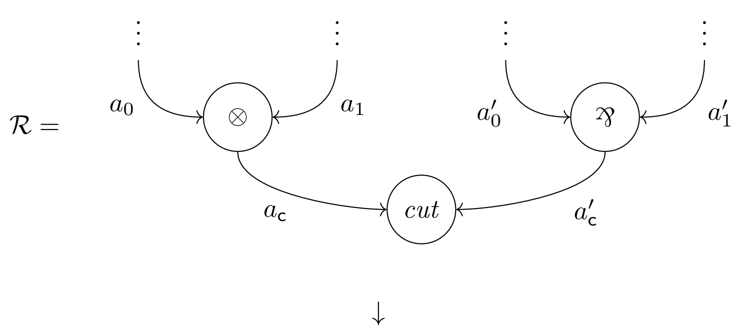
Axiom cut case: let a_0 and a_{c} be the two conclusions of the *ax* node, a_{c} and a_1 be the two premises of the cut c; the retract $\mathcal{R}'$ is thus obtained by identifying a_0 and a_1, removing a_{c}, the *cut* node c and the *ax* node, all other part of the graph being unchanged.



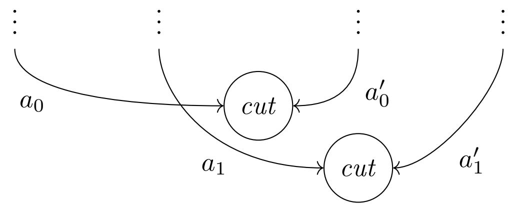
If a is any arrow in $\mathcal{R}'$ we define:

- $\mathcal{L}_{\mathsf{c}}(a) = a^+$ if $a \neq a_0 = a_1$ and,
- $\mathcal{L}_{\mathsf{c}}(a_0 = a_1) = a_1^+ a_{\mathsf{c}}^- a_0^+$ (recall that a_0 and a_1 are distinct arrows in $\mathcal{R}$ but are identified in $\mathcal{R}'$).

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 $⅋$-node, a_{c} and a'_{c} the conclusion respectively of the $\otimes$ and the $⅋$ node and the premises of the cut c. The retract $\mathcal{R}'$ is thus 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 $⅋$ nodes, leaving all other part of the graph unchanged.



$\mathcal{R}' =$



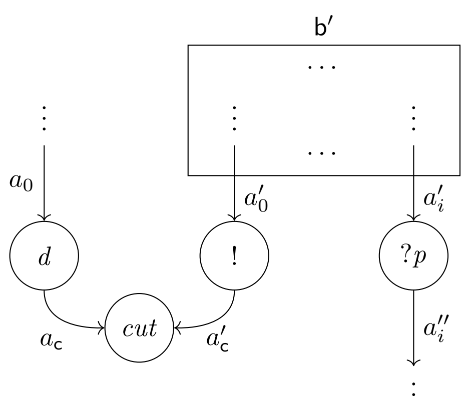
We define:

- $\mathcal{L}_{\mathsf{c}}(a_\epsilon) = a_\epsilon^+ a_{\mathsf{c}}^+$ 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 $\mathcal{R}$.

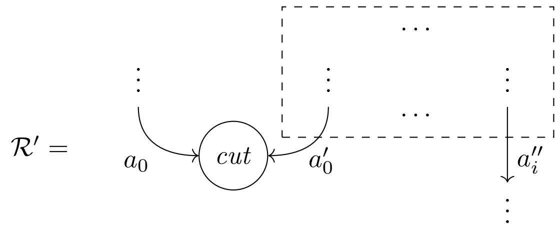
If a is any arrow in $\mathcal{R}'$ we define: $\mathcal{L}_{\mathsf{c}}(a) = a^+$. Note that a_i^+ and a'^-_i are composable in $\mathcal{R}'$ but not in $\mathcal{R}$.

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 thus 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, resourcing a''_i on the source of a'_i and removing a'_i for $i = 1, \dots, k$, and finally removing b' from the box structure of $\mathcal{R}$.

$\mathcal{R} =$



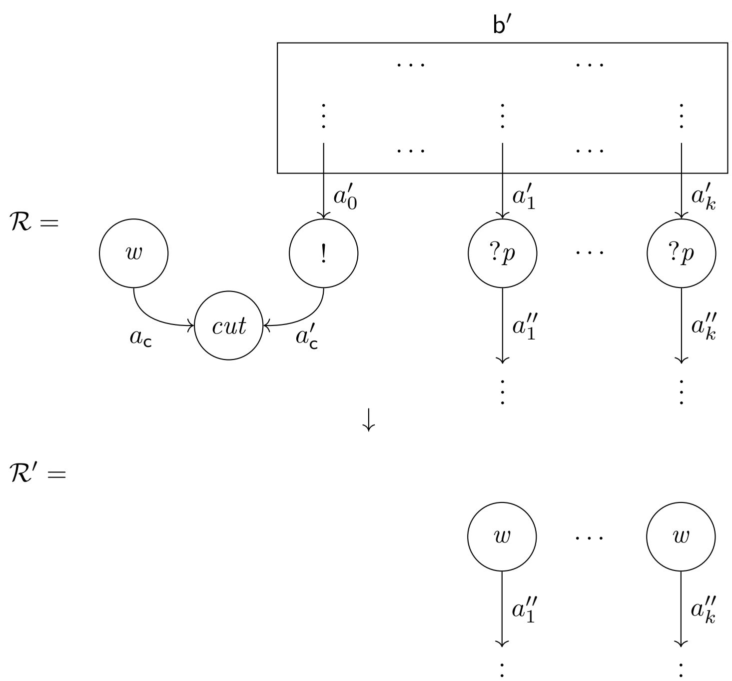
$\downarrow$



We define:

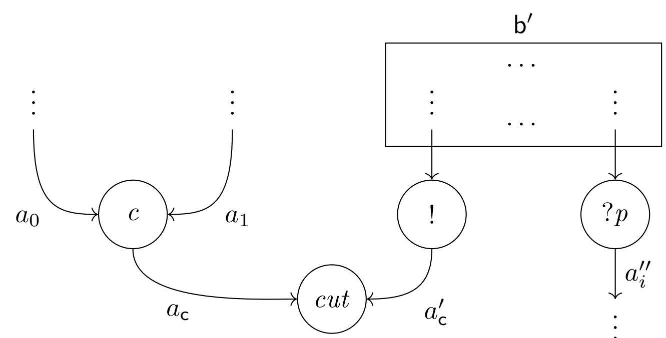
- $\mathcal{L}_{\mathsf{c}}(a_0) = a_0^+ a_{\mathsf{c}}^+$;
- $\mathcal{L}_{\mathsf{c}}(a'_0) = {a'_0}^+ {a'_{\mathsf{c}}}^+$;
- $\mathcal{L}_{\mathsf{c}}(a''_i) = {a'_i}^+ {a''_i}^+$;
- $\mathcal{L}_{\mathsf{c}}(a) = a$ for all other arrows in $\mathcal{R}$.

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 therefore 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.

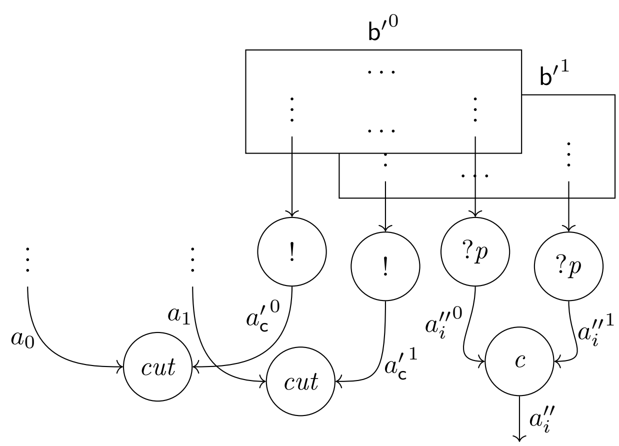


We define $\mathcal{L}_{\mathsf{c}}(a) = a^+$ for all arrows a in $\mathcal{R}'$.

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 $a''_1, \ldots, a''_k$ be the conclusions of the $?p$-nodes $p_1, \ldots p_k$ of b'. The retract $\mathcal{R}'$ is thus 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 $\mathsf{n}_1, \ldots, \mathsf{n}_k$ and $2k$ new arrows ${a''_1}^0, {a''_1}^1, \ldots, {a''_k}^0, {a''_k}^1$, 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.

$\mathcal{R} =$ 

$\downarrow$

$\mathcal{R}' =$ 

We define:

- $\mathcal{L}_{\mathsf{c}}(a_\epsilon) = a_\epsilon^+ a_{\mathsf{c}}^+$ for $\epsilon = 0, 1$;

- $\mathcal{L}_{\mathsf{c}}(a_{\mathsf{c}}'^{\epsilon}) = a_{\mathsf{c}}'^{+}$;

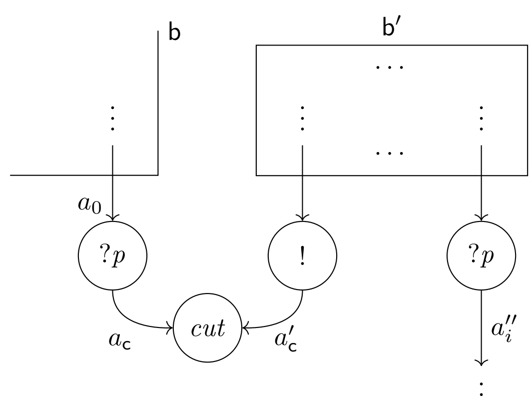
- $\mathcal{L}_{\mathsf{c}}(a'^{\epsilon}) = a'^{+}$ for any arrow a'^{ϵ} in the ϵ-copy b'^{ϵ} of b', copy of an arrow a' in b';

- $\mathcal{L}_{\mathsf{c}}(a_i''^{\epsilon}) = \epsilon_{p_i}$ the empty path at the $?p$-node p_i of b';

- $\mathcal{L}_{\mathsf{c}}(a_i'') = a_i''^{+}$;

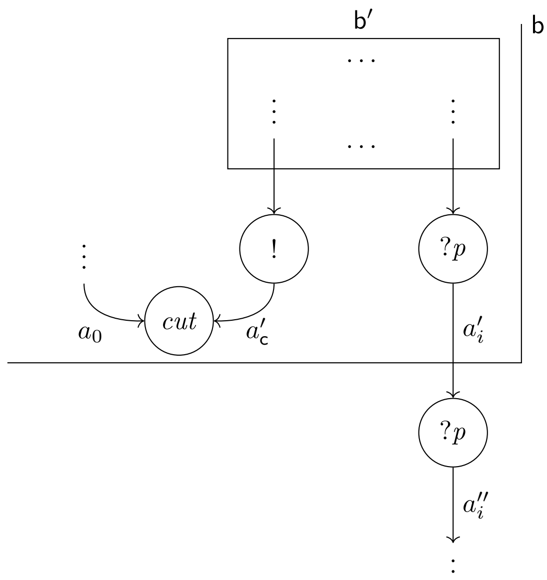
- $\mathcal{L}_{\mathsf{c}}(a) = a^{+}$ for any other arrow in $\mathcal{R}'$.

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', …, p_k' the $?p$-nodes of b' and a_1'', …, a_k'' the conclusions of p_1', …, p_k'. The retract $\mathcal{R}'$ is thus 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, …, p_k as new doors of b and k arrows a_1',… a_k' sourced and targeted respectively on p_i' and p_i and resourcing each a_i'' from p_i.

$\mathcal{R} =$ 

$\downarrow$

$\mathcal{R}' =$



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 $\mathcal{R}'$.

7.2.2 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. 164); 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 slightly abusively denote $\mathcal{L}_{\mathsf{c}}^{-1}(\gamma)$ the set of residuals of γ.

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 $\mathcal{L}_{\mathsf{c}}^{-1}(\Gamma) = \bigcup_{\gamma\in\Gamma} \mathcal{L}_{\mathsf{c}}^{-1}(\gamma)$. Note that if Γ is finite, then so is $\mathcal{L}_{\mathsf{c}}^{-1}(\Gamma)$, 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. 164) 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. 167) 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. 166) 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. 83).

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 $\mathcal{L}_\rho^{-1}(\Gamma) = \mathcal{L}_{\mathsf{c}}^{-1}(\mathcal{L}_{\rho_0}^{-1}(\Gamma))$.

7.2.3 Path reductions

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, a *reduction of* Γ is a sequence ρ of reductions such that at each step at least one element of the residual of Γ crosses the cut being reduced.

Let $\mathcal{R}$ be a proof net, ρ_1 and ρ_2 two sequences of reductions of $\mathcal{R}$ leading to $\mathcal{R}_1$ and $\mathcal{R}_2$. By 2.3.11 there are two sequences of reductions ρ'_2 and ρ'_1 of respectively $\mathcal{R}_1$ and $\mathcal{R}_2$ leading to the same proof-net $\mathcal{R}'$. Furthemore if ρ_1 and ρ_2 are nonempty sequences of *non-weakening* reductions, then so are ρ'_2 and ρ'_1. These results extend to paths reductions:

Theorem 7.2.1 (Confluence of paths reductions)**.** *If Γ is a set of paths in $\mathcal{R}$, Γ_i the residual of Γ in $\mathcal{R}_i$ by the reduction ρ_i for $i = 1, 2$, then Γ_1 and Γ_2 have the same residual Γ' respectively by the reductions ρ'_2 and ρ'_1.*

Note that the theorem is dealing with reductions of proof nets, not with path reductions which are bounded to always fire cuts crossed by some residual of Γ. Indeed path reduction is not confluent, not even locally confluent because 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 and 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 can 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 reduction on saturated sets.

7.2.4 Straight paths

In the sequel we will consider only 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) 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. 164) 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 164, $\gamma_a = a^+_{\mathsf{c}} a^-_1 \overline{\delta_s}\delta_a$ 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 7.2.2. *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.

7.2.5 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 7.2.3 (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 7.2.4. *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, 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 7.2.5 (Why straight paths?). Assuming the notations of the multiplicative reduction p. 164 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. 167. 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.

7.2.6 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 b' (so named for consistency with the notations used in cut elimination steps p. 165);
- γ crosses c;
- the active part of γ contains no (premise or conclusion of an) auxiliary door of b'. Equivalently γ doesn't cross any cut located below an auxiliary door of b', 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 b' 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 b' only by its principal door, except in two possible cases:

1. the first time γ enters b' 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 b' and γ_1 is a path contained in b', sourced on p' and targeted on the principal door of b';
2. the last time γ exits b' may be by an auxiliary door, which is symmetric to the preceding case.

Also note that if γ first enters b' for the first time by crossing the cut c (or symetrically leaves b' 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 7.2.6 (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 $\gamma' \in \mathcal{L}_{\mathsf{c}}^{-1}(\gamma)$ 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 b' can be duplicated but by the special cut assumption all these subpaths are sourced and/or targeted on the principal door of b'. 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) b' 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 b' 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 7.2.7 (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. □

7.3 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.

7.3.1 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 7.3.1 (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.

7.3.2 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 **N** is a one-to-one map σ from a subset of **N**, the *domain* of σ denoted $\operatorname{dom}\sigma$, onto a subset of **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:

- $\operatorname{dom} !(\sigma) = \{n \in \mathbf{N}, n = \langle n_1, n_2\rangle, n_2 \in \operatorname{dom}\sigma\}$;
- for $n = \langle n_1, n_2\rangle \in \operatorname{dom} !(\sigma)$, $!(\sigma)(n) = \langle n_1, \sigma(n_2)\rangle$.

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 $\mathbf{N}$.

[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 [38].

7.3.3 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 $!^{k_1}(x_1)\dots!^{k_n}(x_n)$ (resp. $!^{k_1}(x_1^*)\dots!^{k_n}(x_n^*)$) where the x_i's are 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 $\mathbf{N}$ 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 $\mathbf{N}$ 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.

7.3.4 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 ($⅋$ 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 7.3.2. 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.

7.3.5 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 7.3.3 (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 7.3.4 (Why straight paths (part 2)?)*.* Let $\gamma = a_0^+ a_1^-$ in the multiplicative (p. 164) or contraction (p. 167) 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. 164), 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. 165.

We begin by assuming that γ starts and ends outside the box b' and never crosses an auxiliary door of b'. 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 b', the δ_i's are subpaths entirely contained in b', 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 b' we have $\boldsymbol{w}(\delta_i) = !(u_i)$ where u_i is the weight of δ_i into the subnet contained in b'.

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 $\boldsymbol{w}(\gamma) = v_k x_k^*!(u_k)x_k v_{k-1} \ldots v_1 x_1^*!(u_1)x_1 v_0$ (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 b' 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 b' 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 $\boldsymbol{w}(\gamma) = v_k x_k^*!(u_k)x_k v_{k-1} \ldots v_1 x_1^*!(u_1)x_1 v_0 x_0^*!(u_0)$ 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 $\boldsymbol{w}(\gamma) = \boldsymbol{w}(\gamma')x_0^*$.

[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.

Last γ could enter b' by one auxiliary door. As c is special this can happen only the very first time γ enters b', that is γ starts below an auxiliary door of b', from which we deduce that γ has the form $\gamma = \sigma_0^*\delta_0\beta_0^*\gamma_0\alpha_1\delta_1\beta_1^*\ldots\alpha_k\delta_k\beta_k^*\gamma_k$ where σ_0 is a descent path starting from inside b' and crossing an auxiliary door of b', 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 $\boldsymbol{w}(\gamma) = v_k x_k^*!(u_k)x_k \ldots x_1^*!(u_1)x_1 v_0 x_0^*!(u_0)b_0^*$ where $b_0 = \boldsymbol{w}(\sigma_0)$ is a positive word since σ_0 is a descent path. Therefore $\boldsymbol{w}(\gamma) = v_k!^c(u_k)\ldots!^c(u_1)v_0!^c(u_0)x_0^*b_0^*$.

The residual of γ is $\gamma' = {\sigma_0'}^*\delta_0{\beta_0'}^*\gamma_0\alpha_1'\delta_1{\alpha_1'}^*\ldots\alpha_k'\delta_k{\alpha_k'}^*\gamma_k$ where σ_0' being the residual of σ_0 is still a descent path. Therefore we can compute $\boldsymbol{w}(\gamma') = v_k!^c(u_k)\ldots!^c(u_1)v_0!^c(u_0){b_0'}^*$ 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 $w{b'}^*$ 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 $\boldsymbol{w}(\gamma') = a'w{b'}^*$ 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 7.3.5 (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 b'. 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 7.3.6 (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 7.3.7. 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.

7.4 Proof-nets as operators

The ab^* theorem has another important consequence.

Theorem 7.4.1. *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.

7.4.1 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.*, [40].

7.4.1.1 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 $x \in \ell^2$.

7.4.1.2 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 7.3.2, we therefore get a new model of the equational theory Λ^*, called the *Hilbert space model*.

Remark 7.4.2. 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 goes back in time.

7.4.2 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 $(W_G)_{\mathsf{nn}'}$ 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 $W_{G'} W_G$.

7.4.2.1 The execution formula

We call *interface nodes* in 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 7.3.7, 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 a γ is a Π-*path of length* k or simply a Π^k-*path* if γ is regular of the form $\gamma = \alpha_0\sigma_1\alpha_1 \ldots \sigma_k\alpha_k$ 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.

7.4.2.2 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 7.4.3. *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 7.4.4. *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 $\gamma = \alpha_0 \ldots \sigma_i\alpha_i \ldots \sigma_k\alpha_k$ and $\gamma' = \alpha_0 \ldots \sigma_i\alpha'_i \ldots \sigma'_k\alpha'_k$ 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 7.4.5. 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 7.4.3.

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 7.4.6. *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 7.4.7. 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 $(\pi_{\mathcal{R}}\sigma_{\mathcal{R}})^k$.

7.4.3 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 7.4.8. *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.*

7.4.3.1 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 $A_1, \dots, A_k$ 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.

7.4.3.2 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 7.4.9. *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 7.4.10. 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 7.4.11. The converse, if $\mathcal{R}$ has finitely many regular/persistent paths then $\mathcal{R}$ is strongly normalizable, is also true but trickier to prove.

7.4.4 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 [36].

7.4.4.1 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 7.4.12. *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 $u^l_\sigma(e_i) = e_j$, *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 $k > l$, $u^k_\sigma(e_i) \neq e_j$ so that u_σ is weakly nilpotent.

Conversely suppose $\sigma^k(i) = i$ for some $k > 0$. Then $u^{kl}_\sigma(e_i) = e_i$ for all $l \geq 0$ and u_σ cannot be weakly nilpotent. □

7.4.4.2 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 7.4.13 (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 7.4.14. *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 7.4.15. *Let $\mathcal{R}$ be a proof net (typed or untyped). Then the matrix $\pi\sigma$ viewed as an operator on $(\ell^2)^{\mathsf{I}(\mathcal{R})}$ 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 7.4.12 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 7.4.6, 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. □

7.4.5 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.

7.5 The Interaction Abstract Machine

Let us rephrase the disjoint domains property stated in corollary 7.4.4 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 7.5.1 (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 $\mathbf{N}$ 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 $\mathbf{N}$ model. □

Remark 7.5.2. With a light generalization of the ab^* theorem 7.3.3, 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 $\mathbf{N}$ 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 $\mathbf{N}$ 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 $\mathbf{N}$ model.

7.5.1 The interaction model

Let $\Sigma_{\mathcal{S}}$ be the first order signature containing the following symbols:

- a constant symbol □ (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 □, *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 □.

Remark 7.5.3. 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 $u_1, \dots, u_l$ and a unique π_l such that $\pi = u_1 \cdots u_l \cdot \pi_l$.

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.

7.5.1.1 Embedding the interaction model in the N model.

Recall from section 7.3.2 that the $\mathbf{N}$ model is based on a pairing function $\langle \_, \_ \rangle : \mathbf{N}^2 \xrightarrow{\sim} \mathbf{N}$ that we associate on the left: $\langle n_1, \dots, n_{k+1} \rangle = \langle n_1, \langle \dots, \langle n_k, n_{k+1} \rangle \dots \rangle \rangle$. 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 $\mathbf{N}$ 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 $\mathbf{N}$ 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 $\mathbf{N}$ model.
- Thanks to the condition $\langle 0, 0 \rangle = 0$ if the stacks $u_0 \dots u_N$ and $u_0 \dots u_M$ are equal then $\langle \mathbf{N}(u_0), \dots, \mathbf{N}(u_N), 0 \rangle = \langle \mathbf{N}(u_0), \dots, \mathbf{N}(u_M), 0 \rangle$ so that we may define $\mathbf{N}(u_1 \cdots u_N) = \langle \mathbf{N}(u_1), \dots, \mathbf{N}(u_N), 0 \rangle$.

- 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 7.5.4. *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.

7.5.2 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 7.5.5. *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.

7.5.2.1 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 $(u_0, \dots, u_{d-1})$ and the *d-suffix of* π the substack π_d defined by $\pi = u_0 \cdots u_{d-1} \cdot \pi_d$. If σ is a partial permutation on interaction stacks then $!^d(\sigma)$ acts only on d-suffixes, leaving the d-prefixes unchanged, that is:

$$!^d_{\mathcal{S}}(\sigma)(u_0 \cdots u_{d-1} \cdot \pi) = u_0 \cdots u_{d-1} \cdot \sigma(\pi)$$

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 7.5.6. *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;
- $\boldsymbol{w}_{\mathcal{B}}(a) = \beta$ 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 7.5.7. Contrarily to the **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 7.5.8. *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 Box stack, we choose E for the balanced stack because of the french translation *Équilibré* of *Balanced*.

7.5.2.2 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 $x_0, \dots, x_{l-1}$ and we define $u_\delta = T(u_{\delta_0}, x_l)$.

Lemma 7.5.9. *With the notations just defined for any stacks E and B and any tokens $u_0, \dots, u_l$:*

$$\boldsymbol{w}_{\mathcal{B}}(\delta)(u_0 \cdot E, u_1 \cdots u_l \cdot B) = (u_\delta[u_0, u_1, \dots, u_l] \cdot E, B)$$

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. □

7.5.2.3 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 7.5.10. 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*.

7.5.3 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 $\boldsymbol{w}_\gamma = \boldsymbol{w}_B(\gamma)$.

7.5.3.1 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 7.5.11. *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 $⅋$ 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 7.5.12. 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 7.5.13 (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 7.5.14. 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 7.5.9 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 $\sigma_\gamma = \sigma_2 \circ \sigma_1$. □

7.5.3.2 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 7.5.15. *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):

$$\begin{array}{lllll}
E, & B & \xrightarrow{x^*_{\mathcal{B}}} & u \cdot E_1, \quad B & \text{where } u \cdot E_1 = x^*_{\mathcal{S}}(E) \\
 & & \xrightarrow{\beta^*} & E_1, \quad u \cdot B & \\
 & & \xrightarrow{\boldsymbol{w}_{\gamma''}} & E_2, \quad u \cdot B & \text{by induction hypothesis on } \gamma'', \text{ for some } E_2 \\
 & & \xrightarrow{\beta} & u \cdot E_2, \quad B & \\
 & & \xrightarrow{y_{\mathcal{B}}} & v \cdot E_3 \quad B & \text{where } v \cdot E_3 = y_{\mathcal{S}}(u \cdot E_2) \\
 & & \xrightarrow{\boldsymbol{w}_{\gamma'''}} & E' \quad B & \text{by induction hypothesis on } \gamma''', \text{ for some } E'.
\end{array}$$

□

Let b be a box in a proof net. A *b-path* is a path starting upwardly and ending downwardly 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 7.1 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: $\delta_0\theta_1\delta_1 \ldots \theta_k\delta_k$ 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 downardly on n;
- for $i > 0$, δ_i starts upwardly on a $?p$-node p_i of b; for $i < k$, δ_i ends downardly on p_{i+1} (the p_is are not necessarily distinct);
- each θ_i is a ?-cycle starting downardly and ending upwardly at p_i.

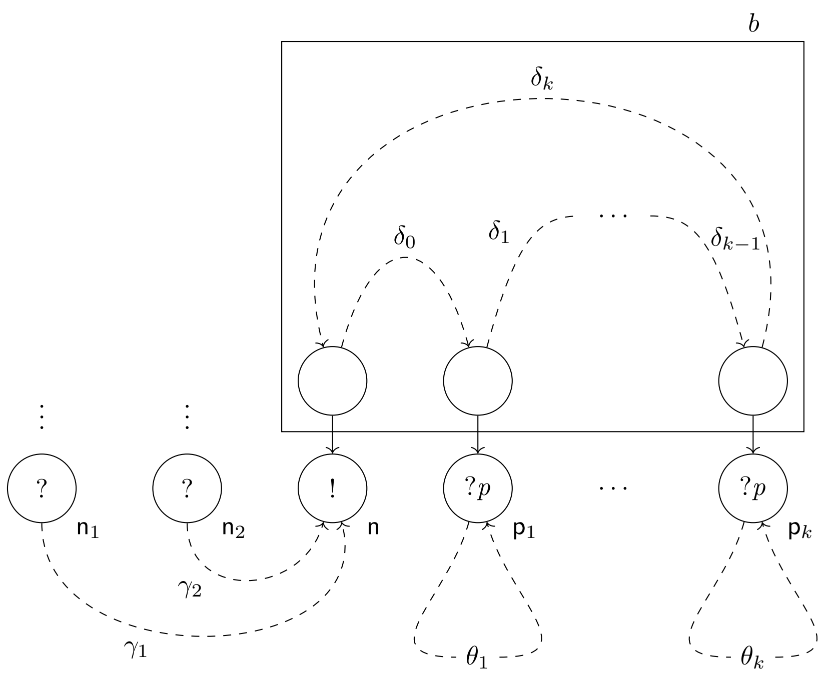


Figure 7.1: The general form of a ?-cycle: γ_i are w.b.p., θ_i are ?-cycles, the p_is are not necessarily distinct

Theorem 7.5.16 (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 7.5.15. 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 $(u \cdot E, B)$:

$$\begin{array}{llllll} u \cdot E, & B & \xrightarrow{\beta^*} & E, & u \cdot B & \\ & & \xrightarrow{\boldsymbol{w}_{\delta_0}} & E', & u \cdot B & \text{where } E' \text{ is given by the lemma} \\ & & \xrightarrow{\beta^*} & u \cdot E', & B & \end{array}$$

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:

$$\begin{array}{llll} u \cdot E, & B \xrightarrow{\boldsymbol{w}_{\delta_1}} & u \cdot E, & \sigma_{\gamma_1}(B)) \\ & \xrightarrow{\boldsymbol{w}_{\delta_0}} & u \cdot \tau_{\delta_0}(E), & \sigma_{\gamma_1}(B)) \end{array}$$

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 $\delta = \delta_0 \theta_1 \delta_1 \ldots \theta_k \delta_k$, 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 7.5.15 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 $(u \cdot E, B) \in \mathrm{dom}\, \boldsymbol{w}_{\theta'_i}$ 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 7.5.9 there is a $\Sigma_{\mathcal{S}}$-term $u_{\alpha_i}[x_0, x_1, \ldots, x_{l_i}]$ such that $x_i(u_0 \cdot E, u_1 \cdots u_{l_i} \cdot B_i) = (u_{\alpha_i}[u_0, \ldots, u_{l_i}] \cdot E, B_i)$.

Putting all this together we have

$$\boldsymbol{w}_\delta = \beta \boldsymbol{w}_{\delta'_k} \beta^* t^*_{\mathcal{B}}\, x^*_k w_{\theta'_k} x_k \ \ldots\ t_{\mathcal{B}} \beta \boldsymbol{w}_{\delta'_1} \beta^* t^*_{\mathcal{B}}\, x^*_1 w_{\theta'_1} x_1\, t_{\mathcal{B}} \beta \boldsymbol{w}_{\delta'_0} \beta^*$$

and we may compute the action of $\boldsymbol{w}_\delta$ on the bistack $(u \cdot E, B)$:

$$\begin{array}{llrll}
u \cdot E, \quad B & \xrightarrow{\beta^*} & E, & u \cdot B & \\
 & \xrightarrow{\boldsymbol{w}_{\delta'_0}} & u_0 \cdot E'_0, & u \cdot B & u_0 \text{ and } E'_0 \text{ defined by } u_0 \cdot E'_0 = \rho_0(E) \\
 & \xrightarrow{\beta} & u \cdot u_0 \cdot E'_0, & B & \\
 & \xrightarrow{t_{\mathcal{B}}} & T(u_0, u) \cdot E'_0, & B & \\
 & \xrightarrow{x_1} & u_{\alpha_1} \cdot E'_0, & B_1 & \text{where } B_1 \text{ is defined by } B = u_1 \cdots u_{l_1} \cdot B_1 \\
 & & & & \text{and } u_{\alpha_1} = u_{\alpha_1}[T(u_0, u), u_1, \ldots, u_{l_i}] \\
 & \xrightarrow{\boldsymbol{w}_{\theta'_1}} & u_{\alpha_1} \cdot E_1, & B_1 & \text{where } E_1 = \tau_{\theta'_i}(E'_0) \\
 & \xrightarrow{x^*_1} & T(u_0, u) \cdot E_1, & B & \\
 & \xrightarrow{t^*_{\mathcal{B}}} & u \cdot u_0 \cdot E_1, & B & \\
 & \xrightarrow{\beta^*} & u_0 \cdot E_1, & u \cdot B & \\
 & \xrightarrow{\boldsymbol{w}_{\delta'_1}} & u_1 \cdot E'_1, & u \cdot B & \text{where } u_1 \cdot E'_1 = \rho_1(u_0 \cdot E_1) \\
 & \xrightarrow{\beta} & u \cdot u_1 \cdot E'_1, & B & \\
 & \xrightarrow{t_{\mathcal{B}}} & T(u_1, u) \cdot E'_1, & B & \\
 & & \vdots & & \\
 & \longrightarrow & T(u_{k-1}, u) \cdot E'_{k-1}, & B & \\
 & \xrightarrow{x_k} & u_{\alpha_k} \cdot E'_{k-1}, & B_k & \\
 & \xrightarrow{\boldsymbol{w}_{\theta'_k}} & u_{\alpha_k} \cdot E_k, & B_k & \\
 & \xrightarrow{x^*_k} & u \cdot u_{k-1} \cdot E_k, & B & \\
 & \xrightarrow{\beta^*} & u_{k-1} \cdot E_k, & u \cdot B &
\end{array}$$

$$\begin{aligned} &\xrightarrow{\boldsymbol{w}_{\delta'_k}} && E', \quad u \cdot B \quad \text{where } E' = \rho_k(u_{k-1} \cdot E_k) \\ &\xrightarrow{\beta} && u \cdot E', \quad B \end{aligned}$$

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 7.5.17. 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.

7.5.3.3 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 7.5.18. *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 7.5.9 there is a $\Sigma_\mathcal{S}$ term u_{δ_i} such that $\boldsymbol{w}_{\delta_i}(u \cdot E, u_1 \cdots u_{l_i} \cdot B) = (u_{\delta_i}[u, u_1, \dots, u_{l_i}] \cdot E, B)$ 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 7.5.16 we may compute the action of $\delta_1\theta'$ as:

$$\begin{aligned} (u \cdot E, \quad u_1 \cdots u_{l_i} \cdot B) \quad &\xrightarrow{\boldsymbol{w}_{\delta_1}} \quad (u_{\delta_1}[u, u_1, \dots, u_{l_i}] \cdot E,\, B) \\ &\xrightarrow{\boldsymbol{w}_{\theta'}} \quad (u_{\delta_1}[u, u_1, \dots, u_{l_i}] \cdot E',\, B) \end{aligned}$$

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. 167, 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 7.1 we have $\theta = \alpha \gamma_1 \delta_0 \theta_1 \delta_1 \ldots \theta_k \delta_k \overline{\gamma}_1 \overline{\alpha}$ where α is an exponential branch from a dereliction node d to the root node n_1 of the exponential tree and $\theta' = \alpha' \gamma_1' \delta_0' \theta_1' \delta_1' \ldots \theta_k' \delta_k' \overline{\gamma}_1'' \overline{\alpha}''$ 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 downwardly 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. $\square$

Appendix A

Graphs

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 *node*, $\mathcal{A}(\mathcal{G}) \coloneqq \mathcal{A}$ is the set of *arrow*, 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 *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 $\mathsf{t}(a^-) \coloneqq \mathsf{s}(a)$.

Two edges e_0 and e_1 are *composable edges* 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 $\gamma = (n_0, \vec{e})$ where n_0 is a node (the *source* of the path, also noted $\mathsf{s}(\gamma)$) and $\vec{e}$ is a (possibly infinite) sequence of edges *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 $\mathsf{s}(e_1) = n_0$;
- and, for any $1 \leq i < N$, $\mathsf{t}(e_i) = \mathsf{s}(e_{i+1})$.

Note that we really should call these *undirected path* 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.

Let $\gamma = (n_0, (e_i)_{1 \leq i \leq N})$ be a path. The *target* $\mathsf{t}(\gamma)$ of γ is n_0 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 between two nodes.* A node n is *internal node* to γ if $n = \mathsf{s}(e_i)$ with $i > 1$ (or equivalently $n = \mathsf{t}(e_i)$ with $i < N$). 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_0}, in which case $\mathsf{s}(\gamma) = \mathsf{t}(\gamma) = n_0$ and γ has no internal node. We will thus often identify a path with its sequence of edges.

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 paths* and write $\gamma_0\gamma_1$ for their *composition of paths* or *composition of paths*:

$$\gamma_0\gamma_1 = (n_0, (e_i)_{1\leq i\leq N_0+N_1})$$

so that $\mathsf{s}(\gamma_0\gamma_1) = \mathsf{s}(\gamma_0) = n_0$ and $\mathsf{t}(\gamma_0\gamma_1) = \mathsf{t}(\gamma_1)$.

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_0} is clearly neutral when composed on the left with any path of source n_0, on the right with any path of target n_0. 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 edge* $\overline{e}$ of an edge e by $\overline{e} \coloneqq a^-$ if $e = a^+$, and $\overline{e} \coloneqq a^+$ if $e = a^-$. Then, for any finite path γ in $\mathcal{G}$ with edges $(e_i)_{0\leq i<N}$, we define the reverse path $\overline{\gamma} \coloneqq (\mathsf{t}(\gamma), (\overline{e}_{N-1-i})_{0\leq i<N})$ so that $\mathsf{s}(\overline{\gamma}) = \mathsf{t}(\gamma)$ and $\mathsf{t}(\overline{\gamma}) = \mathsf{s}(\gamma)$. The reverse operation is compatible with composition in the sense that:

$$\overline{\epsilon_{n_0}} = \epsilon_{n_0} \text{ and,}$$
$$\overline{\gamma_0\gamma_1} = \overline{\gamma_1}\,\overline{\gamma_0}$$

for any finite paths γ_0 and γ_1. The category $\mathcal{G}^*$ is thus an *involutive category.*

We say a path γ *arrow crossed by a path* an arrow a if either a^+ or a^- occurs as an edge of γ. And we say γ *edge crossed by a path* an edge e if e or $\overline{e}$ is a subpath of γ. A path γ is *simple path* if it does not cross the same arrow (or, equivalently, the same edge) twice. We say two paths γ_1 and γ_2 are *disjoint paths* 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 disjoint.

We write $n \simeq_{\mathcal{G}} n'$ if there exists a path (equivalently, a simple path) from n to n'. We obtain that $\simeq_{\mathcal{G}}$ is an equivalence relation, and say n and n' are *connected nodes* 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 graph* if it is non empty any two nodes are connected by a path: in other words, it has exactly one connected component.

A path γ is said to be *closed path* if $\mathsf{s}(\gamma) = \mathsf{t}(\gamma)$, and *open path* otherwise. A *cyclic path*, also called a *cycle*, is a non empty simple path γ that is closed. A graph is said to be *acyclic graph* if it has no cycle.

Lemma A.0.1 (Acyclic Connected Components)**.** *In a finite acyclic graph, the number of connected components is the number of nodes minus the number of arrows.*

[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 $\mathcal{G}$, whose arrows are the edges of $\mathcal{G}$.

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.0.2 (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.0.1. 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 path* if it contains no reversed arrow. 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 [42].

B.0.1 Definitions and Notations

An *abstract reduction system (ARS)* (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:

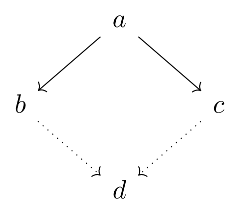
- $a \to b$ if $(a, b) \in \to$. b is called a 1-*step reduct* of a.
- $a \leftarrow b$ if $b \to a$.
- $a \to^{=} b$ if $a = b$ or $a \to b$ ($\to^{=}$ is the reflexive closure of $\to$).
- $a \to^{+} b$ if there exists a finite sequence $(a_k)_{0 \leq k \leq N}$ ($N \geq 1$) of elements of A such that $a = a_0$, $a_N = b$ 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 $\mathcal{A} \restriction_a \mathcal{A} \restriction_a = (A \restriction_a, \to \cap (A \restriction_a \times A \restriction_a))$ 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 reduction sequence* and its length is $N-1$. We use the notation

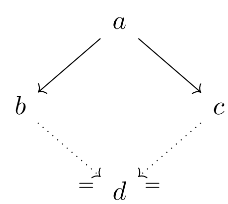
$a \to^k b$ if there exists a finite reduction sequence of length k starting from a and ending on b.

B.0.2 Confluence

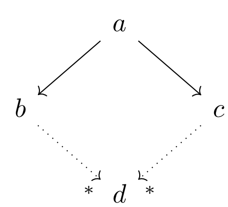
An ARS $(A, \to)$ has the *diamond property* if for any a, b and c in A with $a \to b$ and $a \to c$, there exists some d in A such that both $b \to d$ and $c \to d$. Thus diagrammatically:



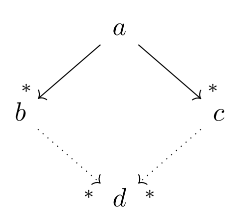
An ARS $(A, \to)$ is *sub-confluent* if for any a, b and c in A with $a \to b$ and $a \to c$, there exists some d in A such that both $b \to^= d$ and $c \to^= d$. Thus diagrammatically:



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:



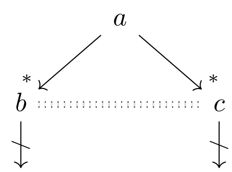
An ARS $(A, \to)$ is *confluent* if for any a, b and c in A with $a \to^* b$ and $a \to^* c$, there exists some d in A such that both $b \to^* d$ and $c \to^* d$. Thus diagrammatically:



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* 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:

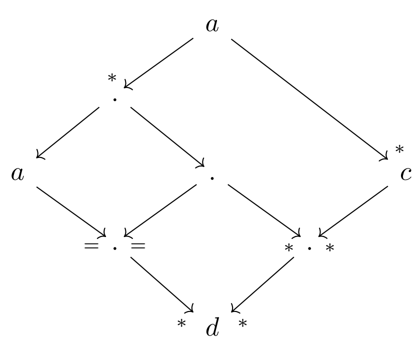


Proposition B.0.1 (Confluence Properties)**.** *For any ARS,*

- *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.0.3 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.0.2 (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.0.3 (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.0.4 (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 *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 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 *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.0.5 (Normalization Properties)**.** *For any ARS, μ-decreasing for some $\mu \Longrightarrow$ strongly normalizing $\Longrightarrow$ weakly μ-decreasing for some $\mu \Longrightarrow$ 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 $\mu(A \restriction_a)$.

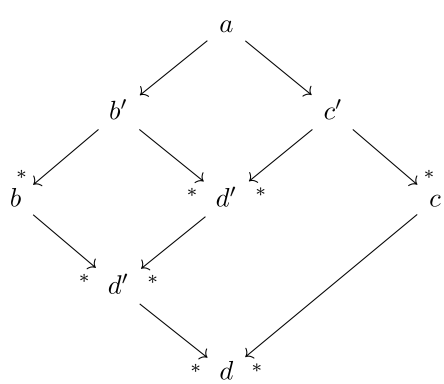
□

Proposition B.0.6 (Weak Normalization and Confluence)**.** *For any ARS, weakly normalizing $\wedge$ unique normal form $\Longrightarrow$ 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.0.7 (Newman's Lemma)**.** *For any* ARS*, strongly normalizing* $\wedge$ *locally confluent* $\Longrightarrow$ *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.0.2). 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.



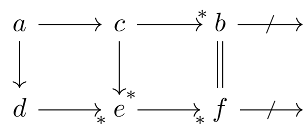
□

Proposition B.0.8 (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.0.7 and B.0.1, $\mathcal{A}\restriction_c$ 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.0.4 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.0.9 (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.0.4, $(B, \to_B^+)$ is strongly normalizing, thus $\leftarrow^+$ is a well founded relation. We can conclude with Proposition B.0.5 since $\mathcal{A}$ is then φ-decreasing. □

Proposition B.0.10 (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 $\mathcal{A}$.*

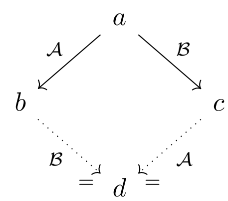
Proof. Assume b and c are normal forms with $a \to_A^* b$ and $a \to_A^* c$, then $\varphi(a) \to_B^* \varphi(b)$ and $\varphi(a) \to_B^* \varphi(c)$ with $\varphi(b)$ and $\varphi(c)$ normal forms. This entails $\varphi(b) = \varphi(c)$ by unique normal form for $\mathcal{B}$, and finally $b = c$ since φ is injective on normal forms. □

B.0.5 Commutation

In this section, we consider two ARSs $\mathcal{A} = (A, \rightarrow_{\mathcal{A}})$ and $\mathcal{B} = (A, \rightarrow_{\mathcal{B}})$ on the same set A.

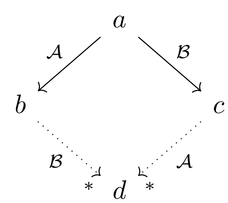
The ARS $\mathcal{A} \bowtie \mathcal{B}$ is defined as $(A, \rightarrow_{\mathcal{A}\bowtie\mathcal{B}})$ with $\rightarrow_{\mathcal{A}\bowtie\mathcal{B}} = \rightarrow_{\mathcal{A}} \cup \rightarrow_{\mathcal{B}}$. Note that $\rightarrow^*_{\mathcal{A}\bowtie\mathcal{B}} = (\rightarrow^*_{\mathcal{A}} \cup \rightarrow^*_{\mathcal{B}})^*$

We say that $\mathcal{A}$ and $\mathcal{B}$ *sub-commute* if for any a, b and c in A such that $a \rightarrow_{\mathcal{A}} b$ and $a \rightarrow_{\mathcal{B}} c$, there exists d such that $b \rightarrow^{=}_{\mathcal{B}} d$ and $c \rightarrow^{=}_{\mathcal{A}} d$. Thus diagrammatically:



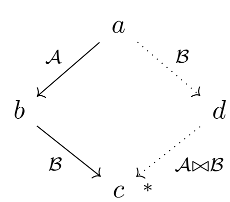
With this definition, an ARS is sub-confluent if it sub-commutes with itself.

We say that $\mathcal{A}$ and $\mathcal{B}$ *locally commute* if for any a, b and c in A such that $a \rightarrow_{\mathcal{A}} b$ and $a \rightarrow_{\mathcal{B}} c$, there exists d such that $b \rightarrow^{*}_{\mathcal{B}} d$ and $c \rightarrow^{*}_{\mathcal{A}} d$. Thus diagrammatically:



With this definition, an ARS is locally confluent if it locally commutes with itself.

We say that $\mathcal{A}$ *quasi-commute* $\mathcal{B}$, if for any a, b and c in A such that $a \rightarrow_{\mathcal{A}} b$ and $b \rightarrow_{\mathcal{B}} c$, there exists d such that $a \rightarrow_{\mathcal{B}} d$ and $d \rightarrow^{*}_{\mathcal{A}\bowtie\mathcal{B}} c$. Thus diagrammatically:



Proposition B.0.11 (Commutation of Strong Normalization)**.** *If $\mathcal{A} = (A, \rightarrow_{\mathcal{A}})$ and $\mathcal{B} = (A, \rightarrow_{\mathcal{B}})$ 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 $B = \{a \in A \mid \exists b \in B_0, a \rightarrow^*_{\mathcal{A}\bowtie\mathcal{B}} b\}$ which is non-empty as well ($B_0 \subseteq B$) and is such that $a \rightarrow^*_{\mathcal{A}\bowtie\mathcal{B}} b$ 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 $\to_{\mathcal{A}}$-reduct in B'. If a has no 1-step $\to_{\mathcal{A}}$-reduct in B, we have an element with no 1-step $\to_{\mathcal{A}\bowtie\mathcal{B}}$-reduct in B. Otherwise $a \to_{\mathcal{A}} b$ for some b which is in B and not in B' thus there exists $c \in B$ such that $b \to_{\mathcal{B}} c$. By quasi-commutation, we have d such that $a \to_{\mathcal{B}} d \to^*_{\mathcal{A}\bowtie\mathcal{B}} c$. We have $d \in B$ since $d \to^*_{\mathcal{A}\bowtie\mathcal{B}} c$ but this contradicts the fact that $a \in B'$.

This means a cannot have a 1-step $\to_{\mathcal{A}\bowtie\mathcal{B}}$-reduct in B, so that a belongs to B_0 and has no 1-step $\to_{\mathcal{A}\bowtie\mathcal{B}}$-reduct in B_0. $\square$

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] 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 .

[30] 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.

[31] André Joyal, Ross Street, and Dominic Verity. "Traced Monoidal Categories". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 119 (1996), pp. 447–468.

[32] 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.

[33] Olivier Laurent. "Sequentialization of Multiplicative Proof Nets". Unpublished note. Available at http://perso.ens-lyon.fr/olivier.laurent/seqmll.pdf . Apr. 2013.

[34] Jean-Jacques Lévy. "Lambda-Calcul Étiqueté". Thèse de Doctorat. Université Paris 7, 1978.

[35] 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.

[36] 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.

[37] 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.

[38] Mario Petrich. *Inverse Semigroups*. Pure and Applied Mathematics. John Wiley & sons, 1984.

[39] Laurent Regnier. "Une équivalence sur les lambda-termes". In: *Theoretical Computer Science* 126 (1994), pp. 281–292.

[40] Walter Rudin. *Real and complex analysis*. McGraw-Hill Education, 1987.

[41] Thomas Seiller. "Logique dans le facteur hyperfini, géométrie de l'interaction et complexité". Thèse de Doctorat. Université d'Aix-Marseille, Nov. 2012.

[42] Terese. *Term Rewriting Systems*. Vol. 55. Cambridge tracts in theoretical computer science. Cambridge University Press, 2003.

Index

1-step reduct, **213**
2-category, 129
$\mathbf{N}$ model of Λ^*, **176**
$\mathcal{A} \bowtie \mathcal{B}$, **220**
$\mathcal{B}$ model, **196**
$\mathcal{B}$-weigth, **196**
$\mathcal{S}$ model, **194**
μ-decreasing, **217**
μ-increasing, **217**
!-cycles, **201**
$\rightarrow$-minimal, 216
$\rightarrow$-minimal, **215**
$\mathsf{s}(\gamma)$, **209**
$\mathsf{t}(\gamma)$, **209**
$\mathtt{ps}(\pi)$, **49**
ε, **209**
?-cycles, **201**
$a \leftarrow b$, **213**
$a \rightarrow^* b$, **213**
$a \rightarrow^+ b$, **213**
$a \rightarrow^= b$, **213**
$a \rightarrow b$, **213**
$a \simeq b$, **213**
ab^*-form, **178**

abstract reduction system, **213**
Active and passive parts of a path, **171**
acyclic, **53**, **210**
acyclic graph, 210, 211
Algebra of a monad, 134
Almost connected, **75**
arrows, **209**

Balanced stack, **196**
Bistack, **196**
blocking ⅋, **62**
Box stack, **196**
Breaking a path by a cut, **170**

Cartesian closed categories, 131
Cartesian product, 131
Category of paths of $\mathcal{G}$, **210**
clasping chain, **75**
Clasping rank, **75**
clasps, **64**
closed, **210**
composable, **209, 210**
composition, **210**
concatenation, **210**
conclusion, **48**
conclusion sequent, **48**
conclusions, **47**
confluent, 215, 217, 218
confluent, **214**
connected, **210**
connected component, 210
connected component, **210**
connected graph, 211
Connected proof net with jumps, **69**
connected sequential, **60**
Consistent paths, **162**
convergent, 59, 85, 87, 89
convergent, **216**
crosses, **210**
Curryfication, 132
cycle, **210**
cyclic path, **210**

decreasing, 217
diagram, 130
diamond property, 215
diamond property, **214**
directed, **211**
directed acyclic graph, 47

directed acyclic graph, **211**
Discrete category, 131
disjoint, **210**
DM-normal, **19**
Dual of a DM-normal formula, **20**
Dynamic algebra, **175**

edge, **209**
Eilenberg-Moore category, 134
Equalizer, 131
Evaluation map, 132
Exchange of premises, **170**
Execution formula, **161**
Execution path, **161**
Exponential token, **193**

Final ?-cycle, **202**
Finally separing, **192**
finite, **213**
Fragment, **23**
Free algebras of a monad, 134
Functors, 128

Geometry of interaction, **161**
GoI situation, **162**
graph, **209**

Hilbert space model, **185**

incoming arrow, 47
incoming arrow, **209**
increasing, 218
Initial ?-cycle, **202**
Initial jump function, **70**
Initial node, **70**
Initially separing, **192**
Interaction Abstract Machine, **161**
Interaction model, **194**
Interaction stack, **193**
Interaction token, **193**
Interface node, **186**
internal, **209**
Internal hom, 131
Internal hom object, 132
internal nodes, **48**
involutive category, **210**

Jump function, **69**

Kleisli category, 134

Legal path, **162**, **205**
length, **209**
Lifting functor, **163**
Limiting projective cone, 130
Linear function, 102
locally commute, 220
locally commute, **220**
locally confluent, 215, 218, 220
locally confluent, **214**

Monomial, **184**

Natural transformations, 129
Negated atom, **19**
nodes, **209**
normal form, **215**
Normal path, **172**

open, **210**
outgoing arrow, 47
outgoing arrow, **209**

path, **209**
path from n to n', **209**
Persistent paths, **161**
Positive atom, **19**
prefix, **210**
Premise node, **48**
premises, **47**
Projection, 131
Projective cone, 130
Projective limits, 130
proof net, **53**
Proof net with jumps, **69**
Proof structure with jumps, **69**
proper cycle, **63**

quasi-commute, 220
quasi-commutes over, **220**

Reduction of Γ, **170**
reduction sequence, **213**
Regular path, **179**

Regular path, **161**
Residuals of a path, **169**
restriction, **213**
reverse, **210**

sequential, **60**
Sharing graphs, **162**
Sharing reduction, **162**
simple, **210**
simulation, 219
simulation, **219**
source, **209**
Special cut for a path, **173**
Special reduction, **175**
Stable functions, 98
Straight cycle, **191**
Straight paths, **171**
Strict reduction, **83**
strict simulation, 219
strict simulation, **219**
strongly normalizing, 216–220
strongly normalizing, **216**
sub-commute, 220
sub-commute, **220**
sub-confluent, 215, 220
sub-confluent, **214**
subpath, **210**
suffix, **210**

target, **209**
terminal, **48**
Terminal objects, 130
Trace of stable function, 100
typed proof structure, **48**
typing, **48**

undirected paths, **209**
unique normal form, 215, 217, 219
unique normal form property, **215**
untyped multiplicative proof structure, **47**

Virtual cuts, **199**

W.b.p., **199**
weakly μ-decreasing, **217**
weakly decreasing, 217
weakly normalizing, 217, 218
weakly normalizing, **216**
Weight, **178**
Well balanced paths, **199**
well founded, 216
well founded, **216**