

LLWiki book

December 12, 2024

Informations

This document has not been built as a linear L^AT_EX document but from an extraction of LLWiki. This explains why the structure of the document may look strange at various points.

LLWiki is a collaborative project aiming at the writing of a compendium about linear logic [Gir87]. A previous project with similar goals is [Linear Logic Pages](#).

The LLWiki is a collective work created under the authority of the editorial board:

- Emmanuel Beffara (IML)
- Thomas Ehrhard (PPS)
- Jean-Yves Girard (IML)
- Olivier Laurent (LIP)
- Damiano Mazza (LIPN)
- Laurent Regnier (IML)
- Lorenzo Tortora de Falco (Roma 3)

The editorial board is granted a monopoly of rights on the content of this collective work.¹

The content of the LLWiki is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 2.0 France](#). This means you can share and adapt this content, under the following conditions:

- you must cite the “*editorial board of the LLWiki*” as the licensor;
- you may not use this content for commercial purposes;
- if you distribute a possibly modified version of this content, you must do so under the same or similar license.

¹The contributors are the individuals owning an account on the LLWiki, which allows them to edit the content. By applying for the creation of their account, the candidates to the status of contributor agree to grant the monopoly of rights on the LLWiki as a collective work to the editorial board; they moreover agree that their contributions are distributed under the terms of the [Creative Commons Attribution-Noncommercial-Share Alike 2.0 France](#), the editorial board being cited as the licensor.

The contributors however keep their rights on their respective contributions and are free to exploit them for their own sake.

Contents

I	Syntax	8
1	Sequent calculus	9
1.1	Formulas	9
1.2	Sequents and proofs	10
1.3	Equivalences	11
1.3.1	De Morgan laws	11
1.3.2	Fundamental equivalences	12
1.3.3	Definability	12
1.4	Properties of proofs	13
1.4.1	Cut elimination and consequences	13
1.4.2	Expansion of identities	14
1.4.3	Reversibility	14
1.5	One-sided sequent calculus	15
1.6	Variations	16
1.6.1	Exponential rules	16
1.6.2	Non-symmetric sequents	16
1.6.3	Mix rules	17
1.7	Equiprovability	17
1.7.1	Definition	17
1.8	Isomorphism	17
1.8.1	Definition	17
1.8.2	List of isomorphisms	17
1.8.2.1	Linear negation	17
1.8.2.2	Neutrals	17
1.8.2.3	Commutativity	18
1.8.2.4	Associativity	18
1.8.2.5	Multiplicative-additive distributivity	18
1.8.2.6	Linear implication	18
1.8.2.7	The exponential isomorphisms	18
1.8.2.8	Quantifiers	18
1.9	List of equivalences	18
1.9.1	Multiplicatives	18
1.9.2	Additives	19
1.9.3	Quantifiers	19
1.9.4	Exponentials	19
1.9.5	Polarities	19
1.9.6	Second order encodings	19
1.9.7	Miscellaneous	19
1.10	Lattice of exponential modalities	20

1.11	Provable formulas	21
1.11.1	Distributivities	21
1.11.1.1	Standard distributivities	21
1.11.1.2	Linear distributivities	21
1.11.2	Factorizations	21
1.11.3	Identities	21
1.11.4	Additive structure	21
1.11.5	Quantifiers	22
1.11.6	Exponential structure	22
1.11.7	Monoidality of exponentials	22
1.11.8	Promotion principles	22
1.11.9	Commutations	22
1.12	Non provable formulas	22
1.13	Mix	23
1.13.1	Binary Mix rule	23
1.13.2	Nullary Mix rule	23
1.14	Additive cut rule	23
1.15	Reversibility and focusing	24
1.15.1	Reversibility	24
1.15.2	Generalized connectives and rules	25
1.15.3	Focusing	26
1.16	Positive formula	28
1.16.1	Positive connectives	28
1.16.2	Generalized structural rules	29
1.17	Negative formula	29
1.17.1	Negative connectives	29
1.17.2	Generalized structural rules	30
1.18	Regular formula	30
1.18.1	Alternative characterization	30
1.18.2	Regular connectives	30
2	Intuitionistic linear logic	31
2.1	Sequent Calculus	31
2.2	The intuitionistic fragment of linear logic	31
2.3	Input / output polarities	32
3	Polarized linear logic	33
3.1	Polarization	33
3.2	Deduction rules	33
4	Fragment	34
4.1	Motivations	35
4.2	Multiplicative fragments	35
4.3	Additive fragments	36
4.4	Exponential fragments	37
4.4.1	About exponential rules	38
4.5	The provability problem	39
4.5.1	The finite model property	39
4.5.2	Provability	39
4.6	The cut elimination problem	39

5	Proof-nets: a formal account of nets	40
5.1	Preliminaries	40
5.1.1	The short story	40
5.2	Nets	40
5.2.1	Wires	40
5.2.2	Nets	41
5.2.3	Subnets	42
5.2.4	Rewriting	42
5.2.5	Typing	42
5.2.6	Boxes	43
6	System L	44
6.1	Definitions	44
6.2	Typing	44
6.3	Reduction rules	45
7	Translations of intuitionistic logic	46
7.1	Call-by-name Girard's translation $A \rightarrow B \mapsto !A \multimap B$	46
7.2	Call-by-value translation $A \rightarrow B \mapsto !(A \multimap B)$	47
7.2.1	Alternative presentation	48
7.3	Call-by-value Girard's translation $A \rightarrow B \mapsto !(A \multimap B)$	49
8	Translations of classical logic	50
8.1	T-translation $A \rightarrow B \mapsto !?A \multimap ?B$	50
8.1.1	Alternative presentation	52
8.2	Q-translation $A \rightarrow B \mapsto !(A \multimap ?B)$	52
8.2.1	Alternative presentation	54
9	Light linear logics	56
9.1	Elementary linear logic	56
9.2	Light linear logic	57
9.3	Soft linear logic	57
II	Semantics	58
10	Semantics	59
10.1	Orthogonality relation	59
11	Coherent semantics	60
11.1	The cartesian closed structure of coherent semantics	60
11.1.1	Coherent spaces	60
11.1.1.1	As domains	60
11.1.1.2	As graphs	61
11.1.1.3	As cliqued spaces	61
11.1.1.4	Equivalence of definitions	61
11.1.2	Stable functions	62
11.1.3	Cartesian product	63
11.2	The monoidal structure of coherent semantics	63
11.2.1	Linear functions	63
11.2.2	Tensor product	64

11.2.3	Linear negation	64
11.3	Exponentials	65
11.4	Dual connectives and neutrals	65
11.4.1	The direct sum	65
11.4.2	The par and the why not	66
11.4.3	One and bottom	66
11.4.4	Zero and top	66
11.5	After coherent semantics	66
11.5.1	Sequentiality	67
11.5.1.1	The parallel or is not stable	67
11.5.1.2	The Gustave function is stable	68
11.5.2	Multiplicative neutrals and the mix rule	68
12	Phase semantics	70
12.1	Introduction	70
12.2	Preliminaries: relation and closure operators	70
12.2.1	Relations and operators on subsets	70
12.2.2	Closure operators	71
12.3	Phase Semantics	72
12.3.1	Phase spaces	72
12.3.2	Additive connectives	72
12.3.3	Multiplicative connectives	73
12.3.4	Properties	73
12.3.5	Exponentials	74
12.4	Soundness	74
12.5	Completeness	74
12.6	Cut elimination	75
12.7	The Rest	75
13	Categorical semantics	76
13.1	Basic category theory recalled	76
13.2	Overview	76
13.3	Modeling IMLL	77
13.4	Modeling the additives	78
13.5	Modeling ILL	79
13.6	Modeling negation	80
13.6.1	*-autonomous categories	80
13.6.2	Compact closed categories	81
13.7	Other categorical models	82
13.7.1	Lafont categories	82
13.7.2	Seely categories	82
13.7.3	Linear categories	82
13.8	Properties of categorical models	82
13.8.1	The Kleisli category	82
14	Relational semantics	83
14.1	The category of sets and relations	83
14.1.1	Monoidal structure	83
14.1.2	Additives	84
14.1.3	Exponentials	84

14.2	Interpretation of propositional linear logic (LL_0)	84
15	Finiteness semantics	86
15.1	Finiteness spaces	86
15.1.1	Multiplicatives	87
15.1.2	Additives	87
15.1.3	Exponentials	88
16	Geometry of interaction	89
16.1	Introduction	89
16.1.1	The Geometry of Interaction as operators	89
16.1.1.1	Partial isometries	90
16.1.1.2	The $*$ -autonomous structure	90
16.1.1.3	The exponentials	90
16.1.2	The Geometry of Interaction as an abstract machine	91
16.2	GoI for MELL: partial isometries	91
16.2.1	Operators, partial isometries	91
16.2.2	Partial permutations	92
16.2.3	The proof space	93
16.2.4	From operators to matrices: internalization/externalization	94
16.3	GoI for MELL: the $*$ -autonomous structure	95
16.3.1	The tensor and the linear application	95
16.3.2	The identity	95
16.3.3	The execution formula, version 1: application	96
16.3.4	The tensor rule	98
16.3.5	Other monoidal constructions	99
16.3.5.1	Contraposition	99
16.3.5.2	Commutativity	99
16.3.5.3	Distributivity	100
16.3.5.4	Weak distributivity	100
16.3.6	Execution formula, version 2: composition	100
16.4	GoI for MELL: exponentials	101
16.4.1	The tensor product of Hilbert spaces	101
16.4.1.1	An algebra isomorphism	101
16.4.1.2	Interpretation of exponentials	102
17	Game semantics	103
17.1	Preliminary definitions and notations	103
17.1.1	Sequences, Polarities	103
17.1.2	Sequences on Components	103
17.2	Games and Strategies	104
17.2.1	Game constructions	104
17.2.2	Strategies	104
A	Notations	106
A.1	Logical systems	106
A.2	Formulas and proof trees	106
A.2.1	Formulas	106
A.2.2	Rule names	106
A.3	Semantics	107

A.3.1	Coherent spaces	107
A.3.2	Finiteness spaces	107
A.4	Nets	107
A.5	Miscellaneous	107

Part I
Syntax

Chapter 1

Sequent calculus

This article presents the language and sequent calculus of second-order linear logic and the basic properties of this sequent calculus. The core of the article 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 at the end of the page.

1.1 Formulas

Atomic formulas, written α, β, γ , are predicates of the form $p(t_1, \dots, t_n)$, 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. 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 (see chapter [A](#)).

Formulas, represented by capital letters A, B, C , are built using the following connectives:

α	atom	A^\perp	negation
$A \otimes B$	tensor	$A \wp B$	par
1	one	\perp	bottom
$A \oplus B$	plus	$A \& B$	with
0	zero	\top	top
$!A$	of course	$?A$	why not
$\exists \xi. A$	there exists	$\forall \xi. A$	for all

Each line (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](#). The *tensor* and *with* connectives are conjunctions while *par* and *plus* are disjunctions. The exponential connectives are called *modalities*, and traditionally read *of course A* for $!A$ and *why not A* for $?A$. Quantifiers may apply to first- or second-order variables.

There is no connective for implication in the syntax of standard linear logic. Instead, a *linear implication* is defined similarly to the decomposition $A \rightarrow B = \neg A \vee B$ in classical logic, as $A \multimap B := A^\perp \wp B$.

Free and bound variables and first-order substitution 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 and $B[x_1, \dots, x_n]$ is a formula with variables x_i , then the formula $A[B/X]$ is A where every atom $X(t_1, \dots, t_n)$ is replaced by $B[t_1, \dots, t_n]$.

1.2 Sequents and proofs

A sequent is an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite multisets of formulas. For a multiset $\Gamma = A_1, \dots, A_n$, the notation $? \Gamma$ represents the multiset $?A_1, \dots, ?A_n$. Proofs are labelled trees of sequents, built using the following inference rules:

- Identity group:

$$\frac{}{A \vdash A} \text{ axiom} \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \text{ cut}$$

- Negation:

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, A^\perp \vdash \Delta} n_L \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash A^\perp, \Delta} n_R$$

- Multiplicative group:

- tensor:

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta} \otimes_L \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'} \otimes_R$$

- par:

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \wp B \vdash \Delta, \Delta'} \wp_L \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \wp B, \Delta} \wp_R$$

- one:

$$\frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta} 1_L \qquad \frac{}{\vdash 1} 1_R$$

- bottom:

$$\frac{}{\perp \vdash} \perp_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta} \perp_R$$

- Additive group:

- plus:

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

- with:

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

- zero:

$$\frac{}{\Gamma, 0 \vdash \Delta} 0_L$$

- top:

$$\frac{}{\Gamma \vdash \top, \Delta} \top_R$$

- Exponential group:

- of course:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, !A \vdash \Delta} d_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma, !A \vdash \Delta} w_L \qquad \frac{\Gamma, !A, !A \vdash \Delta}{\Gamma, !A \vdash \Delta} c_L$$

$$\frac{!A_1, \dots, !A_n \vdash B, ?B_1, \dots, ?B_m}{!A_1, \dots, !A_n \vdash !B, ?B_1, \dots, ?B_m} !_R$$

– why not:

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash ?A, \Delta} d_R \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash ?A, \Delta} w_R \quad \frac{\Gamma \vdash ?A, ?A, \Delta}{\Gamma \vdash ?A, \Delta} c_R$$

$$\frac{!A_1, \dots, !A_n, A \vdash ?B_1, \dots, ?B_m}{!A_1, \dots, !A_n, ?A \vdash ?B_1, \dots, ?B_m} ?_L$$

- Quantifier group (in the \exists_L and \forall_R rules, ξ must not occur free in Γ or Δ):

– there exists:

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists \xi. A \vdash \Delta} \exists_L \quad \frac{\Gamma \vdash \Delta, A[t/x]}{\Gamma \vdash \Delta, \exists x. A} \exists_R^1 \quad \frac{\Gamma \vdash \Delta, A[B/X]}{\Gamma \vdash \Delta, \exists X. A} \exists_R^2$$

– for all:

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x. A \vdash \Delta} \forall_L^1 \quad \frac{\Gamma, A[B/X] \vdash \Delta}{\Gamma, \forall X. A \vdash \Delta} \forall_L^2 \quad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, \forall \xi. A} \forall_R$$

The left rules for *of course* and right rules for *why not* are called *dereliction*, *weakening* and *contraction* rules. The right rule for *of course* and the left rule for *why not* are called *promotion* rules. Note the fundamental fact that there are no contraction and weakening rules for arbitrary formulas, but only for the formulas starting with the $?$ modality. This is what distinguishes linear logic from classical logic: if weakening and contraction were allowed for arbitrary formulas, then \otimes and $\&$ would be identified, as well as \oplus and \wp , 1 and \top , 0 and \perp . By *identified*, we mean here that replacing a \otimes with a $\&$ or vice versa would preserve provability.

Sequents are considered as multisets, in other words as sequences up to permutation. An alternative presentation would be to define a sequent as a finite sequence of formulas and to add the exchange rules:

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta} \text{exchange}_L \quad \frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2} \text{exchange}_R$$

1.3 Equivalences

Two formulas A and B are (linearly) equivalent, written $A \circ\!\circ B$, if both implications $A \multimap B$ and $B \multimap A$ are provable. Equivalently, $A \circ\!\circ B$ if both $A \vdash B$ and $B \vdash A$ are provable. Another formulation of $A \circ\!\circ B$ is that, for all Γ and Δ , $\Gamma \vdash \Delta, A$ is provable if and only if $\Gamma \vdash \Delta, B$ is provable.

Two related notions are **isomorphism** (stronger than equivalence) and **equiprovability** (weaker than equivalence).

1.3.1 De Morgan laws

Negation is involutive: $A \circ\!\circ A^{\perp\perp}$

Duality between connectives:

$$\begin{array}{cc|cc} \hline (A \otimes B)^\perp & \circ\!\circ A^\perp \wp B^\perp & (A \wp B)^\perp & \circ\!\circ A^\perp \otimes B^\perp \\ 1^\perp & \circ\!\circ \perp & \perp^\perp & \circ\!\circ 1 \\ (A \oplus B)^\perp & \circ\!\circ A^\perp \& B^\perp & (A \& B)^\perp \circ\!\circ A^\perp \oplus B^\perp \\ 0^\perp & \circ\!\circ \top & \top^\perp & \circ\!\circ 0 \\ \hline \end{array}$$

$$\begin{array}{cccc} (!A)^\perp & \circ\text{-} & ?(A^\perp) & \circ\text{-} & !(A^\perp) \\ (\exists\xi.A)^\perp & \circ\text{-} & \forall\xi.(A^\perp) & \circ\text{-} & \exists\xi.(A^\perp) \end{array}$$

1.3.2 Fundamental equivalences

- Associativity, commutativity, neutrality:

$$\begin{array}{llll} A \otimes (B \otimes C) \circ\text{-} (A \otimes B) \otimes C & A \otimes B \circ\text{-} B \otimes A & A \otimes 1 \circ\text{-} A \\ A \wp (B \wp C) \circ\text{-} (A \wp B) \wp C & A \wp B \circ\text{-} B \wp A & A \wp \perp \circ\text{-} A \\ A \oplus (B \oplus C) \circ\text{-} (A \oplus B) \oplus C & A \oplus B \circ\text{-} B \oplus A & A \oplus 0 \circ\text{-} A \\ A \& (B \& C) \circ\text{-} (A \& B) \& C & A \& B \circ\text{-} B \& A & A \& \top \circ\text{-} A \end{array}$$

- Idempotence of additives:

$$A \oplus A \circ\text{-} A \quad A \& A \circ\text{-} A$$

- Distributivity of multiplicatives over additives:

$$\begin{array}{ll} A \otimes (B \oplus C) \circ\text{-} (A \otimes B) \oplus (A \otimes C) & A \otimes 0 \circ\text{-} 0 \\ A \wp (B \& C) \circ\text{-} (A \wp B) \& (A \wp C) & A \wp \top \circ\text{-} \top \end{array}$$

- Defining property of exponentials:

$$\begin{array}{ll} !(A \& B) \circ\text{-} !A \otimes !B & !\top \circ\text{-} 1 \\ ?(A \oplus B) \circ\text{-} ?A \wp ?B & ?0 \circ\text{-} \perp \end{array}$$

- Monoidal structure of exponentials:

$$\begin{array}{ll} !A \otimes !A \circ\text{-} !A & !1 \circ\text{-} 1 \\ ?A \wp ?A \circ\text{-} ?A & ?\perp \circ\text{-} \perp \end{array}$$

- Digging:

$$!!A \circ\text{-} !A \quad ??A \circ\text{-} ?A$$

- Other properties of exponentials:

$$\begin{array}{ll} !?!A \circ\text{-} !?A & !?1 \circ\text{-} 1 \\ ??!A \circ\text{-} ?!A & ?!\perp \circ\text{-} \perp \end{array}$$

These properties of exponentials lead to the [lattice of exponential modalities](#).

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

$$\begin{array}{llll} \exists\xi.\exists\psi.A \circ\text{-} \exists\psi.\exists\xi.A & \exists\xi.(A \oplus B) \circ\text{-} \exists\xi.A \oplus \exists\xi.B & \exists\zeta.(A \otimes B) \circ\text{-} A \otimes \exists\zeta.B & \exists\zeta.A \circ\text{-} A \\ \forall\xi.\forall\psi.A \circ\text{-} \forall\psi.\forall\xi.A & \forall\xi.(A \& B) \circ\text{-} \forall\xi.A \& \forall\xi.B & \forall\zeta.(A \wp B) \circ\text{-} A \wp \forall\zeta.B & \forall\zeta.A \circ\text{-} A \end{array}$$

1.3.3 Definability

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

- $0 \circ\text{-} \forall X.X$
- $1 \circ\text{-} \forall X.(X \multimap X)$
- $A \oplus B \circ\text{-} \forall X.(!(A \multimap X) \multimap !(B \multimap X) \multimap X)$

The constants \top and \perp and the connective $\&$ can be defined by duality.

Any pair of connectives that has the same rules as \otimes/\wp is equivalent to it, the same holds for additives, but not for exponentials.

Other [basic equivalences](#) exist.

1.4 Properties of proofs

1.4.1 Cut elimination and consequences

Theorem 1 (cut elimination). *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.*

This property is proved using a set of rewriting rules on proofs, using appropriate termination arguments (see the specific articles on [cut elimination](#) for detailed proofs), it is the core of the proof/program correspondence.

It has several important consequences:

Definition 1 (subformula). *The subformulas of a formula A are A and, inductively, the subformulas of its immediate subformulas:*

- the immediate subformulas of $A \otimes B$, $A \wp B$, $A \oplus B$, $A \& B$ are A and B ,
- the only immediate subformula of $!A$ and $?A$ is A ,
- 1 , \perp , 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 2 (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 premisses are either formulas of the conclusion, or immediate subformulas of it, therefore cut-free proofs have the subformula property. \square

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 3 (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 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 left negation rule $A^\perp \vdash$ is provable, and by the cut rule one gets empty conclusion, which is not possible. As particular cases, since 1 and \top are provable, \perp and 0 are not, since they are equivalent to 1^\perp and \top^\perp respectively. \square

1.4.2 Expansion of identities

Let us write $\pi : \Gamma \vdash \Delta$ to signify that π is a proof with conclusion $\Gamma \vdash \Delta$.

Proposition 1 (η -expansion). *For every proof π , there is a proof π' with the same conclusion as π in which the axiom rule is only used with atomic formulas. If π is cut-free, then there is a cut-free π' .*

Proof. It suffices to prove that for every formula A , the sequent $A \vdash A$ has a cut-free proof in which the axiom rule is used only for atomic formulas. We prove this by induction on A .

- If A is atomic, then $A \vdash A$ is an instance of the atomic axiom rule.
- If $A = A_1 \otimes A_2$ then we have:

$$\frac{\frac{\pi_1 : A_1 \vdash A_1 \quad \pi_2 : A_2 \vdash A_2}{A_1, A_2 \vdash A_1 \otimes A_2} \otimes_R}{A_1 \otimes A_2 \vdash A_1 \otimes A_2} \otimes_L$$

where π_1 and π_2 exist by induction hypothesis.

- If $A = A_1 \wp A_2$ then we have:

$$\frac{\frac{\pi_1 : A_1 \vdash A_1 \quad \pi_2 : A_2 \vdash A_2}{A_1 \wp A_2 \vdash A_1, A_2} \wp_L}{A_1 \wp A_2 \vdash A_1 \wp A_2} \wp_R$$

where π_1 and π_2 exist by induction hypothesis.

- All other connectives follow the same pattern. □

The interesting thing with η -expansion is that, we can always assume that each connective is explicitly introduced by its associated rule (except in the case where there is an occurrence of the \top rule).

1.4.3 Reversibility

Definition 2 (reversibility). *A connective c is called reversible if*

- *for every proof $\pi : \Gamma \vdash \Delta, c(A_1, \dots, A_n)$, there is a proof π' with the same conclusion in which $c(A_1, \dots, A_n)$ is introduced by the last rule,*
- *if π is cut-free then there is a cut-free π' .*

Proposition 2. *The connectives $\wp, \perp, \&, \top$ and \forall are reversible.*

Proof. Using the η -expansion property, we assume that the axiom rule is only applied to atomic formulas. Then each top-level connective is introduced either by its associated (left or right) rule or in an instance of the 0_L or \top_R rule.

For \wp , consider a proof $\pi : \Gamma \vdash \Delta, A \wp B$. If $A \wp B$ is introduced by a \wp_R rule (not necessarily the last rule in π), then if we remove this rule we get a proof of $\vdash \Gamma, A, B$ (this can be proved by a straightforward induction on π). If it is introduced in the context of a 0_L or \top_R rule, then this rule can be changed so that $A \wp B$ is replaced by A, B . In either case, we can apply a final \wp rule to get the expected proof.

For \perp , the same technique applies: if it is introduced by a \perp_R rule, then remove this rule to get a proof of $\vdash \Gamma$, if it is introduced by a 0_L or \top_R rule, remove the \perp from this rule, then apply the \perp rule at the end of the new proof.

For $\&$, consider a proof $\pi : \Gamma \vdash \Delta, A \& B$. If the connective is introduced by a $\&$ rule then this rule is applied in a context like

$$\frac{\pi_1 : \Gamma' \vdash \Delta', A \quad \pi_2 : \Gamma' \vdash \Delta', B}{\Gamma' \vdash \Delta', A \& B} \&$$

Since the formula $A \& B$ is not involved in other rules (except as context), if we replace this step by π_1 in π we finally get a proof $\pi'_1 : \Gamma \vdash \Delta, A$. If we replace this step by π_2 we get a proof $\pi'_2 : \Gamma \vdash \Delta, B$. Combining π_1 and π_2 with a final $\&$ rule we finally get the expected proof. The case when the $\&$ was introduced in a \top rule is solved as before.

For \top the result is trivial: just choose π' as an instance of the \top rule with the appropriate conclusion.

For \forall , consider a proof $\pi : \Gamma \vdash \Delta, \forall \xi. A$. Up to renaming, we can assume that ξ occurs free only above the rule that introduces the quantifier. If the quantifier is introduced by a \forall rule, then if we remove this rule, we can check that we get a proof of $\Gamma \vdash \Delta, A$ on which we can finally apply the \forall rule. The case when the \forall was introduced in a \top rule is solved as before.

Note that, in each case, if the proof we start from is cut-free, our transformations do not introduce a cut rule. However, if the original proof has cuts, then the final proof may have more cuts, since in the case of $\&$ we duplicated a part of the original proof. \square

A corresponding property for positive connectives is **focusing**, which states that clusters of positive formulas can be treated in one step, under certain circumstances.

1.5 One-sided sequent calculus

The sequent calculus presented above is very symmetric: 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, a sequent $\Gamma, A \vdash \Delta$ is provable if and only if the sequent $\Gamma \vdash A^\perp, \Delta$ is provable. From these remarks, we can define an equivalent one-sided sequent calculus:

- Formulas are considered up to De Morgan duality. Equivalently, one can consider that negation is not a connective but a syntactically defined operation on formulas. In this case, negated atoms α^\perp must be considered as another kind of atomic formulas.
- Sequents have the form $\vdash \Gamma$.

The inference rules are essentially the same except that the left hand side of sequents is kept empty:

- Identity group:

$$\frac{}{\vdash A^\perp, A} \text{ axiom} \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \text{ cut}$$

- Multiplicative group:

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \otimes \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} \wp \quad \frac{}{\vdash 1} 1 \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \perp$$

- Additive group:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \oplus_1 \quad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \oplus_2 \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \& \quad \frac{}{\vdash \Gamma, \top} \top$$

- Exponential group:

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} d \quad \frac{\vdash \Gamma}{\vdash \Gamma, ?A} w \quad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} c \quad \frac{\vdash ?\Gamma, B}{\vdash ?\Gamma, !B} !$$

- Quantifier group (in the \forall rule, ξ must not occur free in Γ):

$$\frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x.A} \exists^1 \quad \frac{\vdash \Gamma, A[B/X]}{\vdash \Gamma, \exists X.A} \exists^2 \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, \forall \xi.A} \forall$$

Theorem 4. *A two-sided sequent $\Gamma \vdash \Delta$ is provable if and only if the sequent $\vdash \Gamma^\perp, \Delta$ is provable in the one-sided system.*

The one-sided system enjoys the same properties as the two-sided one, including cut elimination, the subformula property, etc. This formulation is often used when studying proofs because it is much lighter than the two-sided form while keeping the same expressiveness. In particular, [proof-nets](#) can be seen as a quotient of one-sided sequent calculus proofs under commutation of rules.

1.6 Variations

1.6.1 Exponential rules

- The promotion rule, on the right-hand side for example, $\frac{!A_1, \dots, !A_n \vdash B, ?B_1, \dots, ?B_m}{!A_1, \dots, !A_n \vdash !B, ?B_1, \dots, ?B_m} !_R$ can be replaced by a *multi-functorial* promotion rule $\frac{A_1, \dots, A_n \vdash B, B_1, \dots, B_m}{!A_1, \dots, !A_n \vdash !B, ?B_1, \dots, ?B_m} !_R \text{mf}$ and a *digging* rule $\frac{\Gamma \vdash ??A, \Delta}{\Gamma \vdash ?A, \Delta} ??$, without modifying the provability.

Note that digging violates the subformula property.

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

1.6.2 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.

1.6.3 Mix rules

It is quite common to consider **mix rules**: $\frac{}{\vdash} \text{Mix}_0$ $\frac{\Gamma \vdash \Delta \quad \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \text{Mix}_2$

1.7 Equiprovability

1.7.1 Definition

Two formulas A and B are equiprovable, when $\vdash A$ is provable if and only if $\vdash B$ is provable.

- for any A and B , $A \otimes B$ and $A \& B$ are equiprovable.
- for any A , A , $!A$ and $\forall \xi A$ are equiprovable.

1.8 Isomorphism

1.8.1 Definition

Two formulas A and B are isomorphic (denoted $A \cong B$), when there are two proofs π of $A \vdash B$ and ρ of $B \vdash A$ such that eliminating the cut on A in

$$\frac{\begin{array}{c} \vdots \pi \\ A \vdash B \end{array} \quad \begin{array}{c} \vdots \rho \\ B \vdash A \end{array}}{B \vdash B} \text{cut}$$

leads to an η -expansion of $\frac{}{B \vdash B} \text{ax}$,

and eliminating the cut on B in

$$\frac{\begin{array}{c} \vdots \pi \\ A \vdash B \end{array} \quad \begin{array}{c} \vdots \rho \\ B \vdash A \end{array}}{A \vdash A} \text{cut}$$

leads to an η -expansion of $\frac{}{A \vdash A} \text{ax}$.

Linear logic admits **many isomorphisms**, but it is not known whether all of them have been discovered or not.

1.8.2 List of isomorphisms

1.8.2.1 Linear negation

$$\begin{array}{l} A^{\perp\perp} \cong A \\ (A \otimes B)^\perp \cong A^\perp \wp B^\perp \qquad 1^\perp \cong \perp \\ (A \wp B)^\perp \cong A^\perp \otimes B^\perp \qquad \perp^\perp \cong 1 \\ (A \& B)^\perp \cong A^\perp \oplus B^\perp \qquad \top^\perp \cong 0 \\ (A \oplus B)^\perp \cong A^\perp \& B^\perp \qquad 0^\perp \cong \top \\ (!A)^\perp \cong ?A^\perp \\ (?A)^\perp \cong !A^\perp \end{array}$$

1.8.2.2 Neutrals

$$\begin{array}{l} A \otimes 1 \cong 1 \otimes A \cong A \\ A \wp \perp \cong \perp \wp A \cong A \\ A \& \top \cong \top \& A \cong A \\ A \oplus 0 \cong 0 \oplus A \cong A \end{array}$$

1.8.2.3 Commutativity

$$\begin{aligned} A \otimes B &\cong B \otimes A \\ A \wp B &\cong B \wp A \\ A \& B &\cong B \& A \\ A \oplus B &\cong B \oplus A \end{aligned}$$

1.8.2.4 Associativity

$$\begin{aligned} (A \otimes B) \otimes C &\cong A \otimes (B \otimes C) \\ (A \wp B) \wp C &\cong A \wp (B \wp C) \\ (A \& B) \& C &\cong A \& (B \& C) \\ (A \oplus B) \oplus C &\cong A \oplus (B \oplus C) \end{aligned}$$

1.8.2.5 Multiplicative-additive distributivity

$$\begin{aligned} A \otimes (B \oplus C) &\cong (A \otimes B) \oplus (A \otimes C) & A \otimes 0 &\cong 0 \\ A \wp (B \& C) &\cong (A \wp B) \& (A \wp C) & A \wp \top &\cong \top \end{aligned}$$

1.8.2.6 Linear implication

$$\begin{aligned} A \multimap B &\cong A^\perp \wp B \\ A \multimap B &\cong B^\perp \multimap A^\perp \\ A \otimes B \multimap C &\cong A \multimap B \multimap C \end{aligned}$$

1.8.2.7 The exponential isomorphisms

$$\begin{aligned} !(A \& B) &\cong !A \otimes !B & !\top &\cong 1 \\ ?(A \oplus B) &\cong ?A \wp ?B & ?0 &\cong \perp \end{aligned}$$

1.8.2.8 Quantifiers

$$\begin{aligned} \forall \xi_1. \forall \xi_2. A &\cong \forall \xi_2. \forall \xi_1. A \\ \exists \xi_1. \exists \xi_2. A &\cong \exists \xi_2. \exists \xi_1. A \\ \forall \xi. (A \wp B) &\cong A \wp \forall \xi. B \quad (\xi \notin A) \\ \exists \xi. (A \otimes B) &\cong A \otimes \exists \xi. B \quad (\xi \notin A) \\ \forall \xi. (A \& B) &\cong (\forall \xi. A) \& (\forall \xi. B) & \forall \xi. \top &\cong \top \\ \exists \xi. (A \oplus B) &\cong (\exists \xi. A) \oplus (\exists \xi. B) & \exists \xi. 0 &\cong 0 \end{aligned}$$

1.9 List of equivalences

Each [isomorphism](#) gives an equivalence of formulas. The following equivalences are *not* isomorphisms.

1.9.1 Multiplicatives

$$\begin{aligned} A \multimap A &\multimap A \otimes (A^\perp \wp A) \multimap (A \otimes A^\perp) \wp A \\ A \wp A^\perp &\multimap (A \wp A^\perp) \otimes (A \wp A^\perp) \end{aligned}$$

1.9.2 Additives

$$\begin{aligned} A \& A & \multimap A \\ A \oplus A & \multimap A \end{aligned}$$

$$\begin{aligned} A \& (A \oplus B) & \multimap A & A \oplus \top & \multimap \top \\ A \oplus (A \& B) & \multimap A & A \& 0 & \multimap 0 \end{aligned}$$

1.9.3 Quantifiers

$$\begin{aligned} \forall X.A & \multimap A & (X \notin A) \\ \exists X.A & \multimap A & (X \notin A) \end{aligned}$$

1.9.4 Exponentials

$$\begin{aligned} !A & \multimap !A \otimes !A & ?A & \multimap ?A \wp ?A \\ !A & \multimap !!A & ?A & \multimap ??A \\ !?A & \multimap !?!A & ?!A & \multimap ?!?!A \end{aligned}$$

Some of these equivalences are related with the [lattice of exponential modalities](#).

1.9.5 Polarities

$$\begin{array}{l} \hline ?N \multimap N \quad (\text{N negative}) \\ !P \multimap P \quad (\text{P positive}) \\ ?!R \multimap R \quad (\text{R regular}) \\ !?L \multimap L \quad (\text{L co-regular}) \\ \hline \end{array}$$

1.9.6 Second order encodings

$$\begin{aligned} A & \multimap \forall X.(A \otimes X^\perp) \wp X \\ A & \multimap \exists X.(A \wp X^\perp) \otimes X \end{aligned}$$

$$\begin{aligned} A \& B & \multimap \exists X.!(A \wp X^\perp) \otimes !(B \wp X^\perp) \otimes X & \top & \multimap \exists X.X \\ A \oplus B & \multimap \forall X.?(A \otimes X^\perp) \wp ?(B \otimes X^\perp) \wp X & 0 & \multimap \forall X.X \end{aligned}$$

$$\begin{aligned} \perp & \multimap \exists X.X \otimes X^\perp \\ 1 & \multimap \forall X.X^\perp \wp X \end{aligned}$$

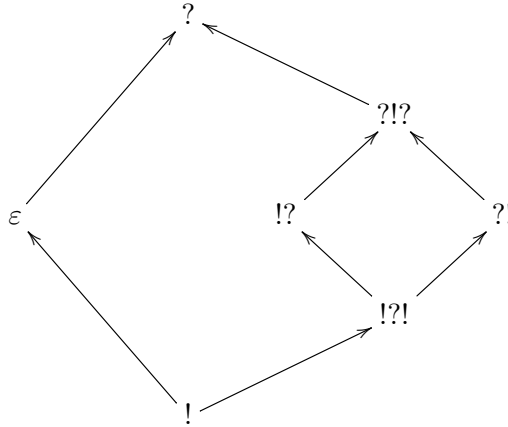
$$\begin{aligned} \forall \xi.A & \multimap \exists X.(\forall \xi.(A \wp X^\perp)) \otimes X \\ \exists \xi.A & \multimap \forall X.(\exists \xi.(A \otimes X^\perp)) \wp X \end{aligned}$$

1.9.7 Miscellaneous

$$\begin{aligned} 1 & \multimap !(A^\perp \wp A) \\ \perp & \multimap ?(A^\perp \otimes A) \end{aligned}$$

$$\begin{aligned} !?(!A \& !B) & \multimap !(?!A \& ?!B) \\ ?!(?A \oplus ?B) & \multimap ?(!?A \oplus !?B) \end{aligned}$$

1.10 Lattice of exponential modalities



An *exponential modality* is an arbitrary (possibly empty) sequence of the two exponential connectives $!$ and $?$. It can be considered itself as a unary connective. This leads to the notation μA for applying an exponential modality μ to a formula A .

There is a preorder relation on exponential modalities defined by $\mu \lesssim \nu$ if and only if for any formula A we have $\mu A \vdash \nu A$. It induces an [equivalence](#) relation on exponential modalities by $\mu \sim \nu$ if and only if $\mu \lesssim \nu$ and $\nu \lesssim \mu$.

Lemma 1. For any formula A , $!A \vdash A$ and $A \vdash ?A$.

Lemma 2 (Functoriality). If A and B are two formulas such that $A \vdash B$ then, for any exponential modality μ , $\mu A \vdash \mu B$.

Lemma 3. For any formula A , $!A \vdash !!A$ and $??A \vdash ?A$.

Lemma 4. For any formula A , $!A \vdash !?!A$ and $?!A \vdash ?A$.

This allows to prove that any exponential modality is equivalent to one of the following seven modalities: ε (the empty modality), $!$, $?$, $!?$, $?!$, $!?!$ or $!!!$. Indeed any sequence of consecutive $!$ or $?$ in a modality can be simplified into only one occurrence, and then any alternating sequence of length at least four can be simplified into a smaller one.

Proof. We obtain $!!A \vdash !A$ by functoriality from $!A \vdash A$ (and similarly for $?A \vdash ??A$). From $!A \vdash !?!A$, we deduce $?!A \vdash ?!?!A$ by functoriality and $!B \vdash !?!B$ (with $A = ?B$). In a similar way we have $!?!A \vdash !?A$ and $?!?!A \vdash ?!A$. \square

The order relation induced on equivalence classes of exponential modalities with respect to \sim can be proved to be the one represented on the picture in the top of this section. All the represented relations are valid.

Proof. We have already seen $!A \vdash A$ and $!A \vdash !?!A$. By functoriality we deduce $!?!A \vdash !?A$ and by $A = ?!B$ we deduce $!?!B \vdash ?!B$.

The others are obtained from these ones by duality: $A \vdash B$ entails $B^\perp \vdash A^\perp$. \square

Lemma 5. If α is an atom, $? \alpha \not\vdash \alpha$ and $\alpha \not\vdash ?! \alpha$.

We can prove that no other relation between classes is true (by relying on the previous lemma).

Proof. From the lemma and $A \vdash ?A$, we have $? \alpha \not\vdash ?! ? \alpha$.

Then $?$ cannot be smaller than any other of the seven modalities (since they are all smaller than ε or $?!$). For the same reason, ε cannot be smaller than $!$, $!?$, $?!$ or $!?!$. This entails that $?!$ is only smaller than $?$ since it is not smaller than ε (by duality from ε not smaller than $!?!$).

From these, $?!\alpha \not\vdash !? \alpha$ and $! ? \alpha \not\vdash ?!\alpha$, we deduce that no other relation is possible. \square

The order relation on equivalence classes of exponential modalities is a lattice.

1.11 Provable formulas

Important provable formulas are given by [isomorphisms](#) and by [equivalences](#).

In many of the cases below the [converse implication does not hold](#).

1.11.1 Distributivities

1.11.1.1 Standard distributivities

$$\begin{aligned} A \oplus (B \& C) &\multimap (A \oplus B) \& (A \oplus C) \\ A \otimes (B \& C) &\multimap (A \otimes B) \& (A \otimes C) \\ \exists \xi.(A \& B) &\multimap (\exists \xi.A) \& (\exists \xi.B) \end{aligned}$$

1.11.1.2 Linear distributivities

$$\begin{aligned} A \otimes (B \wp C) &\multimap (A \otimes B) \wp C \\ \exists \xi.(A \wp B) &\multimap A \wp \exists \xi.B & (\xi \notin A) \\ A \otimes \forall \xi.B &\multimap \forall \xi.(A \otimes B) & (\xi \notin A) \end{aligned}$$

1.11.2 Factorizations

$$\begin{aligned} (A \& B) \oplus (A \& C) &\multimap A \& (B \oplus C) \\ (A \wp B) \oplus (A \wp C) &\multimap A \wp (B \oplus C) \\ (\forall \xi.A) \oplus (\forall \xi.B) &\multimap \forall \xi.(A \oplus B) \end{aligned}$$

1.11.3 Identities

$$\begin{aligned} 1 &\multimap A^\perp \wp A \\ A \otimes A^\perp &\multimap \perp \end{aligned}$$

1.11.4 Additive structure

$$\begin{array}{lll} A \& B &\multimap A & A \& B &\multimap B & A &\multimap \top \\ A &\multimap A \oplus B & B &\multimap A \oplus B & 0 &\multimap A \end{array}$$

1.11.5 Quantifiers

$$\begin{array}{ll}
A \multimap \forall \xi. A & (\xi \notin A) \\
\exists \xi. A \multimap A & (\xi \notin A) \\
\\
\forall \xi_1. \forall \xi_2. A \multimap \forall \xi. A[\xi/\xi_1, \xi/\xi_2] \\
\exists \xi. A[\xi/\xi_1, \xi/\xi_2] \multimap \exists \xi_1. \exists \xi_2. A
\end{array}$$

1.11.6 Exponential structure

Provable formulas involving exponential connectives only provide us with the [lattice of exponential modalities](#).

$$\begin{array}{ll}
!A \multimap A & A \multimap ?A \\
!A \multimap 1 & \perp \multimap ?A
\end{array}$$

1.11.7 Monoidality of exponentials

$$\begin{array}{ll}
?(A \wp B) \multimap ?A \wp ?B \\
!A \otimes !B \multimap !(A \otimes B) \\
\\
!(A \& B) \multimap !A \& !B \\
?A \oplus ?B \multimap ?(A \oplus B) \\
\\
?(A \& B) \multimap ?A \& ?B \\
!A \oplus !B \multimap !(A \oplus B)
\end{array}$$

1.11.8 Promotion principles

$$\begin{array}{ll}
!A \otimes ?B \multimap ?(A \otimes B) \\
!(A \wp B) \multimap ?A \wp !B
\end{array}$$

1.11.9 Commutations

$$\begin{array}{l}
\exists \xi. ?A \multimap ?\exists \xi. A \\
! \forall \xi. A \multimap \forall \xi. !A \\
? \forall \xi. A \multimap \forall \xi. ?A \\
\exists \xi. !A \multimap !\exists \xi. A
\end{array}$$

1.12 Non provable formulas

$$\begin{array}{l}
A \& (B \oplus C) \not\multimap (A \& B) \oplus (A \& C) \\
(A \oplus B) \& (A \oplus C) \not\multimap A \oplus (B \& C) \\
(A \otimes B) \wp C \not\multimap A \otimes (B \wp C) \\
A \not\multimap !A \\
! ?!A \not\multimap A \\
! ?!A \not\multimap !A \\
! ?A \not\multimap ?!A \\
?!A \not\multimap !?A
\end{array}$$

1.13 Mix

The usual notion of Mix is the binary version of the rule but a nullary version also exists.

1.13.1 Binary Mix rule

$$\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta} \text{Mix}_2$$

The Mix_2 rule is equivalent to $\perp \vdash 1$:

$$\frac{\frac{\overline{\vdash 1} \quad 1}{\vdash 1, 1} \quad \frac{\overline{\vdash 1} \quad 1}{\vdash 1, 1}}{\vdash 1, 1} \text{Mix}_2 \quad \frac{\frac{\frac{\vdash \Gamma}{\vdash \Gamma, \perp} \perp \quad \vdash 1, 1}{\vdash \Gamma, 1} \text{cut} \quad \frac{\vdash \Delta}{\vdash \Delta, \perp} \perp}{\vdash \Gamma, \Delta} \text{cut}}$$

They are also equivalent to the principle $A \otimes B \vdash A \wp B$:

$$\frac{\frac{\frac{\overline{\vdash 1} \quad 1}{\vdash 1 \otimes 1} \otimes \quad \frac{\overline{\vdash 1} \quad 1}{\vdash 1 \wp 1} \wp}{\vdash 1 \wp 1} \text{cut} \quad \frac{\frac{\overline{\vdash \perp}, 1}{\vdash \perp \otimes \perp, 1, 1} \text{ax} \quad \frac{\overline{\vdash \perp}, 1}{\vdash \perp, 1} \text{ax}}{\vdash 1, 1} \otimes}{\vdash 1, 1} \text{cut} \quad \frac{\frac{\overline{\vdash A^\perp}, A}{\vdash A^\perp, A, B^\perp, B} \text{ax} \quad \frac{\overline{\vdash B^\perp}, B}{\vdash B^\perp, B} \text{ax}}{\vdash A^\perp, A, B^\perp, B} \text{Mix}_2}{\frac{\vdash A^\perp, B^\perp, A \wp B}{\vdash A^\perp \wp B^\perp, A \wp B} \wp} \wp$$

1.13.2 Nullary Mix rule

$$\overline{\vdash} \text{Mix}_0$$

The Mix_0 rule is equivalent to $1 \vdash \perp$:

$$\frac{\frac{\overline{\vdash} \text{Mix}_0}{\vdash \perp} \perp \quad \frac{\overline{\vdash 1} \quad 1}{\vdash \perp, \perp} \text{cut}}{\vdash \perp, \perp} \perp \quad \frac{\frac{\overline{\vdash 1} \quad 1}{\vdash \perp} \text{cut} \quad \overline{\vdash 1} \quad 1}{\vdash} \text{cut}}$$

The nullary Mix acts as a unit for the binary one:

$$\frac{\vdash \Gamma \quad \overline{\vdash} \text{Mix}_0}{\vdash \Gamma} \text{Mix}_2$$

If π is a proof which uses no \perp rule and no weakening rule, then (up to the simplification of the pattern $\text{Mix}_0 / \text{Mix}_2$ above into nothing) π is either reduced to a Mix_0 rule or does not contain any Mix_0 rule.

1.14 Additive cut rule

The additive cut rule is: $\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta} \text{cut add}$

In contrary to what happens in classical logic, this rule is *not* admissible in linear logic.

The formula $\alpha \oplus \alpha^\perp$ is not provable in linear logic, while it is derivable with the additive cut rule:

$$\frac{\frac{\overline{\alpha \vdash \alpha}}{\vdash \alpha, \alpha^\perp} \oplus R2 \quad \frac{\overline{\alpha \vdash \alpha}}{\alpha \vdash \alpha \oplus \alpha^\perp} \oplus R1}{\vdash \alpha \oplus \alpha^\perp} \text{cut add}$$

1.15 Reversibility and focusing

1.15.1 Reversibility

Theorem 5. *Negative connectives are reversible:*

- A sequent $\vdash \Gamma, A \wp B$ is provable if and only if $\vdash \Gamma, A, B$ is provable.
- A sequent $\vdash \Gamma, A \& B$ is provable if and only if $\vdash \Gamma, A$ and $\vdash \Gamma, B$ are provable.
- A sequent $\vdash \Gamma, \perp$ is provable if and only if $\vdash \Gamma$ is provable.
- A sequent $\vdash \Gamma, \forall \xi A$ is provable if and only if $\vdash \Gamma, A$ is provable, for some fresh variable ξ .

Proof. We start with the case of the \wp connective. If $\vdash \Gamma, A, B$ is provable, then by the introduction rule for \wp we know that $\vdash \Gamma, A \wp B$ is provable. For the reverse implication we proceed by induction on a proof π of $\vdash \Gamma, A \wp B$.

- If the last rule of π is the introduction of the \wp in $A \wp B$, then the premiss is exactly $\vdash \Gamma, A, B$ so we can conclude.
- The other case where the last rule introduces $A \wp B$ is when π is an axiom rule, hence $\Gamma = A^\perp \otimes B^\perp$. Then we can conclude with the proof
- Otherwise $A \wp B$ is in the context of the last rule. If the last rule is a tensor, then π has the shape
- or the same with $A \wp B$ in the conclusion of π_2 instead. By induction hypothesis on π_1 we get a proof π'_1 of $\vdash \Gamma_1, A, B, C$, then we can conclude with the proof
- The case of the cut rule has the same structure as the tensor rule.
- In the case of the $\&$ rule, we have $A \& B$ in both premisses and we conclude similarly, using the induction hypothesis on both π_1 and π_2 .
- If $A \wp B$ is in the context of a rules for \wp, \oplus, \perp or quantifiers, or in the context of a dereliction, weakening or contraction, the situation is similar as for \otimes except that we have only one premiss.
- If $A \wp B$ is in the context of \top rules, we can freely change the context of the rule to get the expected one.
- The two remaining cases are if the last rule is the rule for 1 or a promotion. By the constraints these rules impose on the contexts, these cases cannot happen.

The $\&$ connective is treated in the same way. In this cases where $A \& B$ is in the context of a rule with two premisses, the premiss where this formula is not present will be duplicated, with one copy in the premiss for A and one in the premiss for B .

The \forall connective is also treated similarly. Its peculiarity is that introducing $\forall \xi$ requires that ξ does not appear free in the context. For all rules with one premiss except the quantifier rules, the set of fresh variables in the same in the premiss and the conclusion, so everything works well. Other rules might change the set of free variables, but problems are avoided by choosing for ξ a variable that is fresh for the whole proof we are considering. \square

Remark that this result is proved using only commutation rules, except when the formula is introduced by an axiom rule. Furthermore, if axioms are applied only on atoms, this particular case disappears.

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:

- $\vdash \Gamma, (A \wp B) \wp (B \& C)$ is provable
- iff $\vdash \Gamma, A \wp B, B \& C$ is provable
- iff $\vdash \Gamma, A \wp B, B$ and $\vdash \Gamma, A \wp B, C$ are provable
- iff $\vdash \Gamma, A, B, B$ and $\vdash \Gamma, A, B, C$ are provable

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

$$\frac{\frac{\frac{\vdash \Gamma, A, B, B}{\vdash \Gamma, A \wp B, B} \quad \frac{\vdash \Gamma, A, B, C}{\vdash \Gamma, A \wp B, C}}{\vdash \Gamma, A \wp B, B \& C}}{\vdash \Gamma, (A \wp B) \wp (B \& C)}$$

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.

1.15.2 Generalized connectives and rules

Definition 3. A positive generalized connective is a parametrized formula $P[X_1, \dots, X_n]$ made from the variables X_i using the connectives $\otimes, \oplus, 1, 0$.

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

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

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

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

but these rules only differ by the commutation of independent rules. In particular, their premisses 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:

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

These are actually different, in particular their premisses 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 premiss X_1, X_2, X_2 and the first positive inference has three premisses, holding X_1, X_2 and X_2 respectively.
- the second branch of the negative inference has a premiss X_1, X_2, X_3 and the second positive inference has three premisses, holding X_1, X_2 and X_3 respectively.

This phenomenon extends to all generalized connectives.

Definition 4. *The branching of a generalized connective $P[X_1, \dots, X_n]$ is the multiset \mathcal{I}_P of multisets over $\{1, \dots, n\}$ defined inductively as*

$$\mathcal{I}_{P \otimes Q} := [I + J \mid I \in \mathcal{I}_P, J \in \mathcal{I}_Q], \mathcal{I}_{P \oplus Q} := \mathcal{I}_P + \mathcal{I}_Q, \mathcal{I}_1 := [\emptyset], \mathcal{I}_0 := [], \mathcal{I}_{X_i} := [[i]].$$

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

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

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

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

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

Theorem 6. *Let N be a negative generalized connective. A sequent $\vdash \Gamma, N[A_1, \dots, A_n]$ is provable if and only if, for each $[i_1, \dots, i_k] \in \mathcal{I}_N$, the sequent $\vdash \Gamma, A_{i_1}, \dots, A_{i_k}$ is provable.*

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

1.15.3 Focusing

Definition 6. *A formula is positive if it has a main connective among $\otimes, \oplus, 1, 0$. It is called negative if it has a main connective among $\wp, \&, \perp, \top$. 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 7. *A proof $\pi \vdash \Gamma, A$ is said to be positively focused on A if it has the shape*

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

where P is a positive generalized connective, the A_i are non-positive and $A = P[A_1, \dots, A_n]$. 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 7. *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 proof π of Γ . As noted above, the result trivially holds if Γ has a non-positive formula. We can thus assume that Γ contains only positive formulas and reason on the nature of the last rule, which is necessarily the introduction of a positive connective (it cannot be an axiom rule because an axiom always has at least one non-positive conclusion).

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

$$\frac{\rho \vdash \Gamma, A \quad \theta \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$$

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

$$\frac{\rho' \vdash \Gamma, A \quad \theta' \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$$

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

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

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

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

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

If the last rule of π introduces a \oplus , we proceed the same way except that there is only one premiss. If the last rule of π introduces a 1 , then it is the only rule of π , which is thus positively focused on this 1 . \square

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.

1.16.2 Generalized structural rules

Positive formulas admit generalized left structural rules corresponding to a structure of \otimes -comonoid: $P \multimap P \otimes P$ and $P \multimap 1$. The following rules are derivable:

$$\frac{\Gamma, P, P \vdash \Delta}{\Gamma, P \vdash \Delta} +cL \qquad \frac{\Gamma \vdash \Delta}{\Gamma, P \vdash \Delta} +wL$$

Proof.

$$\frac{\frac{P \vdash !P}{\Gamma, P \vdash \Delta} \text{ cut} \quad \frac{\frac{\frac{\Gamma, P, P \vdash \Delta}{\Gamma, P, !P \vdash \Delta} !L}{\Gamma, !P, !P \vdash \Delta} !L}{\Gamma, !P \vdash \Delta} !cL}{\Gamma, P \vdash \Delta} \text{ cut} \qquad \frac{P \vdash !P \quad \frac{\Gamma \vdash \Delta}{\Gamma, !P \vdash \Delta} !wL}{\Gamma, P \vdash \Delta} \text{ cut}}$$

□

Positive formulas are also acceptable in the left-hand side context of the promotion rule. The following rule is derivable:

$$\frac{! \Gamma, P_1, \dots, P_n \vdash A, ? \Delta}{! \Gamma, P_1, \dots, P_n \vdash !A, ? \Delta} +!R$$

Proof.

$$\frac{\frac{\frac{\frac{\frac{\frac{! \Gamma, P_1, \dots, P_n \vdash A, ? \Delta}{! \Gamma, P_1, \dots, P_{n-1}, !P_n \vdash A, ? \Delta} !L}{\vdots}}{! \Gamma, P_1, !P_2, \dots, !P_n \vdash A, ? \Delta} !L}{! \Gamma, !P_1, \dots, !P_n \vdash A, ? \Delta} !L}{P_n \vdash !P_n \quad \frac{! \Gamma, !P_1, \dots, !P_n \vdash !A, ? \Delta}{! \Gamma, !P_1, \dots, !P_n \vdash !A, ? \Delta} !R} \text{ cut}}{! \Gamma, !P_1, \dots, !P_{n-1}, P_n \vdash !A, ? \Delta} \text{ cut}}{\vdots} \frac{P_1 \vdash !P_1 \quad ! \Gamma, !P_1, P_2, \dots, P_n \vdash !A, ? \Delta}{! \Gamma, P_1, \dots, P_n \vdash !A, ? \Delta} \text{ cut}}$$

□

1.17 Negative formula

A *negative formula* is a formula N such that $?N \multimap N$ (thus a **algebra** for the **monad** $?$). As a consequence N and $?N$ are **equivalent**.

A formula N is negative if and only if N^\perp is **positive**.

1.17.1 Negative connectives

A connective c of arity n is *negative* if for any negative formulas N_1, \dots, N_n , $c(N_1, \dots, N_n)$ is negative.

Proposition 4 (Negative connectives). $\wp, \perp, \&, \top, ?$ and \forall are negative connectives.

Proof. This is equivalent to the fact that $\otimes, 1, \oplus, 0, !$ and \exists are **positive connectives**. □

More generally, $?A$ is negative for any formula A .

The notion of negative connective is related with but different from the notion of **asynchronous connective**.

1.17.2 Generalized structural rules

Negative formulas admit generalized right structural rules corresponding to a structure of \mathfrak{N} -monoid: $N \mathfrak{N} N \multimap N$ and $\perp \multimap N$. The following rules are derivable:

$$\frac{\Gamma \vdash N, N, \Delta}{\Gamma \vdash N, \Delta} -cR \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash N, \Delta} -wR$$

Proof. This is equivalent to the [generalized left structural rules](#) for positive formulas. \square

Negative formulas are also acceptable in the context of the promotion rule. The following rule is derivable:

$$\frac{\vdash A, N_1, \dots, N_n}{\vdash !A, N_1, \dots, N_n} -!R$$

Proof. This is equivalent to the possibility of having positive formulas in the [left-hand side context of the promotion rule](#). \square

1.18 Regular formula

A *regular formula* is a formula R such that $R \circ\multimap ?!R$.

A formula L is *co-regular* if its dual L^\perp is regular, that is if $L \circ\multimap !?L$.

1.18.1 Alternative characterization

R is regular if and only if it is [equivalent](#) to a formula of the shape $?P$ for some [positive formula](#) P .

Proof. If R is regular then $R \circ\multimap ?!R$ with $!R$ positive. If $R \circ\multimap ?P$ with P positive then R is regular since $P \circ\multimap !P$. \square

1.18.2 Regular connectives

A connective c of arity n is *regular* if for any regular formulas R_1, \dots, R_n , $c(R_1, \dots, R_n)$ is regular.

Proposition 5 (Regular connectives). \mathfrak{N} , \perp and $?!$ define regular connectives.

Proof. If R and S are regular, $R \mathfrak{N} S \circ\multimap ?!R \mathfrak{N} ?!S \circ\multimap ?(!R \oplus !S)$ thus it is regular since $!R \oplus !S$ is positive.

$\perp \circ\multimap ?0$ thus it is regular since 0 is positive.

If R is regular then $?!R$ is regular, since $?!?!R \circ\multimap ?!R$. \square

More generally, $?!A$ is regular for any formula A .

Chapter 2

Intuitionistic linear logic

Intuitionistic Linear Logic (*ILL*) is the **intuitionistic restriction** of linear logic: the sequent calculus of *ILL* is obtained from the **two-sided sequent calculus of linear logic** by constraining sequents to have exactly one formula on the right-hand side: $\Gamma \vdash A$.

The connectives \wp , \perp and $?$ are not available anymore, but the linear implication \multimap is.

2.1 Sequent Calculus

$$\begin{array}{c}
\frac{}{A \vdash A} \text{ ax} \quad \frac{\Gamma \vdash A \quad \Delta, A \vdash C}{\Gamma, \Delta \vdash C} \text{ cut} \\
\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \otimes R \quad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \otimes L \quad \frac{}{\vdash 1} 1R \quad \frac{\Gamma \vdash C}{\Gamma, 1 \vdash C} 1L \\
\frac{\Gamma \vdash A \multimap B}{\Gamma \vdash A \multimap B} \multimap R \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C} \multimap L \\
\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} \& R \quad \frac{\Gamma, A \vdash C}{\Gamma, A \& B \vdash C} \&_1 L \quad \frac{\Gamma, B \vdash C}{\Gamma, A \& B \vdash C} \&_2 L \quad \frac{}{\Gamma \vdash \top} \top R \\
\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \oplus_1 R \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \oplus_2 R \quad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C} \oplus L \quad \frac{}{\Gamma, 0 \vdash C} 0L \\
\frac{\Gamma \vdash A}{\Gamma \vdash !A} !R \quad \frac{\Gamma, A \vdash C}{\Gamma, !A \vdash C} !dL \quad \frac{\Gamma, !A, !A \vdash C}{\Gamma, !A \vdash C} !cL \quad \frac{\Gamma \vdash C}{\Gamma, !A \vdash C} !wL \\
\frac{\Gamma \vdash A}{\Gamma \vdash \forall \xi A} \forall R \quad \frac{\Gamma, A[\tau/\xi] \vdash C}{\Gamma, \forall \xi A \vdash C} \forall L \quad \frac{\Gamma \vdash A[\tau/\xi]}{\Gamma \vdash \exists \xi A} \exists R \quad \frac{\Gamma, A \vdash C}{\Gamma, \exists \xi A \vdash C} \exists L
\end{array}$$

with ξ not free in Γ, C in the rules $\forall R$ and $\exists L$.

2.2 The intuitionistic fragment of linear logic

In order to characterize intuitionistic linear logic inside linear logic, we define the intuitionistic restriction of linear formulas:

$$J ::= X \mid J \otimes J \mid 1 \mid J \multimap J \mid J \& J \mid \top \mid J \oplus J \mid 0 \mid !J \mid \forall \xi J \mid \exists \xi J$$

JLL is the **fragment** of linear logic obtained by restriction to intuitionistic formulas.

Proposition 6 (From *ILL* to *JLL*). *If $\Gamma \vdash A$ is provable in ILL_{012} , it is provable in JLL_{012} .*

Proof. *ILL*₀₁₂ is included in *JLL*₀₁₂. □

Theorem 8 (From *JLL* to *ILL*). *If $\Gamma \vdash \Delta$ is provable in JLL_{12} , it is provable in ILL_{12} .*

Proof. We only prove the first order case, a proof of the full result is given in the PhD thesis of Harold Schellinx [Sch94].

Chapter 3

Polarized linear logic

Polarized linear logic (LLP) is a logic close to plain linear logic in which structural rules, usually restricted to $?$ -formulae, have been [extended](#) to the whole class of so called *negative* formulae.

3.1 Polarization

LLP relies on the notion of *polarization*, that is, it discriminates between two types of formulae, *negative* (noted $M, N\dots$) vs. *positive* ($P, Q\dots$). They are mutually defined as follows:

$$M, N ::= X \mid M \wp N \mid \perp \mid M \& N \mid \top \mid ?P$$

$$P, Q ::= X^\perp \mid P \otimes Q \mid 1 \mid P \oplus Q \mid 0 \mid !N$$

The dual operation $(-)^{\perp}$ extended to propositions exchanges the roles of connectors and reverses the polarity of formulae.

3.2 Deduction rules

They are several design choices for the structure of sequents. In particular, LLP proofs are *focused*, i.e. they contain at most one positive formula. We choose to represent this explicitly using sequents of the form $\vdash \Gamma \mid \Delta$, where Γ is a multiset of negative formulae, and Δ is a *stoup* that contains at most one positive formula (though it may be empty).

$$\frac{}{\vdash P^\perp \mid P} \text{ax} \quad \frac{\vdash \Gamma_1, N \mid \Delta \quad \vdash \Gamma_2 \mid N^\perp}{\vdash \Gamma_1, \Gamma_2 \mid \Delta} \text{cut}$$

$$\frac{\vdash \Gamma, N \mid \cdot}{\vdash \Gamma \mid !N} p \quad \frac{\vdash \Gamma \mid P}{\vdash \Gamma, ?P \mid \cdot} d \quad \frac{\vdash \Gamma, N, N \mid \Delta}{\vdash \Gamma, N \mid \Delta} c \quad \frac{\vdash \Gamma \mid \Delta}{\vdash \Gamma, N \mid \Delta} w$$

$$\frac{\vdash \Gamma_1 \mid P \quad \vdash \Gamma_2 \mid Q}{\vdash \Gamma_1, \Gamma_2 \mid P \otimes Q} \otimes \quad \frac{}{\vdash \cdot \mid 1} 1 \quad \frac{\vdash \Gamma, M, N \mid \Delta}{\vdash \Gamma, M \wp N \mid \Delta} \wp \quad \frac{\vdash \Gamma \mid \Delta}{\vdash \Gamma, \perp \mid \Delta} \perp$$

$$\frac{\vdash \Gamma \mid P}{\vdash \Gamma \mid P \oplus Q} \oplus_1 \quad \frac{\vdash \Gamma \mid Q}{\vdash \Gamma \mid P \oplus Q} \oplus_2 \quad \frac{\vdash \Gamma, M \mid \Delta \quad \vdash \Gamma, N \mid \Delta}{\vdash \Gamma, M \& N \mid \Delta} \& \quad \frac{}{\vdash \Gamma, \top \mid \Delta} \top$$

Chapter 4

Fragment

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. In linear logic, the most well known fragments are obtained by combining/removing in different ways the classes of connectives present in the [language of linear logic](#) itself:

- **Multiplicative connectives:** the conjunction \otimes (*tensor*) and the disjunction \wp (*par*), with their respective units 1 (*one*) and \perp (*bottom*); these connectives are the combinatorial base of linear logic (permutations, circuits, etc.).
- **Additive connectives:** the conjunction $\&$ (*with*) and the disjunction \oplus (*plus*), with their respective units \top (*top*) and 0 (*zero*); the computational content of these connectives, which behave more closely to their intuitionistic counterparts (*e.g.*, $A \& B \multimap A$ and $A \& B \multimap B$ are provable), is strongly related to choice (*if...then...else*, product and sum types, etc.).
- **Exponential connectives:** the modalities $!$ (*of course*) and $?$ (*why not*) handle the structural rules in linear logic, and are necessary to recover the expressive power of intuitionistic or classical logic.
- **Quantifiers:** just as in classical logic, quantifiers may be added to propositional linear logic, at any order. The most frequently considered are the second order ones (in analogy with System F).

The additive and exponential connectives, if taken alone, yield fragments of limited interest, so one usually considers only fragments containing at least the multiplicative connectives (perhaps without units). It is important to observe that the [cut elimination rules](#) of linear logic do not introduce connectives belonging to a different class than that of the pair of dual formulas whose cut is being reduced. Hence, any fragment defined by combining the above classes will enjoy cut elimination. Since cut elimination implies the subformula property, all of the [fundamental equivalences](#) provable in full linear logic remain valid within such fragments, as soon as the formulas concerned belong to the fragment itself.

Conventionally, if LL denotes full linear logic, its fragments 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 may specify whether units and/or quantifiers are present or not, and, for quantifiers, of what order (see chapter [A](#)).

4.1 Motivations

The main interest of studying fragments of linear logic is that these are usually simpler than the whole system, so that certain properties may be first analyzed on fragments, and then extended or adapted to increasingly larger fragments. It may also be interesting to see, given a property that does not hold for full linear logic, whether it holds for a fragment, and where the "breaking point" is situated. Examples of such questions include:

- **logical complexity:** proving cut elimination for full linear logic with second order quantification is equivalent to proving the consistency of second order Peano arithmetic (Girard, via [translations of System F](#) in linear logic). One may expect that smaller fragments have lower logical complexity.
- **provability:** the *provability problem* for a logical system S is defined as follows: given a formula A in the language of S , is A provable in S ? This problem is undecidable in full linear logic with quantifiers, of whatever order (again, because [classical logic can be translated](#) in linear logic). On what fragments does it become decidable? And if it does, what is its computational complexity?
- **computational complexity of cut elimination:** the *cut elimination problem* (Mairson-Terui) for a logical system S is defined as follows: given two proofs of A in S , do they reduce to the same cut-free proof? Although decidable (thanks to strong normalization), this problem is not elementary recursive in full propositional linear logic (Statman, again via the above-mentioned translations). Does the problem fall into any interesting complexity class when applied to fragments?
- **proof nets:** the definition of proof nets, and in particular the formulation of correctness criteria and the study of their complexity, is a good example of how a methodology can be applied to a small fragment of linear logic and later adapted (more or less successfully) to wider fragments.
- **denotational semantics:** several problems related to denotational semantics (formulation of [categorical models](#), full abstraction, full completeness, injectivity, etc.) may be first attacked in the simpler case of fragments, and then extended to wider subsystems.

4.2 Multiplicative fragments

Multiplicative linear logic (MLL) is the simplest of the well known fragments of linear logic. Its formulas are obtained by combining propositional atoms with the connectives *tensor* and *par* only. As a consequence, the [sequent calculus](#) of MLL is limited to the rules axiom, cut, \otimes , and \wp . These rules actually determine the multiplicative connectives: if a dual pair of connectives \otimes' and \wp' is introduced, with the same rules as \otimes and \wp , respectively, then one can show $A \otimes' B$ to be provably equivalent to $A \otimes B$ (and, dually, $A \wp' B$ to be provably equivalent to $A \wp B$).

The cut elimination problem for MLL is **P**-complete (Mairson-Terui), even though there exists a deterministic algorithm solving the problem in logarithmic space if one considers only [eta-expanded](#) proofs (Mairson-Terui). On the other hand, provability for MLL is **NP**-complete, and it remains so even in presence of first order quantifiers.

Another multiplicative fragment, less considered in the literature, can be defined by using the units 1 and \perp instead of the propositional atoms. In this fragment, denoted by MLL_u , one can also eliminate the axiom rule from sequent calculus, since it is redundant. MLL_u is even

simpler than MLL : its provability problem is in \mathbf{P} , and, since all proofs are eta-expandend, its cut elimination problem is in \mathbf{L} .

The union of MLL and MLL_u is the full propositional multiplicative fragment of linear logic, and is denoted by MLL_0 . It has the same properties as MLL , which shows that the presence/absence of propositional atoms (and of the axiom rule) has a non-trivial effect on the complexity of provability and cut elimination, *i.e.*, the complexity is not altered iff $\mathbf{P} \subseteq \mathbf{NP}$ and $\mathbf{L} \subseteq \mathbf{P}$, respectively.

If we add second order quantifiers to MLL (resp. MLL_u), we obtain a system denoted by MLL_2 (resp. MLL_{02}). In MLL_{02} one can show that 1 and \perp are provably equivalent to $\forall X.(X^\perp \wp X)$ and $\exists X.(X^\perp \otimes X)$, respectively. Hence, MLL_2 is as expressive as MLL_{02} . In these second order fragments, provability is undecidable, while cut elimination is still \mathbf{P} -complete.

4.3 Additive fragments

The most studied additive fragments of linear logic are defined by taking MLL or MLL_0 and by enriching their language with the additive connectives, with or without units. The same can be done in presence of quantifiers. We thus obtain:

- $MALL$: formulas built from propositional atoms using $\otimes, \wp, \&, \oplus$;
- $MALL_0$: formulas built from propositional atoms and $1, \perp, \top, 0$, using $\otimes, \wp, \&, \oplus$;
- $MALL_n$: $MALL$ with quantifiers of order n ;
- $MALL_{0n}$: $MALL_0$ with quantifiers of order n .

The **purely additive fragments** are less common:

- ALL : formulas built from propositional atoms using $\&, \oplus$;
- ALL_0 : formulas built from propositional atoms and $\top, 0$, using $\&, \oplus$;
- ALL_n : ALL with quantifiers of order n ;
- ALL_{0n} : ALL_0 with quantifiers of order n .

As for the multiplicative connectives, the additive connectives are also defined by their rules: adding a pair of dual connectives $\&', \oplus'$ to $MALL$, and giving them the same rules as $\&, \oplus$, makes the new connectives provably equivalent to the old ones.

In $MALL_{02}$, the additive units \top and 0 are provably equivalent to $\exists X.X^\perp$ and $\forall X.X$, respectively. Since multiplicative units are also definable in terms of second order quantification, we obtain that $MALL_2$ is as expressive as $MALL_{02}$.

The cut elimination problem is **coNP**-complete for all of the fragments defined above (Mairson-Terui).

Provability is undecidable in any additive fragment as soon as second order quantification is considered. It is decidable, although quite complex, in the propositional and first order case: it is **PSPACE**-complete in $MALL_0$, and **NEXP**-complete in $MALL_{01}$. This latter result is indicative of the fact that the undecidability of predicate calculus is not ascribable to existential quantification alone, but rather to the simultaneous presence of existential quantification and contraction.

4.4 Exponential fragments

The most common proper fragments of linear logic containing the exponential connectives are defined as in the case of the additive fragments, *i.e.*, by adding the modalities on top of MLL and its variants:

- $MELL$: formulas built from propositional atoms using $\otimes, \wp, !, ?$;
- $MELL_0$: formulas built from propositional atoms and $1, \perp$, using $\otimes, \wp, !, ?$;
- $MELL_n$: $MELL$ with quantifiers of order n ;
- $MELL_{0n}$: $MELL_0$ with quantifiers of order n .

If, instead of taking MLL , we add the modalities to $MALL$, we obtain of course various versions of full linear logic:

- LL : full linear logic, without units;
- LL_0 : full linear logic, with units;
- LL_n : LL with quantifiers of order n ;
- LL_{0n} : LL_0 with quantifiers of order n .

In LL_{02} the formulas $A \& B$ and $A \oplus B$ are provably equivalent to $\exists X. (! (X^\perp \wp A) \otimes ! (X^\perp \wp B) \otimes X)$ and $\forall X. (? (X^\perp \otimes A) \wp ? (X^\perp \otimes B) \wp X)$, respectively, for all A, B . Thanks to the second-order definability of units discussed above, we obtain that $MELL_2$ is as expressive as LL_{02} , *i.e.*, full propositional second order linear logic embeds in its second order multiplicative exponential fragment without units.

Girard showed how cut elimination for LL_{02} *without the contraction rule* can be proved by a simple induction up to ω , *i.e.*, in first order Peano arithmetic. This gives a huge gap between the logical complexity of full linear logic and its contraction-free subsystem: in fact, still by Girard's results, we know that cut elimination in $MELL_2$ is equivalent to the consistency of second order Peano arithmetic, for which no ordinal analysis is known. There are nevertheless subsystems of $MELL_2$, the so-called [light subsystems](#) of linear logic, in which the exponential connectives are weakened, whose cut elimination can be proved in second order Peano arithmetic even in presence of contraction.

The cut elimination problem is never elementary recursive in presence of exponential connectives: the simply typed λ -calculus with arrow types only can be encoded in $MELL$, and this is enough for Statman's lower bound to apply. However, it becomes elementary recursive in the above mentioned [light logics](#).

Albeit perhaps surprisingly, provability in LL is already undecidable. This result, obtained by coding Minsky machines with linear logic formulas, contrasts with the situation in classical logic, whose propositional fragment is notoriously decidable. It is indicative of the fact that modalities are themselves a form of quantification, although this claim is far from being clear: as a matter of fact, the decidability of propositional provability in the absence of additives, *i.e.*, in $MELL$ alone, is still an open problem. It is known that adding first order quantification to $MELL$ makes it undecidable.

4.4.1 About exponential rules

In this section, provability is assumed to be in LL_{02} , *i.e.*, full propositional second order linear logic.

In contrast with multiplicative and additive connectives, the modalities of linear logic are not defined by their rules: one may introduce a pair of dual modalities $!, ?$, each with the same rules as $!, ?$, without $!A$ (resp. $?A$) being in general provably equivalent to $!A$ (resp. $?A$).

The [promotion rule](#) is derivable from the following two rules, called *functorial promotion* and *digging*, respectively:

$$\frac{\vdash \Gamma, A}{\vdash ?\Gamma, !A} f! \qquad \frac{\vdash \Gamma, ??A}{\vdash \Gamma, ?A} dig$$

Functorial promotion is itself derivable from dereliction and promotion; the digging rule is also derivable, but only using the cut rule (in fact, digging does not enjoy the subformula property). It may be convenient to consider this pair of rules instead of the standard promotion rule in the context of [categorical semantics](#) of linear logic.

In presence of the digging rule, dereliction, weakening, and contraction can be derived from the following rule, called *multiplexing*, in which $A^{(n)}$ stands for the sequence A, \dots, A containing n occurrences of A :

$$\frac{\vdash \Gamma, A^{(n)}}{\vdash \Gamma, ?A} mux$$

Of course, multiplexing is itself derivable from dereliction, weakening, and contraction. Hence, there are several alternative but equivalent presentations of the exponential fragment of linear logic, such as

1. remove promotion, and replace it with functorial promotion and digging;
2. remove promotion, dereliction, weakening, and contraction, and replace them with functorial promotion, digging, and multiplexing.

Apart from their usefulness in [categorical semantics](#), these alternative formulations are of interest in the context of the so-called [light linear logics](#) mentioned above. For example, *elementary linear logic* is obtained by removing dereliction and digging from formulation 1, and *soft linear logic* is obtained by removing digging from formulation 2.

Multiplexing is invertible in certain circumstances. A sequent $\vdash \Gamma, ?A$ containing no occurrence of $\&$, $!$, or second order \exists is provable iff $\vdash \Gamma, A^{(n)}$ is provable for some n (this is easily checked by induction on cut-free proofs). To see that this does not hold in general, take for instance $A = X^\perp$ and $\Gamma = X \& 1$, or $\Gamma = !X$. The restriction on the presence of additive conjunction can be removed by slightly changing the statement: a sequent $\vdash \Gamma, ?A$ containing no occurrence of $!$ or second order \exists is provable iff $\vdash \Gamma, (A \oplus \perp)^{(n)}$ is provable for some n .

The latter result can be generalized as follows. If A is a formula, $!_n A$ stands for the formula $(A \& 1) \otimes \dots \otimes (A \& 1)$ (n times) and $?_n A$ for the formula $(A \oplus \perp) \wp \dots \wp (A \oplus \perp)$ (n times). Then, we have

Theorem 9 (Approximation Theorem). *Let $\vdash \Gamma$ be a provable sequent containing p occurrences of $!$, q occurrences of $?$, and no occurrence of second order \exists . Then, for all $m_1, \dots, m_p \in \mathbb{N}$, there are $n_1, \dots, n_q \in \mathbb{N}$ such that the sequent obtained from $\vdash \Gamma$ by replacing the p occurrences of $!$ with $!_{m_1}, \dots, !_{m_p}$ and the q occurrences of $?$ with $?_{n_1}, \dots, ?_{n_q}$ is provable.*

A *structural formula* is a formula C such that $C \multimap C \otimes C$ and $C \multimap 1$ are provable. Obviously, any formula of the form $?B$ is structural. However, the promotion rule cannot be extended to arbitrary structural formulas, *i.e.*, the following rule is *not* admissible:

$$\frac{C \vdash A \quad C \vdash C \otimes C \quad C \vdash 1}{C \vdash !A}$$

For instance, if $A = C = \alpha \otimes !(\alpha \multimap \alpha \otimes \alpha) \otimes !(\alpha \multimap 1)$, the three premises are provable but not the conclusion.

The following rule, called *absorption*, is derivable in the standard sequent calculus:

$$\frac{\vdash \Gamma, ?A, A}{\vdash \Gamma, ?A}$$

The absorption rule is useful in the context of proof search in linear logic.

4.5 The provability problem

It is well known that the decidability of the provability problem is connected to the [finite model property](#): if a fragment of a logic with a truth semantics enjoys the finite model property, then the provability in that fragment is decidable. Note of course that the converse may fail.

In this section, we summarize the known results about the validity of the finite model property and the decidability of provability, with its complexity, for the various fragments of linear logic introduced above. Question marks in the tables below denote open problems. For brevity, all fragments are assumed to have units and propositional atoms, *e.g.*, *MLL* actually denotes what we called *MLL*₀ above.

4.5.1 The finite model property

<i>MLL</i>	<i>MALL</i>	<i>MELL</i>	<i>LL</i>
yes	yes	no	no

4.5.2 Provability

	<i>MLL</i>	<i>MALL</i>	<i>MELL</i>	<i>LL</i>
propositional case	NP -complete	PSPACE -complete	?	undecidable
first order case	NP -complete	NEXP -complete	undecidable	undecidable
second order case	undecidable	undecidable	undecidable	undecidable

4.6 The cut elimination problem

In this section, we summarize the known results about the complexity of the cut elimination problem for the various fragments of linear logic introduced above, plus some [light linear logics](#). All fragments are assumed to be propositional; the results do not change in presence of quantification of any order.

<i>MLL</i> _u	<i>MLL</i>	<i>MALL</i>	<i>MSLL</i>	<i>MLLL</i>	<i>MELL</i>
L	P -complete	coNP -complete	EXP -complete	2EXP -complete	not elementary recursive

Notations used in the above table:

- *MSLL*: multiplicative soft linear logic;
- *MLLL*: multiplicative light linear logic.

Chapter 5

Proof-nets: a formal account of nets

We provide a formal account of nets, but it is probably not the best way to learn about proof-nets if you have never seen them before.

The aim of this page is to provide a common framework for describing linear logic proof nets, interaction nets, multiport interaction nets, and the likes, while factoring out most of the tedious, uninteresting details (clearly not the fanciest page of LLWiki).

5.1 Preliminaries

5.1.1 The short story

- the general flavor is that of multiport interaction nets;
- the top/down or passive/active orientation of cells is related with the distinction between premisses and conclusions of rules, (and in that sense, a cut is not a logical rule, but the focus of interaction between two rules);
- cuts are thus wires rather than cells/links: this fits with the intuition of GoI, but not with the most common presentations of proof nets;
- because the notion of subnet is not trivial in multiport interaction nets, and to avoid the use of geometric conditions (boxes must not overlap but can be nested), we introduce boxes as particular cells;
- when representing proof nets, we introduce axioms explicitly as cells, so that axiom-cuts do not vanish.

5.2 Nets

5.2.1 Wires

A *wiring* is the data of a finite set P of ports and of a partition W of P by pairs (the *wires*): if $\{p, q\} \in W$, we write $W(p) = q$ and $W(q) = p$. Hence a wiring is equivalently given by an involutive permutation W of finite domain P , without fixpoints (for all p , $W(p) \neq p$): the wires are then the orbits. Another equivalent presentation is to consider W as a (simple, loopless, undirected) graph, with vertices in P , all of degree 1.

We say two wirings are disjoint when their sets of ports are. A *connection* between two disjoint wirings W and W' is a partial injection $(I, I', f) : P \hookrightarrow P'$: $I \subseteq P$, $I' \subseteq P'$ and f is a bijection $I \cong I'$. We then write $W \bowtie_f W'$ for the wiring obtained by identifying the ports

pairwise mapped by f , and then “straightening” the paths thus obtained to recover wires: notice this might also introduce loops and we write $\langle\langle W|W' \rangle\rangle_f$ for the number of loops thus appeared.

We describe these operations a bit more formally. Write $P = P_0 \uplus I$ and $P' = P'_0 \uplus I'$. Then consider the graph $W ::_f W'$ with vertices in $P \cup P'$, and such that there is an edge between p and q iff $q = W(p)$ or $q = W'(p)$ or $q = f(p)$ or $p = f(q)$: in other words, $W ::_f W' = W \cup W' \cup f \cup f^{-1}$. Vertices in $P_0 \cup P'_0$ are of degree 1, and the others are of degree 2. Hence maximal paths in $W ::_f W'$ are of two kinds:

- straight paths, with both ends in $P_0 \cup P'_0$;
- cycles, with vertices all in $I \cup I'$.

Then the wires in $W \bowtie_f W'$ are the pairs $\{p, p'\}$ such that p and p' are the ends of a path in $W ::_f W'$. And $\langle\langle W|W' \rangle\rangle_f$ is the number of cycles in $W ::_f W'$, or more precisely the number of support sets of cycles (i.e. we forget about the starting vertice of cycles).

Lemma 7. *Consider three wirings (P, W) , (P', W') and (P'', W'') , and two connections $(I, I', f) : P \hookrightarrow P'$ and $(J', J'', g) : P' \hookrightarrow P''$ such that $I' \cap J' = \emptyset$. Then*

$$(W \bowtie_f W') \bowtie_g W'' = W \bowtie_f (W' \bowtie_g W'')$$

and

$$\langle\langle W|W' \rangle\rangle_f + \langle\langle (W \bowtie_f W')|W'' \rangle\rangle_g = \langle\langle W|(W' \bowtie_g W'') \rangle\rangle_f + \langle\langle W'|W'' \rangle\rangle_g.$$

Proof. The first equation holds because open maximal paths in $W ::_f W' \bowtie_g W''$ correspond with those in $W ::_f (W' ::_g W'')$, hence in $(W ::_f W') ::_g W''$, hence in $W \bowtie_f W' ::_g W''$. The second equation holds because both sides are two possible writings for the number of loops in $(W ::_f W') ::_g W''$. \square

5.2.2 Nets

A *signature* is the data of a set Σ of symbols, together with arity functions $\alpha : \Sigma \rightarrow \mathbf{N} \setminus \{0\}$ (number of *active* ports, or conclusions) and $\pi : \Sigma \rightarrow \mathbf{N}$ (number of *passive* ports, or hypotheses). In the remaining, we assume such a signature is given.

A *cell* c with ports in P is the data of a symbol $\sigma \in \Sigma$ and of two disjoint lists of pairwise distinct ports: $\alpha(c) \in P^{\alpha(\sigma)}$ is the list of *active* ports and $\pi(c) \in P^{\pi(\sigma)}$ is the list of *passive* ports.

A *net* is the data of a wiring (P, W) , of a set C of disjoint cells on P , and of a number $L \in \mathbf{N}$ (the number of *loops*). It follows from the definitions that each port $p \in P$ appears in exactly one wire and in at most one cell: we say p is *free* if it is not part of a cell, and p is *internal* otherwise. Moreover, we say p is *dangling* if p is free and $W(p)$ is internal. We write $\text{fp}(R)$ for the set of free ports of a net R .

Generally, the “names” of internal ports of a net are not relevant, but free ports matter most often: internal ports are the analogue of bound variables in λ -terms. More formally, an *isomorphism* from net R to net R' is the data of a bijection of ports $\phi : P \cong P'$ and a bijection of cells $\psi : C \cong C'$ such that:

- for all $p \in P$, $W'(\phi(p)) = \phi(W(p))$;
- for all $c \in C$:
 - $\sigma_{\psi(c)} = \sigma_c$,
 - for all $i \in \{0, \dots, \alpha(\sigma_c) - 1\}$, $\alpha(\psi(c))_i = \phi(\alpha(c)_i)$,
 - for all $j \in \{0, \dots, \pi(\sigma_c) - 1\}$, $\pi(\psi(c))_j = \phi(\pi(c)_j)$;

- $L = L'$.

Observe that under these conditions ψ is uniquely induced by ϕ . We say that the isomorphism ϕ is *nominal* if moreover $p \in \text{fp}(R)$ implies $p = \phi(p)$.

5.2.3 Subnets

We say two nets are disjoint when their sets of ports are. Let R and R' be disjoint nets, and (I, I', f) be a connection between W and W' , such that $I \subseteq \text{fp}(R)$ and $I' \subseteq \text{fp}(R')$. We then write $R \bowtie_f R'$ for the net with wiring $W \bowtie_f W'$, cells $C \cup C'$ and loops $\langle\langle R|R' \rangle\rangle_f = L + L' + \langle\langle W|W' \rangle\rangle_f$. We say this connection is *orthogonal* if $\langle\langle W|W' \rangle\rangle_f = 0$, and it is *modular* if the ports in $I \cup I'$ are all dangling: a modular connection is always orthogonal.

We say R_0 is a *subnet* of R , if there exists a net R' and a connection (I, I', f) between R_0 and R' such that $R = R_0 \bowtie_f R'$.

Lemma 8. *Let R_0, R_1 and R' be nets such that R_0 and R_1 are disjoint from R , (I, I', f) be a connection $\text{fp}(R_0) \hookrightarrow \text{fp}(R')$ and ϕ an isomorphism $R_0 \cong R_1$ such that $\phi(p) = \phi(p')$ for all $p \in \text{fp}(R_0) \setminus I$. Then $R_0 \bowtie_f R'$ is nominally isomorphic to $R_1 \bowtie_{f \circ \phi^{-1}} R'$.*

5.2.4 Rewriting

A *net rewriting rule* is a pair (r_0, r_1) of nets such that $\text{fp}(r_0) = \text{fp}(r_1)$. Then an instance of this rule is a pair (R_0, R_1) such that there exist:

- nets R'_0 and R'_1 isomorphic to r_0 and r_1 respectively, namely $R'_i = \phi_i(r_i)$, so that moreover $\phi_0(p) = \phi_1(p)$ for all $p \in \text{fp}(r_0)$ (in particular, $\text{fp}(R'_0) = \text{fp}(R'_1)$);
- a net R'' , disjoint from R'_0 and R'_1 ;
- a connection $(\text{fp}(R'_0), J, f) : \text{fp}(R'_0) \hookrightarrow \text{fp}(R'')$;

such that each R_i is nominally isomorphic to $R'_i \bowtie_f R''$.

5.2.5 Typing

A typing system on signature Σ is the data of a set Θ of types, an involutive negation $\cdot^\perp : \Theta \rightarrow \Theta$, together with a typing discipline for each symbol, *i.e.* a relation $\Theta(\sigma) \subseteq \Theta^{\alpha(\sigma)} \times \Theta^{\pi(\sigma)}$. We then write $A_1, \dots, A_{\pi(\sigma)} \vdash_\sigma B_1, \dots, B_{\alpha(\sigma)}$ for $(\vec{A}, \vec{B}) \in \Theta(\sigma)$.

Then a *typing* for net $R = (P, W, C)$ is a function $\tau : P \rightarrow \Theta$ such that:

- for all $p \in P$, $\tau(W(p)) = \tau(p)^\perp$;
- for all $c \in C$ of symbol σ , $\tau(\pi(c)) \vdash \tau(\alpha(c))^\perp$;

where, in the last formula, we implicitly generalized τ to lists of ports and \cdot^\perp to lists of formulas, in the obvious, componentwise fashion. The *interface* of the typed net (R, τ) is then the restriction of τ to $\text{fp}(R)$.

The idea, is that a wire $\{p, q\}$ bears the type $\tau(q)$ (resp. $\tau(p)$) in the direction (p, q) (resp. (q, p)), so that a rule $\vec{A} \vdash_\sigma \vec{B}$ reads as an inference from passive inputs (hypotheses) to active outputs (conclusions).

Observe that if (R, τ) is a typed net, and $\phi : R \cong R'$ is an isomorphism, then R' is typed and its interface is $\tau \circ \phi^{-1}$: in particular, if ϕ is nominal, R' and R have the same interface.

Now let (R, τ) and (R', τ') be typed nets and (I, I', f) a connection so that $\tau' \circ f$ and $\tau^\perp = \cdot^\perp \circ \tau$ coincide on I . Then this induces a typing of $R \bowtie_f R'$ preserving the interface on the remaining free ports.

5.2.6 Boxes

A signature with boxes is the data of a signature Σ , together with a *box arity* $\beta(\sigma)$ for all symbol $\sigma \in \Sigma$.

Then a net on signature with boxes (Σ, β) is the same as a net R on Σ plus, for each cell c of R , the data $\beta(c)$ of a $\beta(\sigma_c)$ -tuple of nets (with boxes) with free ports the internal ports of c .

Chapter 6

System L

System L [CMM10] is a family of syntax for a variety of variants of linear logic, inspired from classical calculi such as $\bar{\lambda}\mu\tilde{\mu}$ -calculus. These, in turn, stem from the study of abstract machines for λ -calculus. In this realm, **polarization** and **focusing** are features that appear naturally. Positives are typically values, and negatives pattern-matches. Contraction and weakening are implicit.

We present here a system for explicitly polarized and focused linear logic. Polarization classifies terms and types between positive and negative; focusing separates values from non-values.

6.1 Definitions

Positive types: $P ::= 1 \mid P_1 \otimes P_2 \mid 0 \mid P_1 \oplus P_2 \mid \downarrow N \mid !N$

Negative types: $N ::= \perp \mid N_1 \wp N_2 \mid \top \mid N_1 \& N_2 \mid \uparrow P \mid ?P$

Positive values: $v^+ ::= x^+ \mid () \mid (v_1^+, v_2^+) \mid \text{inl}(v^+) \mid \text{inr}(v^+) \mid \downarrow t^- \mid \mu(?x^+).c$

Positive terms: $t^+ ::= v^+ \mid \mu x^-.c$

Negative terms: $t^- ::= x^- \mid \mu x^+.c \mid \mu().c \mid \mu(x^+, y^+).c \mid \mu[\cdot] \mid \mu[\text{inl}(x^+).c_1 \mid \text{inr}(y^+).c_2] \mid \mu(\downarrow x^-).c \mid ?v^+$

Commands: $c ::= \langle t^+ \mid t^- \rangle$

6.2 Typing

There are as many typing sequents classes as there are terms classes. Typing of positive values corresponds to focused sequents, and commands are cuts.

Positive values: sequents are of the form $\vdash \Gamma :: v^+ : P$.

$$\begin{array}{c}
 \frac{}{\vdash x^+ : P^\perp :: x^+ : P} \text{ax}^+ \\
 \\
 \frac{}{\vdash :: () : 1} 1 \qquad \frac{\vdash \Gamma_1 :: v_1^+ : P_1 \quad \vdash \Gamma_2 :: v_2^+ : P_2}{\vdash \Gamma_1, \Gamma_2 :: (v_1^+, v_2^+) : P_1 \otimes P_2} \otimes \\
 \\
 \frac{\vdash \Gamma :: v^+ : P_1}{\vdash \Gamma :: \text{inl}(v^+) : P_1 \oplus P_2} \oplus_1 \qquad \frac{\vdash \Gamma :: v^+ : P_2}{\vdash \Gamma :: \text{inr}(v^+) : P_1 \oplus P_2} \oplus_2 \\
 \\
 \frac{\vdash \Gamma \mid t^- : N}{\vdash \Gamma :: \downarrow t^- : \downarrow N} \downarrow \qquad \frac{c \vdash ?\Gamma, x^+ : N}{\vdash ?\Gamma :: \mu(?x^+).c : !N} !
 \end{array}$$

Positive terms: sequents are of the form $\vdash \Gamma \mid t^+ : P$.

$$\frac{\vdash \Gamma :: v^+ : P}{\vdash \Gamma \mid v^+ : P} \text{ foc} \qquad \frac{c \vdash \Gamma, x^- : P}{\vdash \Gamma \mid \mu x^- . c : P} \mu^-$$

Negative terms: sequents are of the form $\vdash \Gamma \mid t^- : N$.

$$\frac{}{\vdash x^- : N^\perp \mid x^- : N} \text{ ax}^- \qquad \frac{c \vdash \Gamma, x^+ : N}{\vdash \Gamma \mid \mu x^+ . c : N} \mu^+$$

$$\frac{c \vdash \Gamma}{\vdash \Gamma \mid \mu().c : \perp} \perp \qquad \frac{c \vdash \Gamma, x^+ : N_1, y^+ : N_2}{\vdash \Gamma \mid \mu(x^+, y^+) . c : N_1 \wp N_2} \wp$$

$$\frac{}{\vdash \Gamma \mid \mu[\cdot] : \top} \top \qquad \frac{c_1 \vdash \Gamma, x^+ : N_1 \quad c_2 \vdash \Gamma, y^+ : N_2}{\vdash \Gamma \mid \mu[inl(x^+) . c_1 \mid inr(y^+) . c_2] : N_1 \& N_2} \&$$

$$\frac{c \vdash \Gamma, x^- : P}{\vdash \Gamma \mid \mu(\downarrow x^-) . c : \uparrow P} \uparrow \qquad \frac{\vdash \Gamma :: v^+ : P}{\vdash \Gamma \mid ?v^+ : ?P} ?$$

Commands:

$$\frac{\vdash \Gamma \mid t^+ : P \quad \vdash \Delta \mid t^- : P^\perp}{\langle t^+ \mid t^- \rangle \vdash \Gamma, \Delta} \text{ cut}$$

$$\frac{c \vdash \Gamma}{c \vdash \Gamma, x^+ : ?P} \text{ wkn} \qquad \frac{c \vdash \Gamma, x_1^+ : ?P, x_2^+ : ?P}{c[x_1^+ := x^+, x_2^+ := x^+] \vdash \Gamma, x^+ : ?P} \text{ ctr}$$

6.3 Reduction rules

$$\begin{aligned} \langle v^+ \mid \mu x^+ . c \rangle &\rightarrow c[x^+ := v^+] \\ \langle \mu x^- . c \mid t^- \rangle &\rightarrow c[x^- := t^-] \\ \langle () \mid \mu().c \rangle &\rightarrow c \\ \langle (v_1^+, v_2^+) \mid \mu(x^+, y^+) . c \rangle &\rightarrow c[x^+ := v_1^+, y^+ := v_2^+] \\ \langle inl(v^+) \mid \mu[inl(x^+) . c_1 \mid inr(y^+) . c_2] \rangle &\rightarrow c_1[x^+ := v^+] \\ \langle inr(v^+) \mid \mu[inl(x^+) . c_1 \mid inr(y^+) . c_2] \rangle &\rightarrow c_2[y^+ := v^+] \\ \langle \downarrow t^- \mid \mu(\downarrow x^-) . c \rangle &\rightarrow c[x^- := t^-] \\ \langle \mu(?x^+) . c \mid ?v^+ \rangle &\rightarrow c[x^+ := v^+] \end{aligned}$$

Chapter 7

Translations of intuitionistic logic

The genesis of linear logic comes with a decomposition of the intuitionistic implication. Once linear logic properly defined, it corresponds to a translation of intuitionistic logic into linear logic, often called *Girard's translation*. In fact Jean-Yves Girard has defined two translations in his linear logic paper [Gir87]. We call them the **call-by-name** translation and the **call-by-value** translation.

These translations can be extended to [translations of classical logic](#) into linear logic.

7.1 Call-by-name Girard's translation $A \rightarrow B \mapsto !A \multimap B$

Formulas are translated as:

$$\begin{aligned}
 X^n &= X \\
 (A \rightarrow B)^n &= !A^n \multimap B^n \\
 (A \wedge B)^n &= A^n \& B^n \\
 T^n &= \top \\
 (A \vee B)^n &= !A^n \oplus !B^n \\
 F^n &= 0 \\
 (\forall \xi A)^n &= \forall \xi A^n \\
 (\exists \xi A)^n &= \exists \xi !A^n
 \end{aligned}$$

This is extended to sequents by $(\Gamma \vdash A)^n = !\Gamma^n \vdash A^n$.

This allows one to translate the rules of intuitionistic logic into linear logic:

$$\begin{array}{l}
 \frac{}{A \vdash A} \text{ax} \quad \mapsto \quad \frac{A^n \vdash A^n}{!A^n \vdash A^n} \text{ax} \\
 \\
 \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \text{cut} \quad \mapsto \quad \frac{\frac{!\Gamma^n \vdash A^n}{!\Gamma^n \vdash !A^n} !R \quad !\Delta^n, !A^n \vdash B^n}{!\Gamma^n, !\Delta^n \vdash B^n} \text{cut} \\
 \\
 \frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C} cL \quad \mapsto \quad \frac{!\Gamma^n, !A^n, !A^n \vdash C^n}{!\Gamma^n, !A^n \vdash C^n} !cL \\
 \\
 \frac{\Gamma \vdash C}{\Gamma, A \vdash C} wL \quad \mapsto \quad \frac{!\Gamma^n \vdash C^n}{!\Gamma^n, !A^n \vdash C^n} !wL \\
 \\
 \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow R \quad \mapsto \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C} \rightarrow L \\
 \\
 \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C} \rightarrow L \quad \mapsto \quad \frac{\frac{!\Gamma^n \vdash A^n}{!\Gamma^n \vdash !A^n} !R \quad \frac{B^n \vdash B^n}{!B^n \vdash B^n} \text{ax}}{!\Gamma^n, !A^n \multimap B^n \vdash B^n} \multimap L \\
 \\
 \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C} \rightarrow L \quad \mapsto \quad \frac{\frac{!\Gamma^n, !(A^n \multimap B^n) \vdash B^n}{!\Gamma^n, !(A^n \multimap B^n) \vdash !B^n} !dL \quad \frac{!\Gamma^n, !(A^n \multimap B^n) \vdash !B^n}{!\Gamma^n, !(A^n \multimap B^n) \vdash !B^n} !R}{!\Gamma^n, !\Delta^n, !(A^n \multimap B^n) \vdash C^n} \text{cut}
 \end{array}$$

$$\begin{array}{l}
\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge R \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n \vdash A^n \quad ! \Gamma^n \vdash B^n}{! \Gamma^n \vdash A^n \& B^n} \&R}{A^n \vdash A^n} \text{ax}}{A^n \& B^n \vdash A^n} \&_1 L \\
\frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C} \wedge_1 L \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n, ! A^n \vdash C^n}{!(A^n \& B^n) \vdash A^n} !dL}{!(A^n \& B^n) \vdash !A^n} !R}{! \Gamma^n, !(A^n \& B^n) \vdash C^n} \text{cut} \\
\frac{\Gamma \vdash T}{\Gamma \vdash T} TR \quad \mapsto \quad \frac{\Gamma^n \vdash \top}{\Gamma^n \vdash \top} \top R \\
\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee_1 R \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n \vdash A^n}{! \Gamma^n \vdash !A^n} !R}{! \Gamma^n \vdash !A^n \oplus !B^n} \oplus_1 R}{! \Gamma^n, !A^n \vdash C^n} \oplus L \\
\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \vee L \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n, !A^n \vdash C^n \quad ! \Gamma^n, !B^n \vdash C^n}{! \Gamma^n, !A^n \oplus !B^n \vdash C^n} \oplus L}{! \Gamma^n, !(A^n \oplus B^n) \vdash C^n} !dL}{! \Gamma^n, !(A^n \oplus B^n) \vdash C^n} !dL \\
\frac{\Gamma, F \vdash C}{\Gamma, F \vdash C} FL \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n, 0 \vdash C^n}{! \Gamma^n, !0 \vdash C^n} !dL}{! \Gamma^n, !0 \vdash C^n} !dL}{! \Gamma^n, !0 \vdash C^n} !dL \\
\frac{\Gamma \vdash A}{\Gamma \vdash \forall \xi A} \forall R \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n \vdash A^n}{! \Gamma^n \vdash \forall \xi A^n} \forall R}{A^n[\tau^n/\xi] \vdash A^n[\tau^n/\xi]} \text{ax}}{\forall \xi A^n \vdash A^n[\tau^n/\xi]} \forall L \\
\frac{\Gamma, A[\tau/\xi] \vdash C}{\Gamma, \forall \xi A \vdash C} \forall L \quad \mapsto \quad \frac{\frac{\frac{\frac{\frac{! \Gamma^n, !A^n[\tau^n/\xi] \vdash C^n}{! \Gamma^n, !(A^n[\tau^n/\xi]) \vdash C^n} !R}{! \Gamma^n, !A^n[\tau^n/\xi] \vdash C^n} !dL}{! \Gamma^n, !\forall \xi A^n \vdash ! (A^n[\tau^n/\xi])} !R}{! \Gamma^n, !\forall \xi A^n \vdash C^n} \text{cut} \\
\frac{\Gamma \vdash A[\tau/\xi]}{\Gamma \vdash \exists \xi A} \exists R \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n \vdash A^n[\tau^n/\xi]}{! \Gamma^n \vdash !(A^n[\tau^n/\xi])} !R}{! \Gamma^n \vdash \exists \xi !A^n} \exists R}{! \Gamma^n, !A^n \vdash C^n} \exists L \\
\frac{\Gamma, A \vdash C}{\Gamma, \exists \xi A \vdash C} \exists L \quad \mapsto \quad \frac{\frac{\frac{! \Gamma^n, \exists \xi !A^n \vdash C^n}{! \Gamma^n, \exists \xi !A^n \vdash C^n} \exists L}{! \Gamma^n, \exists \xi !A^n \vdash C^n} !dL}{! \Gamma^n, \exists \xi !A^n \vdash C^n} !dL
\end{array}$$

7.2 Call-by-value translation $A \rightarrow B \mapsto !(A \multimap B)$

Formulas are translated as:

$$\begin{aligned}
X^v &= !X \\
(A \rightarrow B)^v &= !(A^v \multimap B^v) \\
(A \wedge B)^v &= !(A^v \otimes B^v) \\
T^v &= !1 \\
(A \vee B)^v &= !(A^v \oplus B^v) \\
F^v &= !0 \\
(\forall \xi A)^v &= !\forall \xi A^v \\
(\exists \xi A)^v &= !\exists \xi A^v
\end{aligned}$$

The translation of any formula starts with !, we define A^v such that $A^v = !A^v$.

The translation of sequents is $(\Gamma \vdash A)^v = \Gamma^v \vdash A^v$.

This allows one to translate the rules of intuitionistic logic into linear logic:

$$\begin{array}{l}
\frac{\frac{A \vdash A}{\Gamma \vdash A} \text{ax}}{\Gamma, \Delta \vdash B} \text{cut} \quad \mapsto \quad \frac{\frac{A^v \vdash A^v}{\Gamma^v \vdash A^v} \text{ax}}{\Gamma^v, \Delta^v \vdash B^v} \text{cut} \\
\frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C} cL \quad \mapsto \quad \frac{\Gamma^v, A^v, A^v \vdash C^v}{\Gamma^v, A^v \vdash C^v} !cL \\
\frac{\Gamma \vdash C}{\Gamma, A \vdash C} wL \quad \mapsto \quad \frac{\Gamma^v \vdash C^v}{\Gamma^v, A^v \vdash C^v} !wL
\end{array}$$

$$\begin{array}{l}
\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow R \quad \mapsto \quad \frac{\Gamma^v, A^v \vdash B^v}{\Gamma^v \vdash A^v \multimap B^v} \multimap R \\
\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C} \rightarrow L \quad \mapsto \quad \frac{\Gamma^v \vdash A^v \quad \Delta^v, B^v \vdash C^v}{\Gamma^v, \Delta^v, A^v \multimap B^v \vdash C^v} \multimap L \\
\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \wedge R \quad \mapsto \quad \frac{\Gamma^v \vdash A^v \quad \Delta^v \vdash B^v}{\Gamma^v, \Delta^v \vdash A^v \otimes B^v} \otimes R \\
\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C} \wedge L \quad \mapsto \quad \frac{\Gamma^v, A^v, B^v \vdash C^v}{\Gamma^v, A^v \otimes B^v \vdash C^v} \otimes L \\
\frac{}{\vdash T} TR \quad \mapsto \quad \frac{}{\vdash 1} 1R \\
\frac{\Gamma \vdash C}{\Gamma, T \vdash C} TL \quad \mapsto \quad \frac{\Gamma^v \vdash C^v}{\Gamma^v, 1 \vdash C^v} 1L \\
\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee_1 R \quad \mapsto \quad \frac{\Gamma^v \vdash A^v}{\Gamma^v \vdash A^v \oplus B^v} \oplus_1 R \\
\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \vee L \quad \mapsto \quad \frac{\Gamma^v, A^v \vdash C^v \quad \Gamma^v, B^v \vdash C^v}{\Gamma^v, A^v \oplus B^v \vdash C^v} \oplus L \\
\frac{}{\Gamma, F \vdash C} FL \quad \mapsto \quad \frac{}{\Gamma^v, 0 \vdash C^v} 0L \\
\frac{\Gamma \vdash A}{\Gamma \vdash \forall \xi A} \forall R \quad \mapsto \quad \frac{\Gamma^v \vdash A^v}{\Gamma^v \vdash \forall \xi A^v} \forall R \\
\text{We use } (A[\tau/\xi])^v = A^v[\tau^v/\xi]. \\
\frac{\Gamma, A[\tau/\xi] \vdash C}{\Gamma, \forall \xi A \vdash C} \forall L \quad \mapsto \quad \frac{\Gamma^v, A^v[\tau^v/\xi] \vdash C^v}{\Gamma^v, \forall \xi A^v \vdash C^v} \forall L \\
\frac{\Gamma \vdash A[\tau/\xi]}{\Gamma \vdash \exists \xi A} \exists R \quad \mapsto \quad \frac{\Gamma^v \vdash A^v[\tau^v/\xi]}{\Gamma^v \vdash \exists \xi A^v} \exists R \\
\frac{\Gamma, A \vdash C}{\Gamma, \exists \xi A \vdash C} \exists L \quad \mapsto \quad \frac{\Gamma^v, A^v \vdash C^v}{\Gamma^v, \exists \xi A^v \vdash C^v} \exists L
\end{array}$$

7.2.1 Alternative presentation

It is also possible to define A^v as the primitive construction.

$$\begin{array}{l}
X^v = X \\
(A \rightarrow B)^v = !A^v \multimap !B^v \\
(A \wedge B)^v = !A^v \otimes !B^v \\
T^v = 1 \\
(A \vee B)^v = !A^v \oplus !B^v \\
F^v = 0 \\
(\forall \xi A)^v = \forall \xi !A^v \\
(\exists \xi A)^v = \exists \xi !A^v
\end{array}$$

If we define $(\Gamma \vdash A)^v = !\Gamma^v \vdash !A^v$, we have $(\Gamma \vdash A)^v = (\Gamma \vdash A)^v$ and thus we obtain the same translation of proofs.

7.3 Call-by-value Girard's translation $A \rightarrow B \mapsto !(A \multimap B)$

The original version of the call-by-value translation given by Jean-Yves Girard [Gir87] is an optimisation of the previous one using properties of [positive formulas](#).

Formulas are translated as:

$$\begin{aligned} X^w &= !X \\ (A \rightarrow B)^w &= !(A^w \multimap B^w) \\ (A \wedge B)^w &= A^w \otimes B^w \\ T^w &= 1 \\ (A \vee B)^w &= A^w \oplus B^w \\ F^w &= 0 \\ (\forall \xi A)^w &= !\forall \xi A^w \\ (\exists \xi A)^w &= \exists \xi A^w \end{aligned}$$

The translation of any formula is a [positive formula](#).

The translation of sequents is $(\Gamma \vdash A)^w = \Gamma^w \vdash A^w$.

This allows one to translate the rules of intuitionistic logic into linear logic:

$$\begin{array}{l} \frac{}{\Gamma \vdash A} \text{ax} \quad \mapsto \quad \frac{}{A^w \vdash A^w} \text{ax} \\ \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \text{cut} \quad \mapsto \quad \frac{\Gamma^w \vdash A^w \quad \Delta^w, A^w \vdash B^w}{\Gamma^w, \Delta^w \vdash B^w} \text{cut} \\ \frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C} cL \quad \mapsto \quad \frac{\Gamma^w, A^w, A^w \vdash C^w}{\Gamma^w, A^w \vdash C^w} +cL \\ \frac{\Gamma \vdash C}{\Gamma, A \vdash C} wL \quad \mapsto \quad \frac{\Gamma^w \vdash C^w}{\Gamma^w, A^w \vdash C^w} +wL \\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow R \quad \mapsto \quad \frac{\Gamma^w, A^w \vdash B^w}{\Gamma^w \vdash A^w \multimap B^w} \multimap R \\ \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C} \rightarrow L \quad \mapsto \quad \frac{\Gamma^w \vdash A^w \quad \Delta^w, B^w \vdash C^w}{\Gamma^w, \Delta^w, A^w \multimap B^w \vdash C^w} \multimap L \\ \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \wedge R \quad \mapsto \quad \frac{\Gamma^w \vdash A^w \quad \Delta^w \vdash B^w}{\Gamma^w, \Delta^w \vdash A^w \otimes B^w} \otimes R \\ \frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C} \wedge L \quad \mapsto \quad \frac{\Gamma^w, A^w, B^w \vdash C^w}{\Gamma^w, A^w \otimes B^w \vdash C^w} \otimes L \\ \frac{}{\vdash T} TR \quad \mapsto \quad \frac{}{\vdash 1} 1R \\ \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee_1 R \quad \mapsto \quad \frac{\Gamma^w \vdash A^w}{\Gamma^w \vdash A^w \oplus B^w} \oplus_1 R \\ \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \vee L \quad \mapsto \quad \frac{\Gamma^w, A^w \vdash C^w \quad \Gamma^w, B^w \vdash C^w}{\Gamma^w, A^w \oplus B^w \vdash C^w} \oplus L \\ \frac{\Gamma, F \vdash C}{\Gamma, F \vdash C} FL \quad \mapsto \quad \frac{\Gamma^w, 0 \vdash C^w}{\Gamma^w, 0 \vdash C^w} 0L \\ \frac{\Gamma \vdash A}{\Gamma \vdash \forall \xi A} \forall R \quad \mapsto \quad \frac{\Gamma^w \vdash A^w}{\Gamma^w \vdash \forall \xi A^w} \forall R \\ \frac{}{\Gamma \vdash \forall \xi A} \forall L \quad \mapsto \quad \frac{\Gamma^w, (A[\tau/\xi])^w \vdash C^w}{\Gamma^w, \forall \xi A^w \vdash C^w} \forall L \\ \frac{}{\Gamma \vdash \exists \xi A} \exists R \quad \mapsto \quad \frac{\Gamma^w \vdash (A[\tau/\xi])^w}{\Gamma^w \vdash \exists \xi A^w} \exists R \\ \frac{\Gamma, A \vdash C}{\Gamma, \exists \xi A \vdash C} \exists L \quad \mapsto \quad \frac{\Gamma^w, A^w \vdash C^w}{\Gamma^w, \exists \xi A^w \vdash C^w} \exists L \end{array}$$

We use $(A[\tau/\xi])^w \multimap A^w[\tau^w/\xi]$.

$$\begin{array}{l} \frac{\Gamma, A[\tau/\xi] \vdash C}{\Gamma, \forall \xi A \vdash C} \forall L \quad \mapsto \quad \frac{\Gamma^w, (A[\tau/\xi])^w \vdash C^w}{\Gamma^w, \forall \xi A^w \vdash C^w} \forall L \\ \frac{\Gamma \vdash A[\tau/\xi]}{\Gamma \vdash \exists \xi A} \exists R \quad \mapsto \quad \frac{\Gamma^w \vdash (A[\tau/\xi])^w}{\Gamma^w \vdash \exists \xi A^w} \exists R \\ \frac{\Gamma, A \vdash C}{\Gamma, \exists \xi A \vdash C} \exists L \quad \mapsto \quad \frac{\Gamma^w, A^w \vdash C^w}{\Gamma^w, \exists \xi A^w \vdash C^w} \exists L \end{array}$$

Chapter 8

Translations of classical logic

8.1 T-translation $A \rightarrow B \mapsto !?A \multimap ?B$

Formulas are translated as:

$$\begin{aligned}
 X^T &= X \\
 (A \rightarrow B)^T &= !?A^T \multimap ?B^T \\
 (A \wedge B)^T &= ?A^T \& ?B^T \\
 T^T &= \top \\
 (A \vee B)^T &= ?A^T \wp ?B^T \\
 F^T &= \perp \\
 (\neg A)^T &= ?!(A^T)^\perp \\
 (\forall \xi A)^T &= \forall \xi ?A^T \\
 (\exists \xi A)^T &= \exists \xi !?A^T
 \end{aligned}$$

This is extended to sequents by $(\Gamma \vdash \Delta)^T = !?\Gamma^T \vdash ?\Delta^T$.

This allows one to translate the rules of classical logic into linear logic:

$$\begin{aligned}
 \frac{}{A \vdash A} \text{ax} &\mapsto \frac{?A^T \vdash ?A^T}{!A^T \vdash ?A^T} !L \\
 \frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \text{cut} &\mapsto \frac{\frac{!?\Gamma^T \vdash ?A^T, ?\Delta^T}{!?\Gamma^T \vdash !?A^T, ?\Delta^T} !R \quad !?\Gamma'^T, !?A^T \vdash ?\Delta'^T}{!?\Gamma^T, !?\Gamma'^T \vdash ?\Delta^T, ?\Delta'^T} \text{cut} \\
 \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} cL &\mapsto \frac{!?\Gamma^T, !?A^T, !?A^T \vdash ?\Delta^T}{!?\Gamma^T, !?A^T \vdash ?\Delta^T} !cL \\
 \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta} cR &\mapsto \frac{!?\Gamma^T \vdash ?A^T, ?A^T, ?\Delta^T}{!?\Gamma^T \vdash ?A^T, ?\Delta^T} ?cR \\
 \frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} wL &\mapsto \frac{!?\Gamma^T \vdash ?\Delta^T}{!?\Gamma^T, !?A^T \vdash ?\Delta^T} !wL \\
 \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta} wR &\mapsto \frac{!?\Gamma^T \vdash ?\Delta^T}{!?\Gamma^T \vdash ?A^T, ?\Delta^T} ?wR \\
 \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta} \rightarrow R &\mapsto \frac{\frac{!?\Gamma^T, !?A^T \vdash ?B^T, ?\Delta^T}{!?\Gamma^T \vdash !?A^T \multimap ?B^T, ?\Delta^T} \multimap R}{!?\Gamma^T \vdash ?(!?A^T \multimap ?B^T), ?\Delta^T} ?dR
 \end{aligned}$$

$$\begin{array}{c}
\frac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta'} \rightarrow L \quad \mapsto \quad \frac{\frac{\frac{!?\Gamma^T \vdash ?A^T, ?\Delta^T}{!?\Gamma^T \vdash !?A^T, ?\Delta^T} !R \quad \frac{}{?B^T \vdash ?B^T} \text{ax}}{!?\Gamma^T, !?A^T \multimap ?B^T \vdash ?B^T, ?\Delta^T} \multimap L}{\frac{!?\Gamma^T, ?(!?A^T \multimap ?B^T) \vdash ?B^T, ?\Delta^T}{!?\Gamma^T, !?(!?A^T \multimap ?B^T) \vdash ?B^T, ?\Delta^T} ?L}{\frac{!?\Gamma^T, !?(!?A^T \multimap ?B^T) \vdash !?B^T, ?\Delta^T}{!?\Gamma^T, !?(!?A^T \multimap ?B^T) \vdash !?B^T, ?\Delta^T} !dL} !R \quad \frac{}{!?\Gamma^T, !?B^T \vdash ?\Delta^T} \text{cut} \\
\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \wedge R \quad \mapsto \quad \frac{\frac{!?\Gamma^T \vdash ?A^T, ?\Delta^T \quad !?\Gamma^T \vdash ?B^T, ?\Delta^T}{!?\Gamma^T \vdash ?A^T \& ?B^T, ?\Delta^T} \&R}{\frac{!?\Gamma^T \vdash ?(A^T \& B^T), ?\Delta^T}{!?\Gamma^T \vdash ?(A^T \& B^T), ?\Delta^T} ?dR} \&R \\
\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge_1 L \quad \mapsto \quad \frac{\frac{\frac{?A^T \vdash ?A^T}{?A^T \vdash ?A^T} \text{ax}}{?A^T \& ?B^T \vdash ?A^T} \&_1 L}{\frac{?(A^T \& B^T) \vdash ?A^T}{!(A^T \& B^T) \vdash ?A^T} ?L}{\frac{!(A^T \& B^T) \vdash ?A^T}{!(A^T \& B^T) \vdash !?A^T} !dL} !R \quad \frac{}{!?\Gamma^T, !?A^T \vdash ?\Delta^T} \text{cut} \\
\frac{}{\Gamma \vdash \top, \Delta} \top R \quad \mapsto \quad \frac{\frac{}{!?\Gamma^T \vdash \top, ?\Delta^T} \top R}{!?\Gamma^T \vdash ?\top, ?\Delta^T} ?dR} \text{cut} \\
\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta} \vee R \quad \mapsto \quad \frac{\frac{!?\Gamma^T \vdash ?A^T, ?B^T, ?\Delta^T}{!?\Gamma^T \vdash ?A^T \wp ?B^T, ?\Delta^T} \wp R}{\frac{!?\Gamma^T \vdash ?(A^T \wp B^T), ?\Delta^T}{!?\Gamma^T \vdash ?(A^T \wp B^T), ?\Delta^T} ?dR} \wp R \\
\frac{\Gamma, A \vdash \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'} \vee L \quad \mapsto \quad \frac{\frac{\frac{?A^T \vdash ?A^T}{?A^T \wp ?B^T \vdash ?A^T, ?B^T} \text{ax}}{?A^T \wp ?B^T \vdash ?A^T, ?B^T} \wp L}{\frac{?(A^T \wp B^T) \vdash ?A^T, ?B^T}{!(A^T \wp B^T) \vdash ?A^T, ?B^T} ?L}{\frac{!(A^T \wp B^T) \vdash ?A^T, !?B^T}{!(A^T \wp B^T) \vdash ?A^T, !?B^T} !dL} !R \quad \frac{}{!?\Gamma^T, !?B^T \vdash ?\Delta^T} \text{cut} \\
\frac{}{!?\Gamma^T, !?(A^T \wp B^T) \vdash ?A^T, ?\Delta^T} \text{cut} \\
\frac{}{!?\Gamma^T, !?(A^T \wp B^T) \vdash !?A^T, ?\Delta^T} !R \\
\frac{}{!?\Gamma^T, !?(A^T \wp B^T) \vdash ?\Delta^T, ?\Delta^T} \text{cut} \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \perp, \Delta} \perp R \quad \mapsto \quad \frac{\frac{!?\Gamma^T \vdash ?\Delta^T}{!?\Gamma^T \vdash \perp, ?\Delta^T} \perp R}{!?\Gamma^T \vdash ?\perp, ?\Delta^T} ?R \\
\frac{}{F \vdash} FL \quad \mapsto \quad \frac{\frac{\frac{}{\perp \vdash} \perp L}{? \perp \vdash} ?L}{! ? \perp \vdash} !dL} !L \\
\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \neg R \quad \mapsto \quad \frac{\frac{!?\Gamma^T, !?A^T \vdash ?\Delta^T}{!?\Gamma^T \vdash ?!(A^T)^\perp, ?\Delta^T} (\cdot)^\perp R}{\frac{!?\Gamma^T \vdash ?!(A^T)^\perp, ?\Delta^T}{!?\Gamma^T \vdash ?A^T, ?\Delta^T} ?dR} ?dR \\
\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \neg L \quad \mapsto \quad \frac{\frac{!?\Gamma^T, !(A^T)^\perp \vdash ?\Delta^T}{!?\Gamma^T, ?!(A^T)^\perp \vdash ?\Delta^T} (\cdot)^\perp L}{\frac{!?\Gamma^T, ?!(A^T)^\perp \vdash ?\Delta^T}{!?\Gamma^T, ?!(A^T)^\perp \vdash ?\Delta^T} ?L}{\frac{!?\Gamma^T, ?!(A^T)^\perp \vdash ?\Delta^T}{!?\Gamma^T, ?!(A^T)^\perp \vdash ?\Delta^T} ?L} !dL \\
\frac{}{!?\Gamma^T, !?!(A^T)^\perp \vdash ?\Delta^T} !L
\end{array}$$

$$\begin{array}{l}
\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall \xi A, \Delta} \forall R \quad \mapsto \quad \frac{\frac{!?\Gamma^T \vdash ?A^T, ?\Delta^T}{!?\Gamma^T \vdash \forall \xi ?A^T, ?\Delta^T} \forall R}{!?\Gamma^T \vdash ?\forall \xi ?A^T, ?\Delta^T} ?dR \\
\frac{\Gamma, A[\tau/\xi] \vdash \Delta}{\Gamma, \forall \xi A \vdash \Delta} \forall L \quad \mapsto \quad \frac{\frac{\frac{?A^T[\tau^T/\xi] \vdash ?A^T[\tau^T/\xi]}{\forall \xi ?A^T \vdash ?A^T[\tau^T/\xi]} \forall L}{?A^T[\tau^T/\xi] \vdash ?A^T[\tau^T/\xi]} ?L}{!?\forall \xi ?A^T \vdash ?A^T[\tau^T/\xi]} !dL \\
\frac{\Gamma \vdash A[\tau/\xi], \Delta}{\Gamma \vdash \exists \xi A, \Delta} \exists R \quad \mapsto \quad \frac{\frac{\frac{!?\Gamma^T \vdash ?A^T[\tau^T/\xi], ?\Delta^T}{!?\Gamma^T \vdash !?A^T[\tau^T/\xi], ?\Delta^T} !R}{!?\Gamma^T \vdash \exists \xi !?A^T, ?\Delta^T} \exists R}{!?\Gamma^T \vdash ?\exists \xi !?A^T, ?\Delta^T} ?dR \\
\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists \xi A \vdash \Delta} \exists L \quad \mapsto \quad \frac{\frac{\frac{!?\Gamma^T, !?A^T \vdash ?\Delta^T}{!?\Gamma^T, \exists \xi !?A^T \vdash ?\Delta^T} \exists L}{!?\Gamma^T, ?\exists \xi !?A^T \vdash ?\Delta^T} ?L}{!?\Gamma^T, !?\exists \xi !?A^T \vdash ?\Delta^T} !dL
\end{array}$$

cut

8.1.1 Alternative presentation

It is also possible to define $A^{\bar{T}}$ by:

$$\begin{aligned}
X^{\bar{T}} &= ?X \\
(A \rightarrow B)^{\bar{T}} &= !(A^{\bar{T}} \multimap B^{\bar{T}}) \\
(A \wedge B)^{\bar{T}} &= ?(A^{\bar{T}} \& B^{\bar{T}}) \\
T^{\bar{T}} &= ?\top \\
(A \vee B)^{\bar{T}} &= ?(A^{\bar{T}} \wp B^{\bar{T}}) \\
F^{\bar{T}} &= ?\perp \\
(\neg A)^{\bar{T}} &= ??(A^{\bar{T}})^\perp \\
(\forall \xi A)^{\bar{T}} &= ?\forall \xi A^{\bar{T}} \\
(\exists \xi A)^{\bar{T}} &= ?\exists \xi !A^{\bar{T}}
\end{aligned}$$

If we define $(\Gamma \vdash \Delta)^{\bar{T}} = !\Gamma^{\bar{T}} \vdash \Delta^{\bar{T}}$, we have $(\Gamma \vdash \Delta)^{\bar{T}} = (\Gamma \vdash \Delta)^T$ and thus we obtain the same translation of proofs.

8.2 Q-translation $A \rightarrow B \mapsto !(A \multimap ?B)$

Formulas are translated as:

$$\begin{aligned}
X^Q &= !X \\
(A \rightarrow B)^Q &= !(A^Q \multimap ?B^Q) \\
(A \wedge B)^Q &= !(A^Q \otimes B^Q) \\
T^Q &= !1 \\
(A \vee B)^Q &= !(A^Q \oplus B^Q) \\
F^Q &= !0 \\
(\neg A)^Q &= !(A^Q)^\perp \\
(\forall \xi A)^Q &= !\forall \xi ?A^Q \\
(\exists \xi A)^Q &= !\exists \xi A^Q
\end{aligned}$$

The translation of any formula starts with !, we define A^Q such that $A^Q = !A^Q$.

The translation of sequents is $(\Gamma \vdash \Delta)^Q = \Gamma^Q \vdash ?\Delta^Q$.

This allows one to translate the rules of classical logic into linear logic:

$$\begin{array}{c}
\frac{}{A \vdash A} \text{ax} \quad \mapsto \quad \frac{A^Q \vdash A^Q}{A^Q \vdash ?A^Q} ?dR \\
\\
\frac{\Gamma \vdash A, \Delta \quad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \text{cut} \quad \mapsto \quad \frac{\Gamma^Q \vdash ?A^Q, ?\Delta^Q \quad \frac{\Gamma'^Q, A^Q \vdash ?\Delta'^Q}{\Gamma'^Q, ?A^Q \vdash ?\Delta'^Q} ?L}{\Gamma^Q, \Gamma'^Q \vdash ?\Delta^Q, ?\Delta'^Q} \text{cut} \\
\\
\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} cL \quad \mapsto \quad \frac{\Gamma^Q, A^Q, A^Q \vdash ?\Delta^Q}{\Gamma^Q, A^Q \vdash ?\Delta^Q} !cL \\
\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta} cR \quad \mapsto \quad \frac{\Gamma^Q \vdash ?A^Q, ?A^Q, ?\Delta^Q}{\Gamma^Q \vdash ?A^Q, ?\Delta^Q} ?cR \\
\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} wL \quad \mapsto \quad \frac{\Gamma^Q \vdash ?\Delta^Q}{\Gamma^Q, A^Q \vdash ?\Delta^Q} !wL \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta} wR \quad \mapsto \quad \frac{\Gamma^Q \vdash ?\Delta^Q}{\Gamma^Q \vdash ?A^Q, ?\Delta^Q} ?wR \\
\\
\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta} \rightarrow R \quad \mapsto \quad \frac{\frac{\Gamma^Q, A^Q \vdash ?B^Q, ?\Delta^Q}{\Gamma^Q \vdash A^Q \multimap ?B^Q, ?\Delta^Q} \multimap R}{\frac{\Gamma^Q \vdash !(A^Q \multimap ?B^Q), ?\Delta^Q}{\Gamma^Q \vdash ?!(A^Q \multimap ?B^Q), ?\Delta^Q} ?dR} !R \\
\\
\frac{\Gamma \vdash A, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \rightarrow B \vdash \Delta, \Delta'} \rightarrow L \quad \mapsto \quad \frac{\frac{\frac{\frac{\frac{\Gamma'^Q, B^Q \vdash ?\Delta'^Q}{A^Q \vdash A^Q} \text{ax}}{\Gamma'^Q, ?B^Q \vdash ?\Delta'^Q} ?L}{\Gamma'^Q, A^Q \multimap ?B^Q, A^Q \vdash ?\Delta'^Q} \multimap L}{\Gamma'^Q, !(A^Q \multimap ?B^Q), A^Q \vdash ?\Delta'^Q} !dL}{\Gamma^Q \vdash ?A^Q, ?\Delta^Q \quad \frac{\Gamma'^Q, !(A^Q \multimap ?B^Q), ?A^Q \vdash ?\Delta'^Q}{\Gamma^Q, \Gamma'^Q, !(A^Q \multimap ?B^Q) \vdash ?\Delta^Q, ?\Delta'^Q} \text{cut}} ?L \\
\\
\frac{\Gamma \vdash A, \Delta \quad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'} \wedge R \quad \mapsto \quad \frac{\frac{\frac{\frac{\frac{\frac{\Gamma'^Q \vdash ?B^Q, ?\Delta'^Q}{A^Q \vdash A^Q} \text{ax}}{A^Q, B^Q \vdash A^Q \otimes B^Q} \otimes L}{A^Q, B^Q \vdash !(A^Q \otimes B^Q)} !dL}{A^Q, B^Q \vdash ?!(A^Q \otimes B^Q)} ?L}{\Gamma^Q \vdash ?B^Q, ?\Delta'^Q \quad \frac{\Gamma^Q, A^Q \vdash ?!(A^Q \otimes B^Q), ?\Delta'^Q}{\Gamma^Q, ?A^Q \vdash ?!(A^Q \otimes B^Q), ?\Delta'^Q} ?L}{\Gamma^Q, \Gamma'^Q \vdash ?!(A^Q \otimes B^Q), ?\Delta^Q, ?\Delta'^Q} \text{cut}} \\
\\
\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge L \quad \mapsto \quad \frac{\frac{\Gamma^Q, A^Q, B^Q \vdash ?\Delta^Q}{\Gamma^Q, A^Q \otimes B^Q \vdash ?\Delta^Q} \otimes L}{\Gamma^Q, !(A^Q \otimes B^Q) \vdash ?\Delta^Q} !dL \\
\\
\frac{}{\vdash T} TR \quad \mapsto \quad \frac{\frac{\frac{\vdash 1}{\vdash !1} 1R}{\vdash ?!1} ?dR}{\vdash ?!1} ?dR \\
\\
\frac{\Gamma \vdash \Delta}{\Gamma, T \vdash \Delta} TL \quad \mapsto \quad \frac{\frac{\Gamma^Q \vdash ?\Delta^Q}{\Gamma^Q, 1 \vdash ?\Delta^Q} 1L}{\Gamma^Q, !1 \vdash ?\Delta^Q} !dL
\end{array}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta} \vee_1 R \quad \mapsto \quad \frac{\frac{\frac{\frac{\overline{A^Q \vdash A^Q} \text{ax}}{A^Q \vdash A^Q \oplus B^Q} \oplus_1 R}{A^Q \vdash !(A^Q \oplus B^Q)} !R}{A^Q \vdash ?!(A^Q \oplus B^Q)} ?dR}{\Gamma^Q \vdash ?A^Q, ?\Delta^Q} ?L}{\Gamma^Q \vdash ?!(A^Q \oplus B^Q), ?\Delta^Q} \text{cut}}{\Gamma^Q, A^Q \vdash ?\Delta^Q} \oplus L$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \vee L \quad \mapsto \quad \frac{\Gamma^Q, A^Q \vdash ?\Delta^Q \quad \Gamma^Q, B^Q \vdash ?\Delta^Q}{\Gamma^Q, A^Q \oplus B^Q \vdash ?\Delta^Q} \oplus L}{\Gamma^Q, !(A^Q \oplus B^Q) \vdash ?\Delta^Q} !dL$$

$$\frac{}{\Gamma, F \vdash \Delta} FL \quad \mapsto \quad \frac{\Gamma^Q, 0 \vdash ?\Delta^Q}{\Gamma^Q, !0 \vdash ?\Delta^Q} 0L}{\Gamma^Q, A^Q \vdash ?\Delta^Q} !dL$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \neg R \quad \mapsto \quad \frac{\Gamma^Q, A^Q \vdash ?\Delta^Q}{\Gamma^Q \vdash (A^Q)^\perp, ?\Delta^Q} (\cdot)^\perp R}{\Gamma^Q \vdash !(A^Q)^\perp, ?\Delta^Q} !R}{\Gamma^Q \vdash ?!(A^Q)^\perp, ?\Delta^Q} ?dR}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \neg L \quad \mapsto \quad \frac{\Gamma^Q \vdash ?A^Q, ?\Delta^Q}{\Gamma^Q, !(A^Q)^\perp \vdash ?\Delta^Q} (\cdot)^\perp L}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall \xi A, \Delta} \forall R \quad \mapsto \quad \frac{\Gamma^Q \vdash ?A^Q, ?\Delta^Q}{\Gamma^Q \vdash \forall \xi ?A^Q, ?\Delta^Q} \forall R}{\Gamma^Q \vdash !\forall \xi ?A^Q, ?\Delta^Q} !R}$$

We use $(A[\tau/\xi])^Q = A^Q[\tau^Q/\xi]$.

$$\frac{\Gamma, A[\tau/\xi] \vdash \Delta}{\Gamma, \forall \xi A \vdash \Delta} \forall L \quad \mapsto \quad \frac{\frac{\Gamma^Q, A^Q[\tau^Q/\xi] \vdash ?\Delta^Q}{\Gamma^Q, ?A^Q[\tau^Q/\xi] \vdash ?\Delta^Q} ?L}{\Gamma^Q, \forall \xi ?A^Q \vdash ?\Delta^Q} \forall L}{\Gamma^Q, !\forall \xi ?A^Q \vdash ?\Delta^Q} !dL}$$

$$\frac{\Gamma \vdash A[\tau/\xi], \Delta}{\Gamma \vdash \exists \xi A, \Delta} \exists R \quad \mapsto \quad \frac{\frac{\frac{\frac{\overline{A^Q[\tau^Q/\xi] \vdash A^Q[\tau^Q/\xi]} \text{ax}}{A^Q[\tau^Q/\xi] \vdash \exists \xi A^Q} \exists R}{A^Q[\tau^Q/\xi] \vdash !\exists \xi A^Q} !R}{A^Q[\tau^Q/\xi] \vdash ?!\exists \xi A^Q} ?dR}{\Gamma^Q \vdash ?A^Q[\tau^Q/\xi], ?\Delta^Q} ?L}{\Gamma^Q \vdash ?!\exists \xi A^Q, ?\Delta^Q} \text{cut}}{\Gamma^Q \vdash ?!\exists \xi A^Q, ?\Delta^Q} \text{cut}}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists \xi A \vdash \Delta} \exists L \quad \mapsto \quad \frac{\Gamma^Q, A^Q \vdash ?\Delta^Q}{\Gamma^Q, \exists \xi A^Q \vdash ?\Delta^Q} \exists L}{\Gamma^Q, !\exists \xi A^Q \vdash ?\Delta^Q} !dL}$$

8.2.1 Alternative presentation

It is also possible to define A^Q as the primitive construction.

$$\begin{aligned}
X^Q &= X \\
(A \rightarrow B)^Q &= !A^Q \multimap ?!B^Q \\
(A \wedge B)^Q &= !A^Q \otimes !B^Q \\
T^Q &= 1 \\
(A \vee B)^Q &= !A^Q \oplus !B^Q \\
F^Q &= 0 \\
(\neg A)^Q &= ?(A^Q)^\perp \\
(\forall \xi A)^Q &= \forall \xi ?!A^Q \\
(\exists \xi A)^Q &= \exists \xi !A^Q
\end{aligned}$$

If we define $(\Gamma \vdash \Delta)^Q = !\Gamma^Q \vdash ?!\Delta^Q$, we have $(\Gamma \vdash \Delta)^Q = (\Gamma \vdash \Delta)^Q$ and thus we obtain the same translation of proofs.

Chapter 9

Light linear logics

Light linear logics are variants of linear logic characterizing complexity classes. They are designed by defining alternative exponential connectives, which induce a complexity bound on the cut-elimination procedure.

Light linear logics are one of the approaches used in *implicit computational complexity*, the area studying the computational complexity of programs without referring to external measuring conditions or particular machine models.

9.1 Elementary linear logic

We present here the intuitionistic version of *elementary linear logic*, ELL. Moreover we restrict to the fragment without additive connectives.

The language of formulas is the same one as that of (multiplicative) ILL:

$$A ::= X \mid A \otimes A \mid A \multimap A \mid !A \mid \forall X A$$

The sequent calculus rules are the same ones as for ILL, except for the rules dealing with the exponential connectives:

$$\frac{\Gamma \vdash A}{! \Gamma \vdash !A} !mf \quad \frac{\Gamma, !A, !A \vdash C}{\Gamma, !A \vdash C} !cL \quad \frac{\Gamma \vdash C}{\Gamma, !A \vdash C} !wL$$

The *depth* of a derivation π is the maximum number of (!mf) rules in a branch of π .

We consider the function $K(.,.)$ defined by:

$$K(0, n) = n, \quad K(k + 1, n) = 2^{K(k, n)}.$$

Theorem 10. *If π is an ELL proof of depth d , and R is the corresponding ELL proof-net, then R can be reduced to its normal form by cut elimination in at most $K(d + 1, |\pi|)$ steps, where $|\pi|$ is the size of π .*

A function f on integers is *elementary recursive* if there exists an integer h and a Turing machine which computes f in time bounded by $K(h, n)$, where n is the size of the input.

Theorem 11. *The functions representable in ELL are exactly the elementary recursive functions.*

One also often considers the *affine* variant of ELL, called *elementary affine logic* EAL, which is defined by adding unrestricted weakening:

$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C} wL$$

It enjoys the same properties as ELL.

Elementary linear logic was introduced together with light linear logic [Gir98].

9.2 Light linear logic

We present the intuitionistic version of *light linear logic* LLL, without additive connectives. The language of formulas is:

$$A ::= X \mid A \otimes A \mid A \multimap A \mid !A \mid \S A \mid \forall X A$$

The sequent calculus rules are the same ones as for ILL, except for the rules dealing with the exponential connectives:

$$\frac{\Gamma \vdash A}{!\Gamma \vdash !A} !f \quad \frac{\Gamma, \Delta \vdash A}{!\Gamma, \S \Delta \vdash \S A} \S \quad \frac{\Gamma, !A, !A \vdash C}{\Gamma, !A \vdash C} !cL \quad \frac{\Gamma \vdash C}{\Gamma, !A \vdash C} !wL$$

In the (!f) rule, Γ must contain *at most one* formula.

The *depth* of a derivation π is the maximum number of (!f) and (§) rules in a branch of π .

Theorem 12. *If π is an LLL proof of depth d , and R is the corresponding LLL proof-net, then R can be reduced to its normal form by cut elimination in $O((d+1)|\pi|^{2^{d+1}})$ steps, where $|\pi|$ is the size of π .*

The class FP is the class of functions on binary lists which are computable in polynomial time on a Turing machine.

Theorem 13. *The class of functions on binary lists representable in LLL is exactly FP.*

In the literature one also often considers the *affine* variant of LLL, called *light affine logic*, LAL.

9.3 Soft linear logic

We consider the intuitionistic version of *soft linear logic*, SLL.

The language of formulas is the same one as that of ILL:

$$A ::= X \mid A \otimes A \mid A \multimap A \mid A \& A \mid A \oplus A \mid !A \mid \forall X A$$

The sequent calculus rules are the same ones as for ILL, except for the rules dealing with the exponential connectives:

$$\frac{\Gamma \vdash A}{!\Gamma \vdash !A} !mf \quad \frac{\Gamma, A^{(n)} \vdash C}{\Gamma, !A \vdash C} \text{mplex}$$

The rule mplex is the *multiplexing* rule. In its premise, $A^{(n)}$ stands for n occurrences of formula A . As particular instances of mplex for $n = 0$ and 1 respectively, we get weakening and dereliction:

$$\frac{\Gamma \vdash C}{\Gamma, !A \vdash C} \quad \frac{\Gamma, A \vdash C}{\Gamma, !A \vdash C}$$

The *depth* of a derivation π is the maximum number of (!mf) rules in a branch of π .

Theorem 14. *If π is an SLL proof of depth d , and R is the corresponding SLL proof-net, then R can be reduced to its normal form by cut elimination in $O(|\pi|^d)$ steps, where $|\pi|$ is the size of π .*

Theorem 15. *The class of functions on binary lists representable in SLL is exactly FP.*

Soft linear logic was introduced in [Laf04].

Part II

Semantics

Chapter 10

Semantics

Linear Logic has numerous semantics some of which are described in details in the next sections.

- [Coherent semantics](#)
- [Phase semantics](#)
- [Categorical semantics](#)
- [Relational semantics](#)
- [Finiteness semantics](#)
- [Geometry of interaction](#)
- [Game semantics](#)

[Common properties](#) may be found in most of these models. We will denote by $A \longrightarrow B$ the fact that there is a canonical morphism from A to B and by $A \cong B$ the fact that there is a canonical [isomorphism](#) between A and B . By “canonical” we mean that these (iso)morphisms are natural transformations.

10.1 Orthogonality relation

Orthogonality relations are used pervasively throughout linear logic models, being often used to define somehow the duality operator $(-)^{\perp}$.

Definition 8 (Orthogonality relation). *Let A and B be two sets. An orthogonality relation on A and B is a binary relation $\mathcal{R} \subseteq A \times B$. We say that $a \in A$ and $b \in B$ are orthogonal, and we note $a \perp b$, whenever $(a, b) \in \mathcal{R}$.*

Let us now assume an orthogonality relation over A and B .

Definition 9 (Orthogonal sets). *Let $\alpha \subseteq A$. We define its orthogonal set α^{\perp} as $\alpha^{\perp} := \{b \in B \mid \forall a \in \alpha, a \perp b\}$.*

Symmetrically, for any $\beta \subseteq B$, we define $\beta^{\perp} := \{a \in A \mid \forall b \in \beta, a \perp b\}$.

Orthogonal sets define Galois connections and share many common properties.

Proposition 7. *For any sets $\alpha, \alpha' \subseteq A$:*

- $\alpha \subseteq \alpha^{\perp\perp}$
- If $\alpha \subseteq \alpha'$, then $\alpha'^{\perp} \subseteq \alpha^{\perp}$
- $\alpha^{\perp\perp\perp} = \alpha^{\perp}$

Chapter 11

Coherent semantics

Coherent semantics was invented by Girard in the paper *The system F, 15 years later* [Gir86] with the objective of building a denotational interpretation of second order intuitionistic logic (aka polymorphic lambda-calculus).

Coherent semantics is based on the notion of *stable functions* that was initially proposed by Gérard Berry. Stability is a condition on Scott continuous functions that expresses the determinism of the relation between the output and the input: the typical Scott continuous but non stable function is the *parallel or* because when the two inputs are both set to **true**, only one of them is the reason why the result is **true** but there is no way to determine which one.

A further achievement of coherent semantics was that it allowed to endow the set of stable functions from X to Y with a structure of domain, thus closing the category of coherent spaces and stable functions. However the most interesting point was the discovery of a special class of stable functions, *linear functions*, which was the first step leading to Linear Logic.

11.1 The cartesian closed structure of coherent semantics

There are three equivalent definitions of coherent spaces: the first one, *coherent spaces as domains*, is interesting from a historical point of view as it emphasises the fact that coherent spaces are particular cases of Scott domains. The second one, *coherent spaces as graphs*, is the most commonly used and will be our “official” definition in the sequel. The last one, *cliqued spaces* is a particular example of a more general scheme that one could call “symmetric reducibility”; this scheme is underlying lots of constructions in linear logic such as [phase semantics](#) or the proof of strong normalisation for proof-nets.

11.1.1 Coherent spaces

A coherent space X is a collection of subsets of a set $|X|$ satisfying some conditions that will be detailed shortly. The elements of X are called the *cliques* of X (for reasons that will be made clear in a few lines). The set $|X|$ is called the *web* of X and its elements are called the *points* of X ; thus a clique is a set of points. Note that the terminology is a bit ambiguous as the points of X are the elements of the web of X , not the elements of X .

The definitions below give three equivalent conditions that have to be satisfied by the cliques of a coherent space.

11.1.1.1 As domains

The cliques of X have to satisfy:

- subset closure: if $x \subset y \in X$ then $x \in X$,
- singletons: $\{a\} \in X$ for $a \in |X|$.
- binary compatibility: if A is a family of pairwise compatible cliques of X , that is if $x \cup y \in X$ for any $x, y \in A$, then $\bigcup A \in X$.

A coherent space is thus ordered by inclusion; one easily checks that it is a domain. In particular finite cliques of X correspond to compact elements.

11.1.1.2 As graphs

There is a reflexive and symmetric relation \supset_X on $|X|$ (the *coherence relation*) such that any subset x of $|X|$ is a clique of X iff $\forall a, b \in x, a \supset_X b$. In other terms X is the set of complete subgraphs of the simple unoriented graph of the \supset_X relation; this is the reason why elements of X are called *cliques*.

The *strict coherence relation* \frown_X on X is defined by: $a \frown_X b$ iff $a \neq b$ and $a \supset_X b$.

A coherent space in the domain sense is seen to be a coherent space in the graph sense by setting $a \supset_X b$ iff $\{a, b\} \in X$; conversely one can check that cliques in the graph sense are subset closed and satisfy the binary compatibility condition.

A coherent space is completely determined by its web and its coherence relation, or equivalently by its web and its strict coherence.

11.1.1.3 As cliqued spaces

Definition 10 (Duality). *Let $x, y \subseteq |X|$ be two sets. We will say that they are dual, written $x \perp y$ if their intersection contains at most one element: $\text{Card}(x \cap y) \leq 1$. As usual, it defines an *orthogonality relation* over $\mathfrak{P}(|X|)$.*

The last way to express the conditions on the cliques of a coherent space X is simply to say that we must have $X^{\perp\perp} = X$.

11.1.1.4 Equivalence of definitions

Let X be a cliqued space and define a relation on $|X|$ by setting $a \supset_X b$ iff there is $x \in X$ such that $a, b \in x$. This relation is obviously symmetric; it is also reflexive because all singletons belong to X : if $a \in |X|$ then $\{a\}$ is dual to any element of X^\perp (actually $\{a\}$ is dual to any subset of $|X|$), thus $\{a\}$ is in $X^{\perp\perp}$, thus in X .

Let $a \supset_X b$. Then $\{a, b\} \in X$; indeed there is an $x \in X$ such that $a, b \in x$. This x is dual to any $y \in X^\perp$, that is meets any $y \in X^\perp$ in at most one point. Since $\{a, b\} \subset x$ this is also true of $\{a, b\}$, so that $\{a, b\}$ is in $X^{\perp\perp}$ thus in X .

Now let x be a clique for \supset_X and y be an element of X^\perp . Suppose $a, b \in x \cap y$, then since a and b are coherent (by hypothesis on x) we have $\{a, b\} \in X$ and since $y \in X^\perp$ we must have that $\{a, b\}$ and y meet in at most one point. Thus $a = b$ and we have shown that x and y are dual. Since y was arbitrary this means that x is in $X^{\perp\perp}$, thus in X . Finally we get that any set of pairwise coherent points of X is in X . Conversely given $x \in X$ its points are obviously pairwise coherent so eventually we get that X is a coherent space in the graph sense.

Conversely given a coherent space X in the graph sense, one can check that it is a cliqued space. Call *anticlique* a set $y \subset |X|$ of pairwise incoherent points: for all a, b in y , if $a \supset_X b$ then $a = b$. Any anticlique intersects any clique in at most one point: let x be a clique and y be an anticlique, then if $a, b \in x \cap y$, since $a, b \in x$ we have $a \supset_X b$ and since y is an anticlique we have $a = b$. Thus $y \in X^\perp$. Conversely given any $y \in X^\perp$ and $a, b \in y$, suppose $a \supset_X b$. Then

$\{a, b\} \in X$, thus $\{a, b\} \perp y$ which entails that $\{a, b\}$ has at most one point so that $a = b$: we have shown that any two elements of y are incoherent.

Thus the collection of anticliques of X is the dual X^\perp of X . Note that the incoherence relation defined above is reflexive and symmetric, so that X^\perp is a coherent space in the graph sense. Thus we can do for X^\perp exactly what we've just done for X and consider the anti-anticliques, that is the anticliques for the incoherent relation which are the cliques for the in-incoherent relation. It is not difficult to see that this in-incoherence relation is just the coherence relation we started with; we thus obtain that $X^{\perp\perp} = X$, so that X is a cliqued space.

11.1.2 Stable functions

Definition 11 (Stable function). *Let X and Y be two coherent spaces. A function $F : X \rightarrow Y$ is stable if it satisfies:*

- *it is non decreasing: for any $x, y \in X$ if $x \subset y$ then $F(x) \subset F(y)$;*
- *it is continuous (in the Scott sense): if A is a directed family of cliques of X , that is if for any $x, y \in A$ there is a $z \in A$ such that $x \cup y \subset z$, then $\bigcup_{x \in A} F(x) = F(\bigcup A)$;*
- *it satisfies the stability condition: if $x, y \in X$ are compatible, that is if $x \cup y \in X$, then $F(x \cap y) = F(x) \cap F(y)$.*

This definition is admittedly not very tractable. An equivalent and most useful characterisation of stable functions is given by the following theorem.

Theorem 16. *Let $F : X \rightarrow Y$ be a non-decreasing function from the coherent space X to the coherent space Y . The function F is stable iff it satisfies: for any $x \in X$, $b \in |Y|$, if $b \in F(x)$ then there is a finite clique $x_0 \subset x$ such that:*

- $b \in F(x_0)$,
- *for any $y \subset x$ if $b \in F(y)$ then $x_0 \subset y$ (x_0 is “the” minimum sub-clique of x such that $b \in F(x_0)$).*

Note that the stability condition doesn't depend on the coherent space structure and can be expressed more generally for continuous functions on domains. However, as mentioned in the introduction, the restriction to coherent spaces allows to endow the set of stable functions from X to Y with a structure of coherent space.

Definition 12 (The space of stable functions). *Let X and Y be coherent spaces. We denote by X_{fin} the set of finite cliques of X . The function space $X \rightarrow Y$ is defined by:*

- $|X \rightarrow Y| = X_{\text{fin}} \times |Y|$,
- $(x_0, a) \supset_{X \rightarrow Y} (y_0, b)$ iff $\begin{cases} \text{if } x_0 \cup y_0 \in X \text{ then } a \supset_Y b, \\ \text{if } x_0 \cup y_0 \in X \text{ and } a = b \text{ then } x_0 = y_0 \end{cases}$.

One could equivalently define the strict coherence relation on $X \rightarrow Y$ by: $(x_0, a) \frown_{X \rightarrow Y} (y_0, b)$ iff when $x_0 \cup y_0 \in X$ then $a \frown_Y b$ (equivalently $x_0 \cup y_0 \notin X$ or $a \frown_Y b$).

Definition 13 (Trace of a stable function). *Let $F : X \rightarrow Y$ be a function. The trace of F is the set:*

$$\text{Tr}(F) = \{(x_0, b), x_0 \text{ minimal such that } b \in F(x_0)\}.$$

Theorem 17. *F is stable iff $\text{Tr}(F)$ is a clique of the function space $X \rightarrow Y$.*

In particular the continuity of F entails that if x_0 is minimal such that $b \in F(x_0)$, then x_0 is finite.

Definition 14 (The evaluation function). *Let f be a clique in $X \rightarrow Y$. We define a function $\text{Fun } f : X \rightarrow Y$ by: $\text{Fun } f(x) = \{b \in Y, \text{ there is } x_0 \subset x \text{ such that } (x_0, b) \in f\}$.*

Theorem 18 (Closure). *If f is a clique of the function space $X \rightarrow Y$ then we have $\text{Tr}(\text{Fun } f) = f$. Conversely if $F : X \rightarrow Y$ is a stable function then we have $F = \text{Fun } \text{Tr}(F)$.*

11.1.3 Cartesian product

Definition 15 (Cartesian product). *Let X_1 and X_2 be two coherent spaces. We define the coherent space $X_1 \& X_2$ (read X_1 “with” X_2):*

- the web is the disjoint union of the webs: $|X_1 \& X_2| = \{1\} \times |X_1| \cup \{2\} \times |X_2|$;
- the coherence relation is the serie composition of the relations on X_1 and X_2 : $(i, a) \circ_{X_1 \& X_2} (j, b)$ iff either $i \neq j$ or $i = j$ and $a \circ_{X_i} b$.

This definition is just the way to put a coherent space structure on the cartesian product. Indeed one easily shows the

Theorem 19. *Given cliques x_1 and x_2 in X_1 and X_2 , we define the subset $\langle x_1, x_2 \rangle$ of $|X_1 \& X_2|$ by: $\langle x_1, x_2 \rangle = \{1\} \times x_1 \cup \{2\} \times x_2$. Then $\langle x_1, x_2 \rangle$ is a clique in $X_1 \& X_2$.*

Conversely, given a clique $x \in X_1 \& X_2$, for $i = 1, 2$ we define $\pi_i(x) = \{a \in X_i, (i, a) \in x\}$. Then $\pi_i(x)$ is a clique in X_i and the function $\pi_i : X_1 \& X_2 \rightarrow X_i$ is stable.

Furthermore these two operations are inverse of each other: $\pi_i(\langle x_1, x_2 \rangle) = x_i$ and $\langle \pi_1(x), \pi_2(x) \rangle = x$. In particular any clique in $X_1 \& X_2$ is of the form $\langle x_1, x_2 \rangle$.

Altogether the results above (and a few other more that we shall leave to the reader) allow to get:

Theorem 20. *The category of coherent spaces and stable functions is cartesian closed.*

In particular this means that if we define $\text{Eval} : (X \rightarrow Y) \& X \rightarrow Y$ by: $\text{Eval}(\langle f, x \rangle) = \text{Fun } f(x)$ then Eval is stable.

11.2 The monoidal structure of coherent semantics

11.2.1 Linear functions

Definition 16 (Linear function). *A function $F : X \rightarrow Y$ is linear if it is stable and furthermore satisfies: for any family A of pairwise compatible cliques of X , that is such that for any $x, y \in A$, $x \cup y \in X$, we have $\bigcup_{x \in A} F(x) = F(\bigcup A)$.*

In particular if we take A to be the empty family, then we have $F(\emptyset) = \emptyset$.

The condition for linearity is quite similar to the condition for Scott continuity, except that we dropped the constraint that A is *directed*. Linearity is therefore much stronger than stability: most stable functions are not linear.

However most of the functions seen so far are linear. Typically the function $\pi_i : X_1 \& X_2 \rightarrow X_i$ is linear from which one may deduce that the *with* construction is also a cartesian product in the category of coherent spaces and linear functions.

As with stable function we have an equivalent and much more tractable characterisation of linear function:

Theorem 21. *Let $F : X \rightarrow Y$ be a continuous function. Then F is linear iff it satisfies: for any clique $x \in X$ and any $b \in F(x)$ there is a unique $a \in x$ such that $b \in F(\{a\})$.*

Just as the characterisation theorem for stable functions allowed us to build the coherent space of stable functions, this theorem will help us to endow the set of linear maps with a structure of coherent space.

Definition 17 (The linear functions space). *Let X and Y be coherent spaces. The linear function space $X \multimap Y$ is defined by:*

- $|X \multimap Y| = |X| \times |Y|$,
- $(a, b) \multimap_{X \multimap Y} (a', b')$ iff $\begin{cases} \text{if } a \multimap_X a' \text{ then } b \multimap_Y b' \\ \text{if } a \multimap_X a' \text{ and } b = b' \text{ then } a = a' \end{cases}$

Equivalently one could define the strict coherence to be: $(a, b) \multimap_{X \multimap Y} (a', b')$ iff $a \multimap_X a'$ entails $b \multimap_Y b'$.

Definition 18 (Linear trace). *Let $F : X \rightarrow Y$ be a function. The linear trace of F denoted as $\text{LinTr}(F)$ is the set: $\text{LinTr}(F) = \{(a, b) \in |X| \times |Y| \text{ such that } b \in F(\{a\})\}$.*

Theorem 22. *If F is linear then $\text{LinTr}(F)$ is a clique of $X \multimap Y$.*

Definition 19 (Evaluation of linear function). *Let f be a clique of $X \multimap Y$. We define the function $\text{LinFun } f : X \rightarrow Y$ by: $\text{LinFun } f(x) = \{b \in |Y| \text{ such that there is an } a \in x \text{ satisfying } (a, b) \in f\}$.*

Theorem 23 (Linear closure). *Let f be a clique in $X \multimap Y$. Then we have $\text{LinTr}(\text{LinFun } f) = f$. Conversely if $F : X \rightarrow Y$ is linear then we have $F = \text{LinFun } \text{LinTr}(F)$.*

It remains to define a tensor product and we will get that the category of coherent spaces with linear functions is monoidal symmetric (it is actually *-autonomous).

11.2.2 Tensor product

Definition 20 (Tensor product). *Let X and Y be coherent spaces. Their tensor product $X \otimes Y$ is defined by: $|X \otimes Y| = |X| \times |Y|$ and $(a, b) \multimap_{X \otimes Y} (a', b')$ iff $a \multimap_X a'$ and $b \multimap_Y b'$.*

Theorem 24. *The category of coherent spaces with linear maps and tensor product is [monoidal symmetric closed](#).*

The closedness is a consequence of the existence of the linear isomorphism:

$$\varphi : X \otimes Y \multimap Z \xrightarrow{\sim} X \multimap (Y \multimap Z)$$

that is defined by its linear trace: $\text{LinTr}(\varphi) = \{((a, b), c), (a, (b, c)), a \in |X|, b \in |Y|, c \in |Z|\}$.

11.2.3 Linear negation

Definition 21 (Linear negation). *Let X be a coherent space. We define the incoherence relation on $|X|$ by: $a \multimap_X b$ iff $a \multimap_X b$ entails $a = b$. The incoherence relation is reflexive and symmetric; we call dual or linear negation of X the associated coherent space denoted X^\perp , thus defined by: $|X^\perp| = |X|$ and $a \multimap_{X^\perp} b$ iff $a \multimap_X b$.*

The cliques of X^\perp are called the *anticliques* of X . As seen in the section on cliqued spaces we have $X^{\perp\perp} = X$.

Theorem 25. *The category of coherent spaces with linear maps, tensor product and linear negation is $*$ -autonomous.*

This is in particular consequence of the existence of the isomorphism:

$$\varphi : X \multimap Y \xrightarrow{\sim} Y^\perp \multimap X^\perp$$

defined by its linear trace: $\text{LinTr}(\varphi) = \{(a, b), (b, a)\}, a \in |X|, b \in |Y|\}.$

11.3 Exponentials

In linear algebra, bilinear maps may be factorized through the tensor product. Similarly there is a coherent space $!X$ that allows to factorize stable functions through linear functions.

Definition 22 (Of course). *Let X be a coherent space; recall that X_{fin} denotes the set of finite cliques of X . We define the space $!X$ (read “of course X ”) by: $!|X| = X_{\text{fin}}$ and $x_0 \supset_{!X} y_0$ iff $x_0 \cup y_0$ is a clique of X .*

Thus a clique of $!X$ is a set of finite cliques of X the union of which is a clique of X .

Theorem 26. *Let X be a coherent space. Denote by $\beta : X \rightarrow !X$ the stable function whose trace is: $\text{Tr}(\beta) = \{(x_0, x_0), x_0 \in X_{\text{fin}}\}$. Then for any coherent space Y and any stable function $F : X \rightarrow Y$ there is a unique linear function $\bar{F} : !X \rightarrow Y$ such that $F = \bar{F} \circ \beta$.*

Furthermore we have $X \rightarrow Y = !X \multimap Y$.

Theorem 27 (The exponential isomorphism). *Let X and Y be two coherent spaces. Then there is a linear isomorphism:*

$$\varphi : !(X \& Y) \xrightarrow{\sim} !X \otimes !Y.$$

The iso φ is defined by its trace:

$$\text{Tr}(\varphi) = \{(x_0, (\pi_1(x_0), \pi_2(x_0))), x_0 \text{ finite clique of } X \& Y\}.$$

This isomorphism, that sends an additive structure (the web of a with is obtained by disjoint union) onto a multiplicative one (the web of a tensor is obtained by cartesian product) is the reason why the of course is called an *exponential*.

11.4 Dual connectives and neutrals

By linear negation all the constructions defined so far ($\&, \otimes, !$) have a dual.

11.4.1 The direct sum

The dual of $\&$ is \oplus defined by: $X \oplus Y = (X^\perp \& Y^\perp)^\perp$. An equivalent definition is given by: $|X \oplus Y| = |X \& Y| = \{1\} \times |X| \cup \{2\} \times |Y|$ and $(i, a) \supset_{X \oplus Y} (j, b)$ iff $i = j = 1$ and $a \supset_X b$, or $i = j = 2$ and $a \supset_Y b$.

Theorem 28. *Let x' be a clique of $X \oplus Y$; then x' is of the form $\{i\} \times x$ where $i = 1$ and $x \in X$, or $i = 2$ and $x \in Y$.*

Denote $\text{inl} : X \rightarrow X \oplus Y$ the function defined by $\text{inl}(x) = \{1\} \times x$ and by $\text{inr} : Y \rightarrow X \oplus Y$ the function defined by $\text{inr}(x) = \{2\} \times x$. Then inl and inr are linear.

If $F : X \rightarrow Z$ and $G : Y \rightarrow Z$ are “linear” functions then the function $H : X \oplus Y \rightarrow Z$ defined by $H(\text{inl}(x)) = F(x)$ and $H(\text{inr}(y)) = G(y)$ is linear.

In other terms $X \oplus Y$ is the direct sum of X and Y . Note that in the theorem all functions are *linear*. Things doesn't work so smoothly for stable functions. Historically it was after noting this defect of coherent semantics w.r.t. the intuitionistic implication that Girard was led to discover linear functions.

11.4.2 The par and the why not

We now come to the most mysterious constructions of coherent semantics: the duals of the tensor and the of course.

The *par* is the dual of the tensor, thus defined by: $X \wp Y = (X^\perp \otimes Y^\perp)^\perp$. From this one can deduce the definition in graph terms: $|X \wp Y| = |X \otimes Y| = |X| \times |Y|$ and $(a, b) \frown_{X \wp Y} (a', b')$ iff $a \frown_X a'$ or $b \frown_Y b'$. With this definition one sees that we have:

$$X \multimap Y = X^\perp \wp Y$$

for any coherent spaces X and Y . This equation can be seen as an alternative definition of the par: $X \wp Y = X^\perp \multimap Y$.

Similarly the dual of the of course is called *why not* defined by: $?X = (!X^\perp)^\perp$. From this we deduce the definition in the graph sense which is a bit tricky: $|?X|$ is the set of finite anticliques of X , and given two finite anticliques x and y of X we have $x \frown_{?X} y$ iff there is $a \in x$ and $b \in y$ such that $a \frown_X b$.

Note that both for the par and the why not it is much more convenient to define the strict coherence than the coherence.

With these two last constructions, the equation between the stable function space, the of course and the linear function space may be written:

$$X \rightarrow Y = ?X^\perp \wp Y.$$

11.4.3 One and bottom

Depending on the context we denote by 1 or \perp the coherent space whose web is a singleton and whose coherence relation is the trivial reflexive relation.

Theorem 29. 1 is neutral for tensor, that is, there is a linear isomorphism $\varphi : X \otimes 1 \xrightarrow{\sim} X$. Similarly \perp is neutral for par.

11.4.4 Zero and top

Depending on the context we denote by 0 or \top the coherent space with empty web.

Theorem 30. 0 is neutral for the direct sum \oplus , \top is neutral for the cartesian product $\&$.

Remark 1. It is one of the main defect of coherent semantics w.r.t. linear logic that it identifies the neutrals: in coherent semantics $0 = \top$ and $1 = \perp$. However there is no known semantics of LL that solves this problem in a satisfactory way.

11.5 After coherent semantics

Coherent semantics was an important milestone in the modern theory of logic of programs, in particular because it led to the invention of Linear Logic, and more generally because it establishes a strong link between logic and linear algebra; this link is nowadays acknowledged by the customary use of [monoidal categories](#) in logic. In some sense coherent semantics is a precursor

of many forthcoming works that explore the linear nature of logic as for example [geometry of interaction](#) which interprets proofs by operators or [finiteness semantics](#) which interprets formulas as vector spaces and resulted in [differential linear logic](#)...

Lots of this work have been motivated by the fact that coherent semantics is not complete as a semantics of programs (technically one says that it is not *fully abstract*). In order to see this, let us first come back on the origin of the central concept of *stability* which as pointed above originated in the study of the sequentiality in programs.

11.5.1 Sequentiality

Sequentiality is a property that we will not define here (it would deserve its own article). We rely on the intuition that a function of n arguments is sequential if one can determine which of these argument is examined first during the computation. Obviously any function implemented in a functional language is sequential; for example the function *or* defined à la CAML by:

```
or = fun (x, y) -> if x then true else y
```

examines its argument x first. Note that this may be expressed more abstractly by the property: $\text{or}(\perp, x) = \perp$ for any boolean x : the function *or* needs its first argument in order to compute anything. On the other hand we have $\text{or}(\text{true}, \perp) = \text{true}$: in some case (when the first argument is true), the function doesn't need its second argument at all.

The typical non sequential function is the *parallel or* (that one cannot define in a CAML like language).

For a while one may have believed that the stability condition on which coherent semantics is built was enough to capture the notion of *sequentiality* of programs. A hint was the already mentioned fact that the *parallel or* is not stable. This deserves a bit of explanation.

11.5.1.1 The parallel or is not stable

Let B be the coherent space of booleans, also know as the flat domain of booleans: $|B| = \{tt, ff\}$ where tt and ff are two arbitrary distinct objects (for example one may take $tt = 0$ and $ff = 1$) and for any $b_1, b_2 \in |B|$, define $b_1 \supset_B b_2$ iff $b_1 = b_2$. Then B has exactly three cliques: the empty clique that we shall denote \perp , the singleton $\{tt\}$ that we shall denote T and the singleton $\{ff\}$ that we shall denote F . These three cliques are ordered by inclusion: $\perp \leq T, F$ (we use \leq for \subset to enforce the idea that coherent spaces are domains).

Recall the [definition of the with](#), and in particular that any clique of $B \& B$ has the form $\langle x, y \rangle$ where x and y are cliques of B . Thus $B \& B$ has 9 cliques: $\langle \perp, \perp \rangle$, $\langle \perp, T \rangle$, $\langle \perp, F \rangle$, $\langle T, \perp \rangle$, ... that are ordered by the product order: $\langle x, y \rangle \leq \langle x', y' \rangle$ iff $x \leq x'$ and $y \leq y'$.

With these notations in mind one may define the parallel or by:

Por : $B \& B$	\longrightarrow	B
		$\langle T, \perp \rangle \longrightarrow T$
		$\langle \perp, T \rangle \longrightarrow T$
		$\langle F, F \rangle \longrightarrow F$

The function is completely determined if we add the assumption that it is non decreasing; for example one must have $\text{Por}\langle \perp, \perp \rangle = \perp$ because the lhs has to be less than both T and F (because $\langle \perp, \perp \rangle \leq \langle T, \perp \rangle$ and $\langle \perp, \perp \rangle \leq \langle F, F \rangle$).

The function is not stable because $\langle T, \perp \rangle \cap \langle \perp, T \rangle = \langle \perp, \perp \rangle$, thus $\text{Por}(\langle T, \perp \rangle \cap \langle \perp, T \rangle) = \perp$ whereas $\text{Por}\langle T, \perp \rangle \cap \text{Por}\langle \perp, T \rangle = T \cap T = T$.

Another way to see this is: suppose x and y are two cliques of B such that $tt \in \text{Por}\langle x, y \rangle$, which means that $\text{Por}\langle x, y \rangle = T$; according to the [characterisation theorem of stable functions](#), if Por were stable then there would be a unique minimum x_0 included in x , and a unique minimum

y_0 included in y such that $\text{Por}\langle x_0, y_0 \rangle = T$. This is not the case because both $\langle T, \perp \rangle$ and $\langle T, \perp \rangle$ are minimal such that their value is T .

In other terms, knowing that $\text{Por}\langle x, y \rangle = T$ doesn't tell which of x or y is responsible for that, although we know by the definition of Por that only one of them is. Indeed the Por function is not representable in sequential programming languages such as (typed) lambda-calculus.

So the first genuine idea would be that stability characterises sequentiality; but...

11.5.1.2 The Gustave function is stable

The Gustave function, so-called after an old joke, was found by Gérard Berry as an example of a function that is stable but non sequential. It is defined by:

$$\begin{aligned} B \& B \& B &\longrightarrow B \\ \langle T, F, \perp \rangle &\longrightarrow T \\ \langle \perp, T, F \rangle &\longrightarrow T \\ \langle F, \perp, T \rangle &\longrightarrow T \\ \langle x, y, z \rangle &\longrightarrow F \end{aligned}$$

The last clause is for all cliques x , y and z such that $\langle x, y, z \rangle$ is incompatible with the three cliques $\langle T, F, \perp \rangle$, $\langle \perp, T, F \rangle$ and $\langle F, \perp, T \rangle$, that is such that the union with any of these three cliques is not a clique in $B \& B \& B$. We shall denote x_1 , x_2 and x_3 these three cliques.

We furthermore assume that the Gustave function is non decreasing, so that we get $G\langle \perp, \perp, \perp \rangle = \perp$.

We note that x_1 , x_2 and x_3 are pairwise incompatible. From this we can deduce that the Gustave function is stable: typically if $G\langle x, y, z \rangle = T$ then exactly one of the x_i s is contained in $\langle x, y, z \rangle$.

However it is not sequential because there is no way to determine which of its three arguments is examined first: it is not the first one otherwise we would have $G\langle \perp, T, F \rangle = \perp$ and similarly it is not the second one nor the third one.

In other terms there is no way to implement the Gustave function by a lambda-term (or in any sequential programming language). Thus coherent semantics is not complete w.r.t. lambda-calculus.

The research for a right model for sequentiality was the motivation for lot of work, *e.g.*, *sequential algorithms* by Gérard B erry and Pierre-Louis Currien in the early eighties, that were more recently reformulated as a kind of [game model](#), and the theory of *hypercoherent spaces* by Antonio Bucciarelli and Thomas Ehrhard.

11.5.2 Multiplicative neutrals and the mix rule

Coherent semantics is slightly degenerated w.r.t. linear logic because it identifies multiplicative neutrals (it also identifies additive neutrals but that's yet another problem): the coherent spaces 1 and \perp are equal.

The first consequence of the identity $1 = \perp$ is that the formula $1 \multimap \perp$ becomes provable, and so does the formula \perp . Note that this doesn't entail (as in classical logic or intuitionistic logic) that linear logic is incoherent because the principle $\perp \multimap A$ for any formula A is still not provable.

The equality $1 = \perp$ has also as consequence the fact that $\perp \multimap 1$ (or equivalently the formula $1 \wp 1$) is provable. This principle is also known as the [mix rule](#)

$$\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta} \text{mix}$$

as it can be used to show that this rule is admissible:

$$\frac{\frac{\frac{\vdash \Gamma}{\vdash \Gamma, \perp} \perp_R \quad \frac{\vdash \Delta}{\vdash \Delta, \perp} \perp_R}{\vdash \Gamma, \Delta, \perp \otimes \perp} \quad \frac{}{\vdash 1 \wp 1}}{\vdash \Gamma, \Delta} \text{cut}$$

None of the two principles $1 \multimap \perp$ and $\perp \multimap 1$ are valid in linear logic. To correct this one could extend the syntax of linear logic by adding the mix-rule. This is not very satisfactory as the mix rule violates some principles of [Polarized linear logic](#), typically the fact that as sequent of the form $\vdash P_1, P_2$ where P_1 and P_2 are positive, is never provable.

On the other hand the mix-rule is valid in coherent semantics so one could try to find some other model that invalidates the mix-rule. For example Girard's Coherent Banach spaces were an attempt to address this issue.

Chapter 12

Phase semantics

12.1 Introduction

The semantics given by phase spaces is a kind of "formula and provability semantics", and is thus quite different in spirit from the more usual denotational semantics of linear logic. (Those are rather some "formulas and *proofs* semantics".)

- - - probably a whole lot more of blabla to put here... - - -

12.2 Preliminaries: relation and closure operators

Part of the structure obtained from phase semantics works in a very general framework and relies solely on the notion of relation between two sets.

12.2.1 Relations and operators on subsets

The starting point of phase semantics is the notion of *duality*. The structure needed to talk about duality is very simple: one just needs a relation R between two sets X and Y . Using standard mathematical practice, we can write either $(a, b) \in R$ or $a R b$ to say that $a \in X$ and $b \in Y$ are related.

Definition 23. If $R \subseteq X \times Y$ is a relation, we write $R^\sim \subseteq Y \times X$ for the converse relation: $(b, a) \in R^\sim$ iff $(a, b) \in R$.

Such a relation yields three interesting operators sending subsets of X to subsets of Y :

Definition 24. Let $R \subseteq X \times Y$ be a relation, define the operators $\langle R \rangle$, $[R]$ and $_{}^R$ taking subsets of X to subsets of Y as follows:

1. $b \in \langle R \rangle(x)$ iff $\exists a \in x, (a, b) \in R$
2. $b \in [R](x)$ iff $\forall a \in X, (a, b) \in R \implies a \in x$
3. $b \in x^R$ iff $\forall a \in x, (a, b) \in R$

The operator $\langle R \rangle$ is usually called the *direct image* of the relation, $[R]$ is sometimes called the *universal image* of the relation.

It is trivial to check that $\langle R \rangle$ and $[R]$ are covariant (increasing for the \subseteq relation) while $_{}^R$ is contravariant (decreasing for the \subseteq relation). More interesting:

Lemma 9 (Galois Connections).

1. $\langle R \rangle$ is right-adjoint to $[R^\sim]$: for any $x \subseteq X$ and $y \subseteq Y$, we have $[R^\sim]y \subseteq x$ iff $y \subseteq \langle R \rangle(x)$
2. we have $y \subseteq x^R$ iff $x \subseteq y^{R^\sim}$

This implies directly that $\langle R \rangle$ commutes with arbitrary unions and $[R]$ commutes with arbitrary intersections. (And in fact, any operator commuting with arbitrary unions (resp. intersections) is of the form $\langle R \rangle$ (resp. $[R]$).

Remark 2. The operator $_{}^R$ sends unions to intersections because $_{}^R : \mathcal{P}(X) \rightarrow \mathcal{P}(Y)^{\text{op}}$ is right adjoint to $_{}^{R^\sim} : \mathcal{P}(Y)^{\text{op}} \rightarrow \mathcal{P}(X)$...

12.2.2 Closure operators

Definition 25. A closure operator on $\mathcal{P}(X)$ is a monotonic operator P on the subsets of X which satisfies:

1. for all $x \subseteq X$, we have $x \subseteq P(x)$
2. for all $x \subseteq X$, we have $P(P(x)) \subseteq P(x)$

Closure operators are quite common in mathematics and computer science. They correspond exactly to the notion of *monad* on a preorder...

It follows directly from the definition that for any closure operator P , the image $P(x)$ is a fixed point of P . Moreover:

Lemma 10. $P(x)$ is the smallest fixed point of P containing x .

One other important property is the following:

Lemma 11. Write $\mathcal{F}(P) = \{x \mid P(x) \subseteq x\}$ for the collection of fixed points of a closure operator P . We have that $(\mathcal{F}(P), \cap)$ is a complete inf-lattice.

Remark 3. A closure operator is in fact determined by its set of fixed points: we have $P(x) = \bigcup \{y \mid y \in \mathcal{F}(P), y \subseteq x\}$

Since any complete inf-lattice is automatically a complete sup-lattice, $\mathcal{F}(P)$ is also a complete sup-lattice. However, the sup operation isn't given by plain union:

Lemma 12. If P is a closure operator on $\mathcal{P}(X)$, and if $(x_i)_{i \in I}$ is a (possibly infinite) family of subsets of X , we write $\bigvee_{i \in I} x_i = P(\bigcup_{i \in I} x_i)$.

We have $(\mathcal{F}(P), \cap, \bigvee)$ is a complete lattice.

Proof. easy. □

A rather direct consequence of the Galois connections of the previous section is:

Lemma 13. The operator and $\langle R \rangle \circ [R^\sim]$ and the operator $x \mapsto x^{R^{R^\sim}}$ are closures.

A last trivial lemma:

Lemma 14. We have $x^R = x^{R^{R^\sim R}}$.

As a consequence, a subset $x \subseteq X$ is in $\mathcal{F}(_{}^{R^{R^\sim}})$ iff it is of the form y^{R^\sim} .

Remark 4. Everything gets a little simpler when R is a symmetric relation on X .

12.3 Phase Semantics

12.3.1 Phase spaces

Definition 26 (monoid). A monoid is simply a set X equipped with a binary operation $_ \cdot _$ s.t.:

1. the operation is associative
2. there is a neutral element $1 \in X$

The monoid is commutative when the binary operation is commutative.

Definition 27 (Phase space). A phase space is given by:

1. a commutative monoid $(X, 1, \cdot)$,
2. together with a subset $\perp \subseteq X$.

The elements of X are called phases.

We write \perp for the relation $\{(a, b) \mid a \cdot b \in \perp\}$. This relation is symmetric.

A fact in a phase space is simply a fixed point for the closure operator $x \mapsto x^{\perp\perp}$.

Thanks to the preliminary work, we have:

Corollary 2. The set of facts of a phase space is a complete lattice where:

1. $\bigwedge_{i \in I} x_i$ is simply $\bigcap_{i \in I} x_i$,
2. $\bigvee_{i \in I} x_i$ is $(\bigcup_{i \in I} x_i)^{\perp\perp}$.

12.3.2 Additive connectives

The previous corollary makes the following definition correct:

Definition 28 (additive connectives). If $(X, 1, \cdot, \perp)$ is a phase space, we define the following facts and operations on facts:

1. $\top = X = \emptyset^{\perp}$
2. $0 = \emptyset^{\perp\perp} = X^{\perp}$
3. $x \& y = x \cap y$
4. $x \oplus y = (x \cup y)^{\perp\perp}$

Once again, the next lemma follows from previous observations:

Lemma 15 (additive de Morgan laws). We have

1. $0^{\perp} = \top$
2. $\top^{\perp} = 0$
3. $(x \& y)^{\perp} = x^{\perp} \oplus y^{\perp}$
4. $(x \oplus y)^{\perp} = x^{\perp} \& y^{\perp}$

12.3.3 Multiplicative connectives

In order to define the multiplicative connectives, we actually need to use the monoid structure of our phase space. One interpretation that is reminiscent in phase semantics is that our spaces are collections of *tests* / *programs* / *proofs* / *strategies* that can interact with each other. The result of the interaction between a and b is simply $a \cdot b$.

The set \perp can be thought of as the set of "good" things, and we thus have $a \in x^\perp$ iff " a interacts correctly with all the elements of x ".

Definition 29. *If x and y are two subsets of a phase space, we write $x \cdot y$ for the set $\{a \cdot b \mid a \in x, b \in y\}$.*

Thus $x \cdot y$ contains all the possible interactions between one element of x and one element of y .

The tensor connective of linear logic is now defined as:

Definition 30 (multiplicative connectives). *If x and y are facts in a phase space, we define*

- $1 = \{1\}^{\perp\perp}$;
- $\perp = 1^\perp$;
- the tensor $x \otimes y$ to be the fact $(x \cdot y)^{\perp\perp}$;
- the par connective is the de Morgan dual of the tensor: $x \wp y = (x^\perp \otimes y^\perp)^\perp$;
- the linear arrow is just $x \multimap y = x^\perp \wp y = (x \otimes y^\perp)^\perp$.

Note that by unfolding the definition of \multimap , we have the following, "intuitive" definition of $x \multimap y$:

Lemma 16. *If x and y are facts, we have $a \in x \multimap y$ iff $\forall b \in x, a \cdot b \in y$.*

Proof. easy exercise. □

Readers familiar with realisability will appreciate...

Remark 5. *Some people say that this idea of orthogonality was implicitly present in Tait's proof of strong normalisation. More recently, Jean-Louis Krivine and Alexandre Miquel have used the idea explicitly to do realisability...*

12.3.4 Properties

All the expected properties hold:

Lemma 17.

- The operations \otimes , \wp , \oplus and $\&$ are commutative and associative,
- They have respectively 1 , \perp , 0 and \top as neutral element,
- 0 is absorbing for \otimes ,
- \top is absorbing for \wp ,
- \otimes distributes over \oplus ,
- \wp distributes over $\&$.

12.3.5 Exponentials

Definition 31 (Exponentials). Write I for the set of idempotents of a phase space: $I = \{a \mid a \cdot a = a\}$. We put:

1. $!x = (x \cap I \cap 1)^{\perp\perp}$,
2. $?x = (x^\perp \cap I \cap 1)^\perp$.

This definition captures precisely the intuition behind the exponentials:

- we need to have contraction, hence we restrict to idempotents in x ,
- and weakening, hence we restrict to 1 .

Since I isn't necessarily a fact, we then take the biorthogonal to get a fact...

12.4 Soundness

Definition 32. Let $(X, 1, \cdot)$ be a commutative monoid.

Given a formula A of linear logic and an assignation ρ that associate a fact to any variable, we can inductively define the interpretation $\llbracket A \rrbracket_\rho$ of A in X as one would expect. Interpretation is lifted to sequents as $\llbracket A_1, \dots, A_n \rrbracket_\rho = \llbracket A_1 \rrbracket_\rho \wp \dots \wp \llbracket A_n \rrbracket_\rho$.

Theorem 31. Let Γ be a provable sequent in linear logic. Then $1_X \in \llbracket \Gamma \rrbracket$.

Proof. By induction on $\vdash \Gamma$. □

12.5 Completeness

Phase semantics is complete w.r.t. linear logic. In order to prove this, we need to build a particular commutative monoid.

Definition 33. We define the syntactic monoid as follows:

- Its elements are sequents Γ quotiented by the equivalence relation \cong generated by the rules:
 1. $\Gamma \cong \Delta$ if Γ is a permutation of Δ
 2. $?A, ?A, \Gamma \cong ?A, \Gamma$
- Product is concatenation: $\Gamma \cdot \Delta := \Gamma, \Delta$
- Neutral element is the empty sequent: $1 := \emptyset$.

The equivalence relation intuitively means that we do not care about the multiplicity of $?A$ -formulae.

Lemma 18. The syntactic monoid is indeed a commutative monoid.

Definition 34. The syntactic assignation is the assignation that sends any variable α to the fact $\{\alpha\}^\perp$.

We instantiate the pole as $\perp := \{\Gamma \mid \vdash \Gamma\}$.

Theorem 32. If $\Gamma \in \llbracket \Gamma \rrbracket^\perp$, then $\vdash \Gamma$.

Proof. By induction on Γ . □

12.6 Cut elimination

Actually, the completeness result is stronger, as the proof does not use the cut-rule in the reconstruction of $\vdash \Gamma$. By refining the pole as the set of *cut-free* provable formulae, we get:

Theorem 33. *If $\Gamma \in \llbracket \Gamma \rrbracket^\perp$, then Γ is cut-free provable.*

From soundness, one can retrieve the cut-elimination theorem.

Corollary 3. *Linear logic enjoys the cut-elimination property.*

12.7 The Rest

Chapter 13

Categorical semantics

Constructing denotational models of linear can be a tedious work. Categorical semantics are useful to identify the fundamental structure of these models, and thus simplify and make more abstract the elaboration of those models.

TODO: why categories? how to extract categorical models? etc.

See [Mac98] for a more detailed introduction to category theory. See [Mel09] for a detailed treatment of categorical semantics of linear logic.

13.1 Basic category theory recalled

Definition 35 (Category).

Definition 36 (Functor).

Definition 37 (Natural transformation).

Definition 38 (Adjunction).

Definition 39 (Monad).

13.2 Overview

In order to interpret the various [fragments](#) of linear logic, we define incrementally what structure we need in a categorical setting.

- The most basic underlying structure are **symmetric monoidal categories** which model the symmetric tensor \otimes and its unit 1 .
- The \otimes, \multimap fragment (**IMLL**) is captured by so-called **symmetric monoidal closed categories**.
- Upgrading to **ILL**, that is, adding the exponential $!$ modality to IMLL requires modelling it categorically. There are various ways to do so: using rich enough **adjunctions**, or with an ad-hoc definition of a well-behaved comonad which leads to **linear categories** and close relatives.
- Dealing with the additives $\&, \oplus$ is quite easy, as they are plain **cartesian product** and **coproduct**, usually defined through universal properties in category theory.
- Retrieving \wp, \perp and $?$ is just a matter of dualizing $\otimes, 1$ and $!$, thus requiring the model to be a ***-autonomous category** for that purpose.

13.3 Modeling **IMLL**

A model of **IMLL** is a *closed symmetric monoidal category*. We recall the definition of these categories below.

Definition 40 (Monoidal category). A monoidal category $(\mathcal{C}, \otimes, I, \alpha, \lambda, \rho)$ is a category \mathcal{C} equipped with

- a functor $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ called tensor product,
- an object I called unit object,
- three natural isomorphisms α , λ and ρ , called respectively associator, left unitor and right unitor, whose components are

$$\alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C) \quad \lambda_A : I \otimes A \rightarrow A \quad \rho_A : A \otimes I \rightarrow A$$

such that

- for every objects A, B, C, D in \mathcal{C} , the diagram

$$\begin{array}{ccc} ((A \otimes B) \otimes C) \otimes D & \xrightarrow{\alpha_{A,B,C \otimes D}} & (A \otimes (B \otimes C)) \otimes D \xrightarrow{\alpha_{A,B \otimes C,D}} A \otimes ((B \otimes C) \otimes D) \\ \alpha_{A \otimes B,C,D} \downarrow & & \downarrow A \otimes \alpha_{B,C,D} \\ (A \otimes B) \otimes (C \otimes D) & \xrightarrow{\alpha_{A,B,C \otimes D}} & A \otimes (B \otimes (C \otimes D)) \end{array}$$

commutes,

- for every objects A and B in \mathcal{C} , the diagram

$$\begin{array}{ccc} (A \otimes I) \otimes B & \xrightarrow{\alpha_{A,I,B}} & A \otimes (I \otimes B) \\ \rho_{A \otimes B} \searrow & & \swarrow A \otimes \lambda_B \\ & A \otimes B & \end{array}$$

commutes.

Definition 41 (Braided, symmetric monoidal category). A braided monoidal category is a category together with a natural isomorphism of components

$$\gamma_{A,B} : A \otimes B \rightarrow B \otimes A$$

called braiding, such that the two diagrams

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

and

$$\begin{array}{ccccc}
& & (A \otimes B) \otimes C & \xrightarrow{\gamma_{A \otimes B, C}} & C \otimes (A \otimes B) \\
& \nearrow^{\alpha_{A, B, C}^{-1}} & & & \searrow^{\alpha_{C, A, B}^{-1}} \\
A \otimes (B \otimes C) & & & & (C \otimes A) \otimes B \\
& \searrow_{A \otimes \gamma_{B, C}} & & & \nearrow_{\gamma_{A, C \otimes B}} \\
& & A \otimes (C \otimes B) & \xrightarrow{\alpha_{A, C, B}^{-1}} & (A \otimes C) \otimes B
\end{array}$$

commute for every objects A , B and C .

A symmetric monoidal category is a braided monoidal category in which the braiding satisfies

$$\gamma_{B, A} \circ \gamma_{A, B} = A \otimes B$$

for every objects A and B .

Definition 42 (Closed monoidal category). A monoidal category $(\mathcal{C}, \otimes, I)$ is left closed when for every object A , the functor $B \mapsto A \otimes B$ has a right adjoint, written $B \mapsto (A \multimap B)$. This means that there exists a bijection

$$\mathcal{C}(A \otimes B, C) \cong \mathcal{C}(B, A \multimap C)$$

which is natural in B and C . Equivalently, a monoidal category is left closed when it is equipped with a left closed structure, which consists of

- an object $A \multimap B$,
- a morphism $\text{eval}_{A, B} : A \otimes (A \multimap B) \rightarrow B$, called left evaluation,

for every objects A and B , such that for every morphism $f : A \otimes X \rightarrow B$ there exists a unique morphism $h : X \rightarrow A \multimap B$ making the diagram

$$\begin{array}{ccc}
A \otimes X & & \\
\downarrow A \otimes h & \searrow f & \\
A \otimes (A \multimap B) & \xrightarrow{\text{eval}_{A, B}} & B
\end{array}$$

commute.

Dually, the monoidal category \mathcal{C} is right closed when the functor $B \mapsto B \otimes A$ admits a right adjoint. The notion of right closed structure can be defined similarly.

In a symmetric monoidal category, a left closed structure induces a right closed structure and conversely, allowing us to simply speak of a closed symmetric monoidal category.

13.4 Modeling the additives

Definition 43 (Product). A product (X, π_1, π_2) of two coinital morphisms $f : A \rightarrow B$ and $g : A \rightarrow C$ in a category \mathcal{C} is an object X of \mathcal{C} together with two morphisms $\pi_1 : X \rightarrow B$ and $\pi_2 : X \rightarrow C$ such that there exists a unique morphism $h : A \rightarrow X$ making the diagram

$$\begin{array}{ccc}
& A & \\
& \downarrow h & \\
f & X & g \\
& \swarrow \pi_1 & \searrow \pi_2 \\
& B & C
\end{array}$$

commute.

A category has *finite products* when it has products and a terminal object.

Definition 44 (Monoid). A monoid (M, μ, η) in a monoidal category $(\mathcal{C}, \otimes, I)$ is an object M together with two morphisms $\mu : M \otimes M \rightarrow M$ and $\eta : I \rightarrow M$ such that the diagrams

$$\begin{array}{ccc}
 & (M \otimes M) \otimes M & \xrightarrow{\mu \otimes M} & M \otimes M \\
 & \swarrow \alpha_{M,M,M} & & \downarrow \mu \\
 M \otimes (M \otimes M) & & & \\
 \downarrow M \otimes \mu & & & \\
 M \otimes M & \xrightarrow{\mu} & & M
 \end{array}$$

and

$$\begin{array}{ccccc}
 I \otimes M & \xrightarrow{\eta \otimes M} & M \otimes M & \xleftarrow{M \otimes \eta} & M \otimes I \\
 & \searrow \lambda_M & \downarrow \mu & \swarrow \rho_M & \\
 & & M & &
 \end{array}$$

commute.

Property 1. Categories with products vs monoidal categories.

13.5 Modeling ILL

Introduced in [Ben94].

Definition 45 (Linear-non linear (LNL) adjunction). A linear-non linear adjunction is a symmetric monoidal adjunction between lax monoidal functors

$$(\mathcal{M}, \times, \top) \begin{array}{c} \xrightarrow{(L,l)} \\ \perp \\ \xleftarrow{(M,m)} \end{array} (\mathcal{L}, \otimes, I)$$

in which the category \mathcal{M} has finite products.

$$! = L \circ M$$

This section is devoted to defining the concepts necessary to define these adjunctions.

Definition 46 (Monoidal functor). A lax monoidal functor (F, f) between two monoidal categories $(\mathcal{C}, \otimes, I)$ and $(\mathcal{D}, \bullet, J)$ consists of

- a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ between the underlying categories,
- a natural transformation f of components $f_{A,B} : FA \bullet FB \rightarrow F(A \otimes B)$,
- a morphism $f : J \rightarrow FI$

such that the diagrams

$$\begin{array}{ccc}
(FA \bullet FB) \bullet FC \xrightarrow{\alpha^{FA, FB, FC}} FA \bullet (FB \bullet FC) & & \\
\phi_{A, B} \bullet FC \downarrow & \searrow^{FA \bullet \phi_{B, C}} & \\
F(A \otimes B) \bullet FC & & FA \bullet F(B \otimes C) \\
& \searrow^{\phi_{A \otimes B, C}} & \downarrow \phi_{A, B \otimes C} \\
& & F((A \otimes B) \otimes C) \xrightarrow{F\alpha_{A, B, C}} F(A \otimes (B \otimes C))
\end{array}$$

and

$$\begin{array}{ccc}
FA \bullet J \xrightarrow{FA \bullet \phi} FA \bullet FI & & J \bullet FB \xrightarrow{\phi \bullet FB} FI \bullet FB \\
\rho_{FA} \downarrow & \downarrow \phi_{A, I} & \lambda_{FB} \downarrow & \downarrow \phi_{I, B} \\
FA \xleftarrow{F\rho_A} F(A \otimes I) & \text{and} & FB \xleftarrow{F\lambda_B} F(I \otimes B)
\end{array}$$

commute for every objects A, B and C of \mathcal{C} . The morphisms $f_{A, B}$ and f are called coherence maps.

A lax monoidal functor is strong when the coherence maps are invertible and strict when they are identities.

Definition 47 (Monoidal natural transformation). Suppose that $(\mathcal{C}, \otimes, I)$ and $(\mathcal{D}, \bullet, J)$ are two monoidal categories and $(F, f) : (\mathcal{C}, \otimes, I) \Rightarrow (\mathcal{D}, \bullet, J)$ and $(G, g) : (\mathcal{C}, \otimes, I) \Rightarrow (\mathcal{D}, \bullet, J)$ are two monoidal functors between these categories. A monoidal natural transformation $\theta : (F, f) \rightarrow (G, g)$ between these monoidal functors is a natural transformation $\theta : F \Rightarrow G$ between the underlying functors such that the diagrams

$$\begin{array}{ccc}
FA \bullet FB \xrightarrow{\theta_A \bullet \theta_B} GA \bullet GB & & \\
f_{A, B} \downarrow & \downarrow g_{A, B} & \\
F(A \otimes B) \xrightarrow{\theta_{A \otimes B}} G(A \otimes B) & \text{and} & \begin{array}{ccc} & J & \\ f \swarrow & & \searrow g \\ FI & \xrightarrow{\theta_I} & GI \end{array}
\end{array}$$

commute for every objects A and B of \mathcal{D} .

Definition 48 (Monoidal adjunction). A monoidal adjunction between two monoidal functors $(F, f) : (\mathcal{C}, \otimes, I) \Rightarrow (\mathcal{D}, \bullet, J)$ and $(G, g) : (\mathcal{D}, \bullet, J) \Rightarrow (\mathcal{C}, \otimes, I)$ is an adjunction between the underlying functors F and G such that the unit and the counit $\eta : \mathcal{C} \Rightarrow G \circ F$ and $\varepsilon : F \circ G \Rightarrow \mathcal{D}$ induce monoidal natural transformations between the corresponding monoidal functors.

13.6 Modeling negation

13.6.1 *-autonomous categories

Definition 49 (*-autonomous category). Suppose that we are given a symmetric monoidal closed category $(\mathcal{C}, \otimes, I)$ and an object R of \mathcal{C} . For every object A , we define a morphism $\partial_A : A \rightarrow (A \multimap R) \multimap R$ as follows. By applying the bijection of the adjunction defining (left) closed monoidal categories to the identity morphism $\text{id}_{A \multimap R} : A \multimap R \rightarrow A \multimap R$, we get a morphism $A \otimes (A \multimap R) \rightarrow R$, and thus a morphism $(A \multimap R) \otimes A \rightarrow R$ by precomposing with the symmetry $\gamma_{A \multimap R, A}$. The morphism ∂_A is finally obtained by applying the bijection of the adjunction defining (left) closed monoidal categories to this morphism. The object R is called dualizing when the morphism ∂_A is a bijection for every object A of \mathcal{C} . A symmetric monoidal closed category is *-autonomous when it admits such a dualizing object.

13.6.2 Compact closed categories

Definition 50 (Dual objects). A dual object structure $(A, B, \eta, \varepsilon)$ in a monoidal category $(\mathcal{C}, \otimes, I)$ is a pair of objects A and B together with two morphisms $\eta : I \rightarrow B \otimes A$ and $\varepsilon : A \otimes B \rightarrow I$ such that the diagrams

$$\begin{array}{ccc}
 & A \otimes (B \otimes A) & \xrightarrow{\alpha_{A,B,A}^{-1}} & (A \otimes B) \otimes A \\
 & \nearrow^{A \otimes \eta} & & \searrow^{\varepsilon \otimes A} \\
 A \otimes I & & & I \otimes A \\
 \uparrow \rho_A^{-1} & & & \downarrow \lambda_A \\
 A & \xlongequal{\quad\quad\quad} & & A
 \end{array}$$

and

$$\begin{array}{ccc}
 & (B \otimes A) \otimes B & \xrightarrow{\alpha_{B,A,B}} & B \otimes (A \otimes B) \\
 & \nearrow^{\eta \otimes B} & & \searrow^{B \otimes \varepsilon} \\
 I \otimes B & & & B \otimes I \\
 \uparrow \lambda_B^{-1} & & & \downarrow \rho_B \\
 B & \xlongequal{\quad\quad\quad} & & B
 \end{array}$$

commute. The object A is called a left dual of B (and conversely B is a right dual of A).

Lemma 19. Two left (resp. right) duals of a same object B are necessarily isomorphic.

Definition 51 (Compact closed category). A symmetric monoidal category $(\mathcal{C}, \otimes, I)$ is compact closed when every object A has a right dual A^* . We write $\eta_A : I \rightarrow A^* \otimes A$ and $\varepsilon_A : A \otimes A^* \rightarrow I$ for the corresponding duality morphisms.

Lemma 20. In a compact closed category the left and right duals of an object A are isomorphic.

Property 2. A compact closed category \mathcal{C} is monoidal closed, with closure defined by $\mathcal{C}(A \otimes B, C) \cong \mathcal{C}(B, A^* \otimes C)$.

Proof. To every morphism $f : A \otimes B \rightarrow C$, we associate a morphism $\lceil f \rceil : B \rightarrow A^* \otimes C$ defined as

$$B \xrightarrow{\lambda_B^{-1}} I \otimes B \xrightarrow{\eta_A \otimes B} (A^* \otimes A) \otimes B \xrightarrow{\alpha_{A^*,A,B}} A^* \otimes (A \otimes B) \xrightarrow{A^* \otimes f} A^* \otimes C$$

and to every morphism $g : B \rightarrow A^* \otimes C$, we associate a morphism $\lfloor g \rfloor : A \otimes B \rightarrow C$ defined as

$$A \otimes B \xrightarrow{A \otimes g} A \otimes (A^* \otimes C) \xrightarrow{\alpha_{A,A^*,C}^{-1}} (A \otimes A^*) \otimes C \xrightarrow{\varepsilon_A \otimes C} I \otimes C \xrightarrow{\lambda_C} C$$

It is easy to show that $\lfloor \lceil f \rceil \rfloor = f$ and $\lceil \lfloor g \rfloor \rceil = g$ from which we deduce the required bijection. \square

Property 3. A compact closed category is a (degenerated) $*$ -autonomous category, with the obvious duality structure. In particular, $(A \otimes B)^* \cong A^* \otimes B^*$.

Remark 6. The above isomorphism does not hold in $*$ -autonomous categories in general. This means that models which are compact closed categories identify \otimes and \mathfrak{A} as well as 1 and \perp .

Proof. The dualizing object R is simply I^* .

For any A , the reverse isomorphism $\delta_A : (A \multimap R) \multimap R \rightarrow A$ is constructed as follows:

$$\mathcal{C}((A \multimap R) \multimap R, A) := \mathcal{C}((A \otimes I^{**}) \otimes I^{**}, A) \cong \mathcal{C}((A \otimes I) \otimes I, A) \cong \mathcal{C}(A, A)$$

Identity on A is taken as the canonical morphism required. \square

13.7 Other categorical models

13.7.1 Lafont categories

13.7.2 Seely categories

13.7.3 Linear categories

13.8 Properties of categorical models

13.8.1 The Kleisli category

Chapter 14

Relational semantics

This is the simplest denotational semantics of linear logic. It consists in interpreting a formula A as a set A^* and a proof π of A as a subset π^* of A^* .

14.1 The category of sets and relations

It is the category \mathbf{Rel} whose objects are sets, and such that $\mathbf{Rel}(X, Y) = \mathfrak{P}(X \times Y)$. Composition is the ordinary composition of relations: given $s \in \mathbf{Rel}(X, Y)$ and $t \in \mathbf{Rel}(Y, Z)$, one sets $t \circ s = \{(a, c) \in X \times Z ; \exists b \in Y (a, b) \in s \text{ and } (b, c) \in t\}$ and the identity morphism is the diagonal relation $\text{Id}_X = \{(a, a) ; a \in X\}$.

An isomorphism in the category \mathbf{Rel} is a relation which is a bijection, as easily checked.

14.1.1 Monoidal structure

The tensor product is the usual cartesian product of sets $X \otimes Y = X \times Y$ (which is not a cartesian product in the category \mathbf{Rel} in the categorical sense). It is a bifunctor: given $s_i \in \mathbf{Rel}(X_i, Y_i)$ (for $i = 1, 2$), one sets $s_1 \otimes s_2 = \{((a_1, a_2), (b_1, b_2)) ; (a_i, b_i) \in s_i \text{ for } i = 1, 2\}$. The unit of this tensor product is $1 = \{*\}$ where $*$ is an arbitrary element.

For defining a monoidal category, it is not sufficient to provide the definition of the tensor product functor \otimes and its unit 1 , one has also to provide natural isomorphisms $\lambda_X \in \mathbf{Rel}(1 \otimes X, X)$, $\rho_X \in \mathbf{Rel}(X \otimes 1, X)$ (left and right neutrality of 1 for \otimes) and $\alpha_{X, Y, Z} \in \mathbf{Rel}((X \otimes Y) \otimes Z, X \otimes (Y \otimes Z))$ (associativity of \otimes). All these isomorphisms have to satisfy a number of commutations. In the present case, they are defined in the obvious way.

This monoidal category $(\mathbf{Rel}, \otimes, 1, \lambda, \rho)$ is symmetric, meaning that it is endowed with an additional natural isomorphism $\sigma_{X, Y} \in \mathbf{Rel}(X \otimes Y, Y \otimes X)$, also subject to some commutations. Here, again, this isomorphism is defined in the obvious way (symmetry of the cartesian product). So, to be precise, the SMCC (symmetric monoidal closed category) \mathbf{Rel} is the tuple $(\mathbf{Rel}, \otimes, 1, \lambda, \rho, \alpha, \sigma)$, but we shall simply denote it as \mathbf{Rel} .

The SMCC \mathbf{Rel} is closed. This means that, given any object X of \mathbf{Rel} (a set), the functor $Z \mapsto Z \otimes X$ (from \mathbf{Rel} to \mathbf{Rel}) admits a right adjoint $Y \mapsto (X \multimap Y)$ (from \mathbf{Rel} to \mathbf{Rel}). In other words, for any objects X and Y , we are given an object $X \multimap Y$ and a morphism $\text{ev}_{X, Y} \in \mathbf{Rel}((X \multimap Y) \otimes X, Y)$ with the following universal property: for any morphism $s \in \mathbf{Rel}(Z \otimes X, Y)$, there is a unique morphism $\text{fun}(s) \in \mathbf{Rel}(Z, X \multimap Y)$ such that $\text{ev}_{X, Y} \circ (\text{fun}(s) \otimes \text{Id}_X) = s$.

The definition of all these data is quite simple in \mathbf{Rel} : $X \multimap Y = X \times Y$, $\text{ev}_{X, Y} = \{(((a, b), a), b) ; (a, b) \in X \times Y\}$ and $\text{fun}(s) = \{(c, (a, b)) ; ((c, a), b) \in s\}$.

Let $\perp = 1 = \{*\}$. Then we have $\text{ev} \circ \sigma : \mathbf{Rel}(X \otimes (X \multimap \perp), \perp)$ and hence $\eta_X = \text{fun}(\text{ev} \circ \sigma) \in \mathbf{Rel}(X, (X \multimap \perp) \multimap \perp)$. It is clear that $\eta = \{(a, ((a, *), *)) ; a \in X\}$ and hence η is a natural

isomorphism: one says that the SMCC \mathbf{Rel} is a *-autonomous category, with \perp as dualizing object.

14.1.2 Additives

Given a family $(X_i)_{i \in I}$, let $\&_{i \in I} X_i = \cup_{i \in I} \{i\} \times X_i$. Let $\pi_j \in \mathbf{Rel}(\&_{i \in I} X_i, X_j)$ given by $\pi_j = \{((j, a), a) ; a \in X_j\}$. Then $(\&_{i \in I} X_i, (\pi_i)_{i \in I})$ is the cartesian product of the X_i s in the category \mathbf{Rel} .

14.1.3 Exponentials

One defines $!X$ as the set of all finite multisets of elements of X . Given $s \in \mathbf{Rel}(X, Y)$, one defines $!s = \{([a_1, \dots, a_n], [b_1, \dots, b_n]) ; n \in \mathbb{N} \text{ and } \forall i (a_i, b_i) \in s\}$ where $[a_1, \dots, a_n]$ is the multiset containing a_1, \dots, a_n , taking multiplicities into account. This defines a functor $\mathbf{Rel} \rightarrow \mathbf{Rel}$, that we endow with a comonad structure as follows:

- the counit, called dereliction, is the natural transformation $\text{der}_X \in \mathbf{Rel}(!X, X)$, given by $\text{der}_X = \{([a], a) ; a \in X\}$
- the comultiplication, called digging, is the natural transformation $\text{digg}_X \in \mathbf{Rel}(!X, !!X)$, given by $\text{digg}_X = \{(m_1 + \dots + m_n, [m_1, \dots, m_n]) ; n \in \mathbb{N} \text{ and } m_1, \dots, m_n \in !X\}$

14.2 Interpretation of propositional linear logic (LL_0)

The structure described above gives rise to the following interpretation of formulas and proofs of linear logic.

For all propositional variable X , fix a set $|X|$. Then with each formula A , we associate a set $|A|$ as follows:

- $|A^\perp| = |A|$;
- $|A \otimes B| = |A \wp B| = |A| \times |B|$;
- $|A \& B| = |A \oplus B| = (\{1\} \times |A|) \cup (\{2\} \times |B|)$;
- $!|A| = |?A| = \mathfrak{M}_f(|A|)$.

We then interpret the proofs of LL_0 as follows: with each proof π of sequent $\vdash A_1, \dots, A_n$, we associate a subset $\llbracket \pi \rrbracket \subseteq |A_1| \times \dots \times |A_n|$.

- Identity group:

$$\left[\frac{}{\vdash A^\perp, A} \text{axiom} \right] = \{(a, a) ; a \in |A|\}$$

$$\left[\frac{\begin{array}{c} \vdots \pi \qquad \vdots \rho \\ \vdash \Gamma, A \quad \vdash \Delta, A^\perp \end{array}}{\vdash \Gamma, \Delta} \text{cut} \right] = \{(\gamma, \delta) ; \exists a \in |A|, (\gamma, a) \in \llbracket \pi \rrbracket \wedge (\delta, a) \in \llbracket \rho \rrbracket\}$$

- Multiplicative group:

$$\left[\frac{\begin{array}{c} \vdots \pi \qquad \vdots \rho \\ \vdash \Gamma, A \quad \vdash \Delta, B \end{array}}{\vdash \Gamma, \Delta, A \otimes B} \otimes \right] = \{(\gamma, \delta, a, b) ; (\gamma, a) \in \llbracket \pi \rrbracket \wedge (\delta, b) \in \llbracket \rho \rrbracket\}$$

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma, A, B} \right] \right] = \{ (\gamma, (a, b)) ; (\gamma, a, b) \in \llbracket \pi \rrbracket \}$$

$$\llbracket \overline{\vdash 1} \top \rrbracket = \{ * \}$$

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma} \right] \right] = \{ (\gamma, *) ; \gamma \in \llbracket \pi \rrbracket \}$$

- Additive group:

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma, A} \right] \right] = \{ (\gamma, (1, a)) ; (\gamma, a) \in \llbracket \pi \rrbracket \}$$

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma, B} \right] \right] = \{ (\gamma, (2, b)) ; (\gamma, b) \in \llbracket \pi \rrbracket \}$$

$$\left[\left[\frac{\vdots \pi \quad \vdots \rho}{\vdash \Gamma, A \quad \vdash \Gamma, B} \right] \right] = \{ (\gamma, (1, a)) ; (\gamma, a) \in \llbracket \pi \rrbracket \} \cup \{ (\gamma, (2, b)) ; (\gamma, b) \in \llbracket \rho \rrbracket \}$$

$$\llbracket \overline{\vdash \Gamma, \top} \top \rrbracket = \emptyset$$

- Exponential group:

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma, A} \right] \right] = \{ (\gamma, [a]) ; (\gamma, a) \in \llbracket \pi \rrbracket \}$$

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma} \right] \right] = \{ (\gamma, []) ; \gamma \in \llbracket \pi \rrbracket \}$$

$$\left[\left[\frac{\vdots \pi}{\vdash \Gamma, ?A, ?A} \right] \right] = \{ (\gamma, m + n) ; (\gamma, m, n) \in \llbracket \pi \rrbracket \}$$

$$\left[\left[\frac{\vdots \pi}{\vdash ?A_1, \dots, ?A_n, B} \right] \right] =$$

$$\left\{ \left(\sum_{i=1}^k m_1^i, \dots, \sum_{i=1}^k m_n^i, [b_1, \dots, b_k] \right) ; k \in \mathbb{N} \text{ and } \forall 1 \leq i \leq k, (m_1^i, \dots, m_n^i, b_i) \in \llbracket \pi \rrbracket \right\}$$

Theorem 34. *If proof π' is obtained from π by eliminating a cut, then $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$.*

Chapter 15

Finiteness semantics

The category **Fin** of finiteness spaces and finitary relations was introduced by Ehrhard, refining the [purely relational model of linear logic](#). A finiteness space is a set equipped with a finiteness structure, i.e. a particular set of subsets which are said to be finitary; and the model is such that the usual relational denotation of a proof in linear logic is always a finitary subset of its conclusion. By the usual co-Kleisli construction, this also provides a model of the simply typed lambda-calculus: the cartesian closed category **Fin**.

The main property of finiteness spaces is that the intersection of two finitary subsets of dual types is always finite. This feature allows to reformulate Girard's quantitative semantics in a standard algebraic setting, where morphisms interpreting typed λ -terms are analytic functions between the topological vector spaces generated by vectors with finitary supports. This provided the semantical foundations of Ehrhard-Regnier's differential λ -calculus and motivated the general study of a differential extension of linear logic.

It is worth noticing that finiteness spaces can accomodate typed λ -calculi only: for instance, the relational semantics of fixpoint combinators is never finitary. The whole point of the finiteness construction is actually to reject infinite computations. Indeed, from a logical point of view, computation is cut elimination: the finiteness structure ensures the intermediate sets involved in the relational interpretation of a cut are all finite. In that sense, the finitary semantics is intrinsically typed.

15.1 Finiteness spaces

The construction of finiteness spaces follows a well known pattern. It is given by the following notion of orthogonality: $a \perp a'$ iff $a \cap a'$ is finite. Then one unrolls [familiar definitions](#), as we do in the following paragraphs.

Let A be a set. Denote by $\mathfrak{P}(A)$ the powerset of A and by $\mathfrak{P}_f(A)$ the set of all finite subsets of A . Let $\mathfrak{F} \subseteq \mathfrak{P}(A)$ any set of subsets of A . We define the pre-dual of \mathfrak{F} in A as $\mathfrak{F}^{\perp A} = \{a' \subseteq A; \forall a \in \mathfrak{F}, a \cap a' \in \mathfrak{P}_f(A)\}$. In general we will omit the subscript in the pre-dual notation and just write \mathfrak{F}^{\perp} . For all $\mathfrak{F} \subseteq \mathfrak{P}(A)$, we have the following immediate properties: $\mathfrak{P}_f(A) \subseteq \mathfrak{F}^{\perp}$; $\mathfrak{F} \subseteq \mathfrak{F}^{\perp\perp}$; if $\mathfrak{G} \subseteq \mathfrak{F}$, $\mathfrak{F}^{\perp} \subseteq \mathfrak{G}^{\perp}$. By the last two, we get $\mathfrak{F}^{\perp} = \mathfrak{F}^{\perp\perp\perp}$. A finiteness structure on A is then a set \mathfrak{F} of subsets of A such that $\mathfrak{F}^{\perp\perp} = \mathfrak{F}$.

A finiteness space is a dependant pair $\mathcal{A} = (|\mathcal{A}|, \mathfrak{F}(\mathcal{A}))$ where $|\mathcal{A}|$ is the underlying set (the web of \mathcal{A}) and $\mathfrak{F}(\mathcal{A})$ is a finiteness structure on $|\mathcal{A}|$. We then write \mathcal{A}^{\perp} for the dual finiteness space: $|\mathcal{A}^{\perp}| = |\mathcal{A}|$ and $\mathfrak{F}(\mathcal{A}^{\perp}) = \mathfrak{F}(\mathcal{A})^{\perp}$. The elements of $\mathfrak{F}(\mathcal{A})$ are called the finitary subsets of \mathcal{A} .

Example 1. For all set A , $(A, \mathfrak{P}_f(A))$ is a finiteness space and $(A, \mathfrak{P}_f(A))^{\perp} = (A, \mathfrak{P}(A))$. In

particular, each finite set A is the web of exactly one finiteness space: $(A, \mathfrak{F}_f(A)) = (A, \mathfrak{F}(A))$. We introduce the following two: $0 = 0^\perp = (\emptyset, \{\emptyset\})$ and $1 = 1^\perp = (\{\emptyset\}, \{\emptyset, \{\emptyset\}\})$. We also introduce the finiteness space of natural numbers \mathcal{N} by: $|\mathcal{N}| = \mathbf{N}$ and $a \in \mathfrak{F}(\mathcal{N})$ iff a is finite. We write $\mathcal{O} = \{0\} \in \mathfrak{F}(\mathcal{N})$.

Notice that \mathfrak{F} is a finiteness structure iff it is of the form \mathfrak{G}^\perp . It follows that any finiteness structure \mathfrak{F} is downwards closed for inclusion, and closed under finite unions and arbitrary intersections. Notice however that \mathfrak{F} is not closed under directed unions in general: for all $k \in \mathbf{N}$, write $k\downarrow = \{j; j \leq k\} \in \mathfrak{F}(\mathcal{N})$; then $k\downarrow \subseteq k'\downarrow$ as soon as $k \leq k'$, but $\bigcup_{k \geq 0} k\downarrow = \mathbf{N} \notin \mathfrak{F}(\mathcal{N})$.

15.1.1 Multiplicatives

For all finiteness spaces \mathcal{A} and \mathcal{B} , we define $\mathcal{A} \otimes \mathcal{B}$ by $|\mathcal{A} \otimes \mathcal{B}| = |\mathcal{A}| \times |\mathcal{B}|$ and $\mathfrak{F}(\mathcal{A} \otimes \mathcal{B}) = \{a \times b; a \in \mathfrak{F}(\mathcal{A}), b \in \mathfrak{F}(\mathcal{B})\}^{\perp\perp}$. It can be shown that $\mathfrak{F}(\mathcal{A} \otimes \mathcal{B}) = \{c \subseteq |\mathcal{A}| \times |\mathcal{B}|; c|_l \in \mathfrak{F}(\mathcal{A}), c|_r \in \mathfrak{F}(\mathcal{B})\}$, where $c|_l$ and $c|_r$ are the obvious projections.

Let $f \subseteq A \times B$ be a relation from A to B , we write $f^\perp = \{(\beta, \alpha); (\alpha, \beta) \in f\}$. For all $a \subseteq A$, we set $f \cdot a = \{\beta \in B; \exists \alpha \in a, (\alpha, \beta) \in f\}$. If moreover $g \subseteq B \times C$, we define $g \bullet f = \{(\alpha, \gamma) \in A \times C; \exists \beta \in B, (\alpha, \beta) \in f \wedge (\beta, \gamma) \in g\}$. Then, setting $\mathcal{A} \multimap \mathcal{B} = (\mathcal{A} \otimes \mathcal{B}^\perp)^\perp$, $\mathfrak{F}(\mathcal{A} \multimap \mathcal{B}) \subseteq |\mathcal{A}| \times |\mathcal{B}|$ is characterized as follows:

$$\begin{aligned} f \in \mathfrak{F}(\mathcal{A} \multimap \mathcal{B}) &\iff \forall a \in \mathfrak{F}(\mathcal{A}), f \cdot a \in \mathfrak{F}(\mathcal{B}) \text{ and } \forall b \in \mathfrak{F}(\mathcal{B}^\perp), f^\perp \cdot b \in \mathfrak{F}(\mathcal{A}^\perp) \\ &\iff \forall a \in \mathfrak{F}(\mathcal{A}), f \cdot a \in \mathfrak{F}(\mathcal{B}) \text{ and } \forall \beta \in |\mathcal{B}|, f^\perp \cdot \{\beta\} \in \mathfrak{F}(\mathcal{A}^\perp) \\ &\iff \forall \alpha \in |\mathcal{A}|, f \cdot \{\alpha\} \in \mathfrak{F}(\mathcal{B}) \text{ and } \forall b \in \mathfrak{F}(\mathcal{B}^\perp), f^\perp \cdot b \in \mathfrak{F}(\mathcal{A}^\perp) \end{aligned}$$

The elements of $\mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$ are called finitary relations from \mathcal{A} to \mathcal{B} . By the previous characterization, the identity relation $\text{id}_{\mathcal{A}} = \{(\alpha, \alpha); \alpha \in |\mathcal{A}|\}$ is finitary, and the composition of two finitary relations is also finitary. One can thus define the category **Fin** of finiteness spaces and finitary relations: the objects of **Fin** are all finiteness spaces, and $\mathbf{Fin}(\mathcal{A}, \mathcal{B}) = \mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$. Equipped with the tensor product \otimes , **Fin** is symmetric monoidal, with unit 1; it is monoidal closed by the definition of \multimap ; it is $*$ -autonomous by the obvious isomorphism between \mathcal{A}^\perp and $\mathcal{A} \multimap 1$.

Example 2. Setting $\mathcal{S} = \{(k, k+1); k \in \mathbf{N}\}$ and $\mathcal{P} = \{(k+1, k); k \in \mathbf{N}\}$, we have $\mathcal{S}, \mathcal{P} \in \mathbf{Fin}(\mathcal{N}, \mathcal{N})$ and $\mathcal{P} \bullet \mathcal{S} = \text{id}_{\mathcal{N}}$.

15.1.2 Additives

We now introduce the cartesian structure of **Fin**. We define $\mathcal{A} \oplus \mathcal{B}$ by $|\mathcal{A} \oplus \mathcal{B}| = |\mathcal{A}| \uplus |\mathcal{B}|$ and $\mathfrak{F}(\mathcal{A} \oplus \mathcal{B}) = \{a \uplus b; a \in \mathfrak{F}(\mathcal{A}), b \in \mathfrak{F}(\mathcal{B})\}$ where \uplus denotes the disjoint union of sets: $x \uplus y = (\{1\} \times x) \cup (\{2\} \times y)$. We have $(\mathcal{A} \oplus \mathcal{B})^\perp = \mathcal{A}^\perp \oplus \mathcal{B}^\perp$.¹ The category **Fin** is both cartesian and co-cartesian, with \oplus being the product and co-product, and 0 the initial and terminal object. Projections are given by:

$$\begin{aligned} \lambda_{\mathcal{A}, \mathcal{B}} &= \{((1, \alpha), \alpha); \alpha \in |\mathcal{A}|\} \in \mathbf{Fin}(\mathcal{A} \oplus \mathcal{B}, \mathcal{A}) \\ \rho_{\mathcal{A}, \mathcal{B}} &= \{((2, \beta), \beta); \beta \in |\mathcal{B}|\} \in \mathbf{Fin}(\mathcal{A} \oplus \mathcal{B}, \mathcal{B}) \end{aligned}$$

¹The fact that the additive connectors are identified, i.e. that we obtain a biproduct, is to be related with the enrichment of **Fin** over the monoid structure of set union: see [Fio07]. This identification can also be shown to be an isomorphism of LL with sums of proofs.

and if $f \in \mathbf{Fin}(\mathcal{C}, \mathcal{A})$ and $g \in \mathbf{Fin}(\mathcal{C}, \mathcal{B})$, pairing is given by:

$$\langle f, g \rangle = \{(\gamma, (1, \alpha)); (\gamma, \alpha) \in f\} \cup \{(\gamma, (2, \beta)); (\gamma, \beta) \in g\} \in \mathbf{Fin}(\mathcal{C}, \mathcal{A} \oplus \mathcal{B}).$$

The unique morphism from \mathcal{A} to 0 is the empty relation. The co-cartesian structure is obtained symmetrically.

Example 3. Write $\mathcal{O}^\perp = \{(0, \emptyset)\} \in \mathbf{Fin}(\mathcal{N}, 1)$. Then $\langle \mathcal{O}^\perp, \mathcal{P} \rangle = \{(0, (1, \emptyset))\} \cup \{(k+1, (2, k)); k \in \mathbf{N}\} \in \mathbf{Fin}(\mathcal{N}, 1 \oplus \mathcal{N})$ is an isomorphism.

15.1.3 Exponentials

If A is a set, we denote by $\mathfrak{M}_f(A)$ the set of all finite multisets of elements of A , and if $a \subseteq A$, we write $a^! = \mathfrak{M}_f(a) \subseteq \mathfrak{M}_f(A)$. If $\bar{\alpha} \in \mathfrak{M}_f(A)$, we denote its support by $\text{Support}(\bar{\alpha}) \in \mathfrak{P}_f(A)$. For all finiteness space \mathcal{A} , we define $!\mathcal{A}$ by: $!|\mathcal{A}| = \mathfrak{M}_f(|\mathcal{A}|)$ and $\mathfrak{F}(!\mathcal{A}) = \{a^!; a \in \mathfrak{F}(\mathcal{A})\}^{\perp\perp}$. It can be shown that $\mathfrak{F}(!\mathcal{A}) = \{\bar{a} \subseteq \mathfrak{M}_f(|\mathcal{A}|); \bigcup_{\bar{\alpha} \in \bar{a}} \text{Support}(\bar{\alpha}) \in \mathfrak{F}(\mathcal{A})\}$. Then, for all $f \in \mathbf{Fin}(\mathcal{A}, \mathcal{B})$, we set

$$!f = \{([\alpha_1, \dots, \alpha_n], [\beta_1, \dots, \beta_n]); \forall i, (\alpha_i, \beta_i) \in f\} \in \mathbf{Fin}(!\mathcal{A}, !\mathcal{B}),$$

which defines a functor. Natural transformations $\text{der}_{\mathcal{A}} = \{([\alpha], \alpha); \alpha \in |\mathcal{A}|\} \in \mathbf{Fin}(!\mathcal{A}, \mathcal{A})$ and $\text{digg}_{\mathcal{A}} = \{(\sum_{i=1}^n \bar{\alpha}_i, [\bar{\alpha}_1, \dots, \bar{\alpha}_n]); \forall i, \bar{\alpha}_i \in !|\mathcal{A}|\}$ make this functor a comonad.

Example 4. We have isomorphisms $\{([\], \emptyset)\} \in \mathbf{Fin}(!0, 1)$ and $\{(\bar{\alpha}_l + \bar{\beta}_r, (\bar{\alpha}, \bar{\beta})); (\bar{\alpha}_l, \bar{\alpha}) \in !\lambda_{\mathcal{A}, \mathcal{B}} \wedge (\bar{\beta}_r, \bar{\beta}) \in !\rho_{\mathcal{A}, \mathcal{B}}\} \in \mathbf{Fin}(!(\mathcal{A} \oplus \mathcal{B}), !\mathcal{A} \otimes !\mathcal{B})$.
More generally, we have $!(\mathcal{A}_1 \oplus \dots \oplus \mathcal{A}_n) \cong !\mathcal{A}_1 \otimes \dots \otimes !\mathcal{A}_n$.

Chapter 16

Geometry of interaction

16.1 Introduction

The *geometry of interaction*, GoI in short, was defined in the early nineties by Girard as an interpretation of linear logic into operators algebra: formulae were interpreted by Hilbert spaces and proofs by partial isometries.

This was a striking novelty as it was the first time that a mathematical model of logic (lambda-calculus) didn't interpret a proof of $A \multimap B$ as a morphism *from* A *to* B and proof composition (cut rule) as the composition of morphisms. Rather the proof was interpreted as an operator acting *on* $A \multimap B$, that is a morphism from $A \multimap B$ to $A \multimap B$. For proof composition the problem was then, given an operator on $A \multimap B$ and another one on $B \multimap C$ to construct a new operator on $A \multimap C$. This problem was solved by the *execution formula* that bares some formal analogies with Kleene's formula for recursive functions. For this reason GoI was claimed to be an *operational semantics*, as opposed to traditionnal *denotational semantics*.

The first instance of the GoI was restricted to the *MELL* fragment of linear logic (Multiplicative and Exponential fragment) which is enough to encode lambda-calculus. Since then Girard proposed several improvements: firstly the extension to the additive connectives known as *Geometry of Interaction 3* and more recently a complete reformulation using Von Neumann algebras that allows to deal with some aspects of *implicit complexity*

The GoI has been a source of inspiration for various authors. Danos and Regnier have reformulated the original model exhibiting its combinatorial nature using a theory of reduction of paths in proof-nets and showing the link with abstract machines; the execution formula appears as the composition of two automata interacting through a common interface. Also the execution formula has rapidly been understood as expressing the composition of strategies in game semantics. It has been used in the theory of sharing reduction for lambda-calculus in the Abadi-Gonthier-Lévy reformulation and simplification of Lamping's representation of sharing. Finally the original GoI for the *MELL* fragment has been reformulated in the framework of traced monoidal categories following an idea originally proposed by Joyal.

16.1.1 The Geometry of Interaction as operators

The original construction of GoI by Girard follows a general pattern already mentionned in the section on *coherent semantics* under the name *symmetric reducibility* and that was first put to use in *phase semantics*. First set a general space P called the *proof space* because this is where the interpretations of proofs will live. Make sure that P is a (not necessarily commutative) monoid. In the case of GoI, the proof space is a subset of the space of bounded operators on ℓ^2 .

Second define a particular subset of P that will be denoted by \perp ; then derive a duality on P :

for $u, v \in P$, u and v are dual¹ iff $uv \in \perp$.

Such a duality defines an **orthogonality relation**, with the usual derived definitions and properties.

For the GoI, two dualities have proved to work; we will consider the first one: nilpotency, *ie*, \perp is the set of nilpotent operators in P . Let us explicit this: two operators u and v are dual if there is a nonnegative integer n such that $(uv)^n = 0$. This duality is symmetric: if uv is nilpotent then vu is nilpotent also.

Last define a *type* as a subset T of the proof space that is equal to its bidual: $T = T^{\perp\perp}$. This means that $u \in T$ iff for all operator $v \in T^\perp$, that is such that $u'v \in \perp$ for all $u' \in T$, we have $uv \in \perp$.

The real work² is now to interpret logical operations, that is to associate a type to each formula, an object to each proof and show the *adequacy lemma*: if u is the interpretation of a proof of the formula A then u belongs to the type associated to A .

16.1.1.1 Partial isometries

The first step is to build the proof space. This is constructed as a special set of partial isometries on a separable Hilbert space H which turns out to be generated by partial permutations on the canonical basis of H .

These so-called *p-isometries* enjoy some nice properties, the most important one being that a *p-isometry* is a sum of *p-isometries* iff all the terms of the sum have disjoint domains and disjoint codomains. As a consequence we get that a sum of *p-isometries* is null iff each term of the sum is null.

A second important property is that operators on H can be *externalized* using *p-isometries* into operators acting on $H \oplus H$, and conversely operators on $H \oplus H$ may be *internalized* into operators on H . This is widely used in the sequel.

16.1.1.2 The *-autonomous structure

The second step is to interpret the linear logic multiplicative operations, most importantly the cut rule.

Internalization/externalization is the key for this: typically the type $A \otimes B$ is interpreted by a set of *p-isometries* which are internalizations of operators acting on $H \oplus H$.

The (interpretation of) the cut-rule is defined in two steps: firstly we use nilpotency to define an operation corresponding to lambda-calculus application which given two *p-isometries* in respectively $A \multimap B$ and A produces an operator in B . From this we deduce the composition and finally obtain a structure of *-autonomous category, that is a model of multiplicative linear logic.

16.1.1.3 The exponentials

Finally we turn to define exponentials, that is connectives managing duplication. To do this we introduce an isomorphism (induced by a *p-isometry*) between H and $H \otimes H$: the first component of the tensor is intended to hold the address of the the copy whereas the second component contains the content of the copy.

We eventually get a quasi-model of full MELL; quasi in the sense that if we can construct *p-isometries* for usual structural operations in MELL (contraction, dereliction, digging), the

¹In modern terms one says that u and v are *polar*.

²The difficulty is to find the right duality that will make logical operations interpretable. General conditions that allow to achieve this have been formulated by Hyland and Schalk thanks to their theory of *double glueing*.

interpretation of linear logic proofs is not invariant w.r.t. cut elimination in general. It is however invariant in some good cases, which are enough to get a correction theorem for the interpretation.

16.1.2 The Geometry of Interaction as an abstract machine

16.2 GoI for MELL: partial isometries

16.2.1 Operators, partial isometries

We will denote by H the Hilbert space $\ell^2(\mathbb{N})$ of sequences $(x_n)_{n \in \mathbb{N}}$ of complex numbers such that the series $\sum_{n \in \mathbb{N}} |x_n|^2$ converges. If $x = (x_n)_{n \in \mathbb{N}}$ and $y = (y_n)_{n \in \mathbb{N}}$ are two vectors of H their *scalar product* is:

$$\langle x, y \rangle = \sum_{n \in \mathbb{N}} x_n \bar{y}_n.$$

Two vectors of H are *orthogonal* if their scalar product is nul. We will say that two subspaces are *disjoint* when any two vectors taken in each subspace are orthogonal. Note that this notion is different from the set theoretic one, in particular two disjoint subspaces always have exactly one vector in common: 0.

The *norm* of a vector is the square root of the scalar product with itself:

$$\|x\| = \sqrt{\langle x, x \rangle}.$$

Let us denote by $(e_k)_{k \in \mathbb{N}}$ the canonical *hilbertian basis* of H : $e_k = (\delta_{kn})_{n \in \mathbb{N}}$ where δ_{kn} is the Kronecker symbol: $\delta_{kn} = 1$ if $k = n$, 0 otherwise. Thus if $x = (x_n)_{n \in \mathbb{N}}$ is a sequence in H we have:

$$x = \sum_{n \in \mathbb{N}} x_n e_n.$$

An *operator* on H is a *continuous* linear map from H to H .³ The set of (bounded) operators is denoted by $\mathcal{B}(H)$.

The *range* or *codomain* of the operator u is the set of images of vectors; the *kernel* of u is the set of vectors that are annihilated by u ; the *domain* of u is the set of vectors orthogonal to the kernel, *ie*, the maximal subspace disjoint with the kernel:

- $\text{Codom}(u) = \{u(x), x \in H\}$;
- $\text{Ker}(u) = \{x \in H, u(x) = 0\}$;
- $\text{Dom}(u) = \{x \in H, \forall y \in \text{Ker}(u), \langle x, y \rangle = 0\}$.

These three sets are closed subspaces of H .

The *adjoint* of an operator u is the operator u^* defined by $\langle u(x), y \rangle = \langle x, u^*(y) \rangle$ for any $x, y \in H$. Adjointness is well behaved w.r.t. composition of operators:

$$(uv)^* = v^*u^*.$$

A *projector* is an idempotent operator of norm 0 (the projector on the null subspace) or 1, that is an operator p such that $p^2 = p$ and $\|p\| = 0$ or 1. A projector is auto-adjoint and its domain is equal to its codomain.

³Continuity is equivalent to the fact that operators are *bounded*, which means that one may define the *norm* of an operator u as the sup on the unit ball of the norms of its values:

$$\|u\| = \sup_{\{x \in H, \|x\|=1\}} \|u(x)\|.$$

A *partial isometry* is an operator u satisfying $uu^*u = u$; this condition entails that we also have $u^*uu^* = u^*$. As a consequence uu^* and u^*u are both projectors, called respectively the *initial* and the *final* projector of u because their (co)domains are respectively the domain and the codomain of u :

- $\text{Dom}(u^*u) = \text{Codom}(u^*u) = \text{Dom}(u)$;
- $\text{Dom}(uu^*) = \text{Codom}(uu^*) = \text{Codom}(u)$.

The restriction of u to its domain is an isometry. Projectors are particular examples of partial isometries.

If u is a partial isometry then u^* is also a partial isometry the domain of which is the codomain of u and the codomain of which is the domain of u .

If the domain of u is H that is if $u^*u = 1$ we say that u has *full domain*, and similarly for codomain. If u and v are two partial isometries then we have:

- $uv^* = 0$ iff $u^*uv^*v = 0$ iff the domains of u and v are disjoint;
- $u^*v = 0$ iff $uu^*vv^* = 0$ iff the codomains of u and v are disjoint;
- $uu^* + vv^* = 1$ iff the codomains of u and v are disjoint and their direct sum is H .

16.2.2 Partial permutations

We will now define our proof space which turns out to be the set of partial isometries acting as permutations on the canonical basis $(e_n)_{n \in \mathbb{N}}$.

More precisely a *partial permutation* φ on \mathbb{N} is a one-to-one map defined on a subset D_φ of \mathbb{N} onto a subset C_φ of \mathbb{N} . D_φ is called the *domain* of φ and C_φ its *codomain*. Partial permutations may be composed: if ψ is another partial permutation on \mathbb{N} then $\varphi \circ \psi$ is defined by:

- $n \in D_{\varphi \circ \psi}$ iff $n \in D_\psi$ and $\psi(n) \in D_\varphi$;
- if $n \in D_{\varphi \circ \psi}$ then $\varphi \circ \psi(n) = \varphi(\psi(n))$;
- the codomain of $\varphi \circ \psi$ is the image of the domain: $C_{\varphi \circ \psi} = \{\varphi(\psi(n)), n \in D_{\varphi \circ \psi}\}$.

Partial permutations are well known to form a structure of *inverse monoid* that we detail now.

Given a subset D of \mathbb{N} , the *partial identity* on D is the partial permutation φ defined by:

- $D_\varphi = D$;
- $\varphi(n) = n$ for any $n \in D_\varphi$.

Thus the codomain of φ is D .

The partial identity on D will be denoted by 1_D . Partial identities are idempotent for composition.

Among partial identities one finds the identity on the empty subset, that is the empty map, that we will denote by 0 and the identity on \mathbb{N} that we will denote by 1 . This latter permutation is the neutral for composition.

If φ is a partial permutation there is an inverse partial permutation φ^{-1} whose domain is $D_{\varphi^{-1}} = C_\varphi$ and who satisfies:

$$\begin{aligned}\varphi^{-1} \circ \varphi &= 1_{D_\varphi} \\ \varphi \circ \varphi^{-1} &= 1_{C_\varphi}\end{aligned}$$

16.2.3 The proof space

Given a partial permutation φ one defines a partial isometry u_φ by:

$$u_\varphi(e_n) = \begin{cases} e_{\varphi(n)} & \text{if } n \in D_\varphi \\ 0 & \text{otherwise} \end{cases}$$

In other terms if $x = (x_n)_{n \in \mathbb{N}}$ is a sequence in ℓ^2 then $u_\varphi(x)$ is the sequence $(y_n)_{n \in \mathbb{N}}$ defined by:

$$y_n = x_{\varphi^{-1}(n)} \text{ if } n \in C_\varphi, 0 \text{ otherwise.}$$

We will (not so abusively) write $e_{\varphi(n)} = 0$ when $\varphi(n)$ is undefined so that the definition of u_φ reads:

$$u_\varphi(e_n) = e_{\varphi(n)}.$$

The domain of u_φ is the subspace spanned by the family $(e_n)_{n \in D_\varphi}$ and the codomain of u_φ is the subspace spanned by $(e_n)_{n \in C_\varphi}$. In particular if φ is 1_D then u_φ is the projector on the subspace spanned by $(e_n)_{n \in D}$.

Definition 52. We call p -isometry a partial isometry of the form u_φ where φ is a partial permutation on \mathbb{N} . The proof space \mathcal{P} is the set of all p -isometries.

Proposition 8. Let φ and ψ be two partial permutations. We have:

$$u_\varphi u_\psi = u_{\varphi \circ \psi}.$$

The adjoint of u_φ is:

$$u_\varphi^* = u_{\varphi^{-1}}.$$

In particular the initial projector of u_φ is given by:

$$u_\varphi u_\varphi^* = u_{1_{D_\varphi}}.$$

and the final projector of u_φ is:

$$u_\varphi^* u_\varphi = u_{1_{C_\varphi}}.$$

If p is a projector in \mathcal{P} then there is a partial identity 1_D such that $p = u_{1_D}$.

Projectors commute, in particular we have:

$$u_\varphi u_\varphi^* u_\psi u_\psi^* = u_\psi u_\psi^* u_\varphi u_\varphi^*.$$

Note that this entails all the other commutations of projectors: $u_\varphi^* u_\varphi u_\psi u_\psi^* = u_\psi u_\psi^* u_\varphi^* u_\varphi$ and $u_\varphi^* u_\varphi u_\psi^* u_\psi = u_\psi^* u_\psi u_\varphi^* u_\varphi$.

In particular note that 0 is a p -isometry. The set \mathcal{P} is a submonoid of $\mathcal{B}(H)$ but it is not a subalgebra.⁴ In general given $u, v \in \mathcal{P}$ we don't necessarily have $u + v \in \mathcal{P}$. However we have:

Proposition 9. Let $u, v \in \mathcal{P}$. Then $u + v \in \mathcal{P}$ iff u and v have disjoint domains and disjoint codomains, that is:

$$u + v \in \mathcal{P} \text{ iff } uu^*vv^* = u^*uv^*v = 0.$$

Proof. Suppose for contradiction that e_n is in the domains of u and v . There are integers p and q such that $u(e_n) = e_p$ and $v(e_n) = e_q$ thus $(u + v)(e_n) = e_p + e_q$ which is not a basis vector; therefore $u + v$ is not a p -permutation. \square

As a corollary note that if $u + v = 0$ then $u = v = 0$.

⁴ \mathcal{P} is the normalizing groupoid of the maximal commutative subalgebra of $\mathcal{B}(H)$ consisting of all operators diagonalizable in the canonical basis.

16.2.4 From operators to matrices: internalization/externalization

It will be convenient to view operators on H as acting on $H \oplus H$, and conversely. For this purpose we define an isomorphism $H \oplus H \cong H$ by $x \oplus y \rightsquigarrow p(x) + q(y)$ where $p : H \mapsto H$ and $q : H \mapsto H$ are partial isometries given by:

$$\begin{aligned} p(e_n) &= e_{2n} \\ q(e_n) &= e_{2n+1}. \end{aligned}$$

From the definition p and q have full domain, that is satisfy $p^*p = q^*q = 1$. On the other hand their codomains are disjoint, thus we have $p^*q = q^*p = 0$. As the sum of their codomains is the full space H we also have $pp^* + qq^* = 1$.

Note that we have chosen p and q in \mathcal{P} . However the choice is arbitrary: any two p -isometries with full domain and disjoint codomains would do the job.

Given an operator u on H we may *externalize* it obtaining an operator U on $H \oplus H$ defined by the matrix:

$$U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$$

where the u_{ij} 's are given by:

$$\begin{aligned} u_{11} &= p^*up \\ u_{12} &= p^*uq \\ u_{21} &= q^*up \\ u_{22} &= q^*uq. \end{aligned}$$

The u_{ij} 's are called the *external components* of u . The externalization is functorial in the sense that if v is another operator externalized as:

$$V = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} = \begin{pmatrix} p^*vp & p^*vq \\ q^*vp & q^*vq \end{pmatrix}$$

then the externalization of uv is the matrix product UV .

As $pp^* + qq^* = 1$ we have:

$$u = (pp^* + qq^*)u(pp^* + qq^*) = pu_{11}p^* + pu_{12}q^* + qu_{21}p^* + qu_{22}q^*$$

which entails that externalization is reversible, its converse being called *internalization*.

If we suppose that u is a p -isometry then so are the components u_{ij} 's. Thus the formula above entails that the four terms of the sum have pairwise disjoint domains and pairwise disjoint codomains from which we deduce:

Proposition 10. *If u is a p -isometry and u_{ij} are its external components then:*

- u_{1j} and u_{2j} have disjoint domains, that is $u_{1j}^*u_{1j}u_{2j}^*u_{2j} = 0$ for $j = 1, 2$;
- u_{i1} and u_{i2} have disjoint codomains, that is $u_{i1}u_{i1}^*u_{i2}u_{i2}^* = 0$ for $i = 1, 2$.

As an example of computation in \mathcal{P} let us check that the product of the final projectors of $pu_{11}p^*$ and $pu_{12}q^*$ is null:

$$\begin{aligned} (pu_{11}p^*)(pu_{11}p^*)(pu_{12}q^*)(qu_{12}p^*) &= pu_{11}u_{11}^*u_{12}u_{12}^*p^* \\ &= pp^*upp^*u^*pp^*uqq^*u^*pp^* \\ &= pp^*u(pp^*)(u^*pp^*u)qq^*u^*pp^* \\ &= pp^*u(u^*pp^*u)(pp^*)qq^*u^*pp^* \\ &= pp^*uu^*pp^*u(pp^*)(qq^*)u^*pp^* \\ &= 0 \end{aligned}$$

where we used the fact that all projectors in \mathcal{P} commute, which is in particular the case of pp^* and u^*pp^*u .

16.3 GoI for MELL: the *-autonomous structure

Recall that when u and v are p -isometries we say they are dual when uv is nilpotent, and that \perp denotes the set of nilpotent operators. A *type* is a subset of \mathcal{P} that is equal to its bidual. In particular X^\perp is a type for any $X \subset \mathcal{P}$. We say that X *generates* the type $X^{\perp\perp}$.

16.3.1 The tensor and the linear application

If u and v are two p -isometries summing them doesn't in general produces a p -isometry. However as pup^* and qvq^* have disjoint domains and disjoint codomains it is true that $pup^* + qvq^*$ is a p -isometry. Given two types A and B , we thus define their *tensor* by:

$$A \otimes B = \{pup^* + qvq^*, u \in A, v \in B\}^{\perp\perp}$$

Note the closure by bidual to make sure that we obtain a type.

From what precedes we see that $A \otimes B$ is generated by the internalizations of operators on $H \oplus H$ of the form:

$$\begin{pmatrix} u & 0 \\ 0 & v \end{pmatrix}$$

Remark 7. *This so-called tensor resembles a sum rather than a product. We will stick to this terminology though because it defines the interpretation of the tensor connective of linear logic.*

The linear implication is derived from the tensor by duality: given two types A and B the type $A \multimap B$ is defined by:

$$A \multimap B = (A \otimes B^\perp)^\perp.$$

Unfolding this definition we get:

$$A \multimap B = \{u \in \mathcal{P} \text{ s.t. } \forall v \in A, \forall w \in B^\perp, u.(pvp^* + qwq^*) \in \perp\}.$$

16.3.2 The identity

Given a type A we are to find an operator ι in type $A \multimap A$, thus satisfying:

$$\forall u \in A, v \in A^\perp, \iota(pup^* + qvq^*) \in \perp.$$

An easy solution is to take $\iota = pq^* + qp^*$. In this way we get $\iota(pup^* + qvq^*) = qup^* + pvq^*$. Therefore $(\iota(pup^* + qvq^*))^2 = quvq^* + pvup^*$, from which one deduces that this operator is nilpotent iff uv is nilpotent. It is the case since u is in A and v in A^\perp .

It is interesting to note that the ι thus defined is actually the internalization of the operator on $H \oplus H$ given by the matrix:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

We will see once the composition is defined that the ι operator is the interpretation of the identity proof, as expected.

16.3.3 The execution formula, version 1: application

Definition 53. Let u and v be two operators; as above denote by u_{ij} the external components of u . If $u_{11}v$ is nilpotent we define the application of u to v by:

$$\text{App}(u, v) = u_{22} + u_{21}v \sum_k (u_{11}v)^k u_{12}.$$

Note that the hypothesis that $u_{11}v$ is nilpotent entails that the sum $\sum_k (u_{11}v)^k$ is actually finite. It would be enough to assume that this sum converges. For simplicity we stick to the nilpotency condition, but we should mention that weak nilpotency would do as well.

Theorem 35. If u and v are p -isometries such that $u_{11}v$ is nilpotent, then $\text{App}(u, v)$ is also a p -isometry.

Proof. Let us note $E_k = u_{21}v(u_{11}v)^k u_{12}$. Recall that u_{22} and u_{12} being external components of the p -isometry u , they have disjoint domains. Thus it is also the case of u_{22} and E_k . Similarly u_{22} and E_k have disjoint codomains because u_{22} and u_{21} have disjoint codomains.

Let now k and l be two integers such that $k > l$ and let us compute for example the intersection of the codomains of E_k and E_l :

$$E_k E_k^* E_l E_l^* = (u_{21}v(u_{11}v)^k u_{12})(u_{12}^*(v^* u_{11}^*)^k v^* u_{21}^*)(u_{21}v(u_{11}v)^l u_{12})(u_{12}^*(v^* u_{11}^*)^l v^* u_{21}^*)$$

As $k > l$ we may write $(v^* u_{11}^*)^l = (v^* u_{11}^*)^{k-l-1} v^* u_{11}^* (v^* u_{11}^*)^l$. Let us note $E = u_{11}^* (v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l u_{12}$ so that $E_k E_k^* E_l E_l^* = u_{21}v(u_{11}v)^k u_{12} u_{12}^* (v^* u_{11}^*)^{k-l-1} v^* E u_{12}^* (v^* u_{11}^*)^l v^* u_{21}^*$. We have:

$$\begin{aligned} E &= u_{11}^* (v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l u_{12} \\ &= (u_{11}^* u_{11} u_{11}^*) (v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l u_{12} \\ &= u_{11}^* (u_{11} u_{11}^*) ((v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l) u_{12} \\ &= u_{11}^* ((v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l) (u_{11} u_{11}^*) u_{12} \\ &= u_{11}^* (v^* u_{11}^*)^l v^* u_{21}^* u_{21}v(u_{11}v)^l u_{11} u_{11}^* u_{12} \\ &= 0 \end{aligned}$$

because u_{11} and u_{12} have disjoint codomains, thus $u_{11}^* u_{12} = 0$.

Similarly we can show that E_k and E_l have disjoint domains. Therefore we have proved that all terms of the sum $\text{App}(u, v)$ have disjoint domains and disjoint codomains. Consequently $\text{App}(u, v)$ is a p -isometry. \square

Theorem 36. Let A and B be two types and u a p -isometry. Then the two following conditions are equivalent:

1. $u \in A \multimap B$;
2. for any $v \in A$ we have:
 - $u_{11}v$ is nilpotent and
 - $\text{App}(u, v) \in B$.

Proof. Let v and w be two p -isometries. If we compute

$$(u.(pvp^* + qwq^*))^n = ((pu_{11}p^* + pu_{12}q^* + qu_{21}p^* + qu_{22}q^*)(pvp^* + qwq^*))^n$$

we get a finite sum of monomial operators of the form:

1. $p(u_{11}v)^{i_0}u_{12}w(u_{22}w)^{i_1} \dots u_{21}v(u_{11}v)^{i_m}p^*$
2. $p(u_{11}v)^{i_0}u_{12}w(u_{22}w)^{i_1} \dots u_{12}w(u_{22}w)^{i_m}q^*$,
3. $q(u_{22}w)^{i_0}u_{21}v(u_{11}v)^{i_1} \dots u_{21}v(u_{11}v)^{i_m}p^*$ or
4. $q(u_{22}w)^{i_0}u_{21}v(u_{11}v)^{i_1} \dots u_{12}w(u_{22}w)^{i_m}q^*$,

for all tuples of (nonnegative) integers (i_1, \dots, i_m) such that $i_0 + \dots + i_m + m = n$.

Each of these monomial is a p -isometry. Furthermore they have disjoint domains and disjoint codomains because their sum is the p -isometry $(u.(pvp^* + qwq^*))^n$. This entails that $(u.(pvp^* + qwq^*))^n = 0$ iff all these monomials are null.

Suppose $u_{11}v$ is nilpotent and consider:

$$(\text{App}(u, v)w)^n = \left((u_{22} + u_{21}v \sum_k (u_{11}v)^k u_{12})w \right)^n.$$

Developping we get a finite sum of monomials of the form:

$$5. (u_{22}w)^{l_0}u_{21}v(u_{11}v)^{k_1}u_{12}w(u_{22}w)^{l_1} \dots u_{21}v(u_{11}v)^{k_m}u_{12}w(u_{22}w)^{l_m}$$

for all tuples $(l_0, k_1, l_1, \dots, k_m, l_m)$ such that $l_0 + \dots + l_m + m = n$ and k_i is less than the degree of nilpotency of $u_{11}v$ for all i .

Again as these monomials are p -isometries and their sum is the p -isometry $(\text{App}(u, v)w)^n$, they have pairwise disjoint domains and pairwise disjoint codomains. Note that each of these monomial is equal to q^*Mq where M is a monomial of type 4 above.

As before we thus have that $(\text{App}(u, v)w)^n = 0$ iff all monomials of type 5 are null.

Suppose now that $u \in A \dashv\vdash B$ and $v \in A$. Then, since $0 \in B^\perp$ (0 belongs to any type) $u.(pvp^*) = pu_{11}vp^*$ is nilpotent, thus $u_{11}v$ is nilpotent.

Suppose further that $w \in B^\perp$. Then $u.(pvp^* + qwq^*)$ is nilpotent, thus there is a N such that $(u.(pvp^* + qwq^*))^n = 0$ for any $n \geq N$. This entails that all monomials of type 1 to 4 are null. Therefore all monomials appearing in the developpment of $(\text{App}(u, v)w)^N$ are null which proves that $\text{App}(u, v)w$ is nilpotent. Thus $\text{App}(u, v) \in B$.

Conversely suppose for any $v \in A$ and $w \in B^\perp$, $u_{11}v$ and $\text{App}(u, v)w$ are nilpotent. Let P and N be their respective degrees of nilpotency and put $n = N(P + 1) + N$. Then we claim that all monomials of type 1 to 4 appearing in the development of $(u.(pvp^* + qwq^*))^n$ are null.

Consider for example a monomial of type 1:

$$p(u_{11}v)^{i_0}u_{12}w(u_{22}w)^{i_1} \dots u_{21}v(u_{11}v)^{i_m}p^*$$

with $i_0 + \dots + i_m + m = n$. Note that m must be even.

If $i_{2k} \geq P$ for some $0 \leq k \leq m/2$ then $(u_{11}v)^{i_{2k}} = 0$ thus our monomial is null. Otherwise if $i_{2k} < P$ for all k we have:

$$i_1 + i_3 + \dots + i_{m-1} + m/2 = n - m/2 - (i_0 + i_2 + \dots + i_m)$$

thus:

$$i_1 + i_3 + \dots + i_{m-1} + m/2 \geq n - m/2 - (1 + m/2)P.$$

Now if $m/2 \geq N$ then $i_1 + \dots + i_{m-1} + m/2 \geq N$. Otherwise $1 + m/2 \leq N$ thus

$$i_1 + i_3 + \dots + i_{m-1} + m/2 \geq n - N - NP = N.$$

Since N is the degree of nilpotency of $\text{App}(u, v)w$ we have that the monomial:

$$(u_{22}w)^{i_1}u_{21}v(u_{11}v)^{i_2}u_{12}w \dots (u_{11}v)^{i_{m-2}}u_{12}w(u_{22}w)^{i_{m-1}}$$

is null, thus also the monomial of type 1 we started with. □

Corollary 4. *If A and B are types then we have:*

$$A \multimap B = \{u \in \mathcal{P} \text{ such that } \forall v \in A : u_{11}v \in \perp \text{ and } \text{App}(u, v) \in B\}.$$

As an example if we compute the application of the interpretation of the identity ι in type $A \multimap A$ to the operator $v \in A$ then we have:

$$\text{App}(\iota, v) = \iota_{22} + \iota_{21}v \sum (\iota_{11}v)^k \iota_{12}.$$

Now recall that $\iota = pq^* + qp^*$ so that $\iota_{11} = \iota_{22} = 0$ and $\iota_{12} = \iota_{21} = 1$ and we thus get:

$$\text{App}(\iota, v) = v$$

as expected.

16.3.4 The tensor rule

Let now A, A', B and B' be types and consider two operators u and u' respectively in $A \multimap B$ and $A \multimap B'$. We define an operator $u \otimes u'$ by:

$$\begin{aligned} u \otimes u' &= ppp^*upp^*p^* + qpq^*upp^*p^* + ppp^*uqp^*q^* + qpq^*uqp^*q^* \\ &+ pqp^*u'pq^*p^* + qq^*u'pq^*p^* + pqp^*u'qq^*q^* + qq^*u'qq^*q^* \end{aligned}$$

Once again the notation is motivated by linear logic syntax and is contradictory with linear algebra practice since what we denote by $u \otimes u'$ actually is the internalization of the direct sum $u \oplus u'$.

Indeed if we think of u and u' as the internalizations of the matrices:

$$\begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} u'_{11} & u'_{12} \\ u'_{21} & u'_{22} \end{pmatrix}$$

then we may write:

$$\begin{aligned} u \otimes u' &= ppu_{11}p^*p^* + qpu_{21}p^*p^* + ppu_{12}p^*q^* + qpu_{22}p^*q^* \\ &+ pqu'_{11}q^*p^* + qq'u'_{21}q^*p^* + pqu'_{12}q^*q^* + qq'u'_{22}q^*q^* \end{aligned}$$

Thus the components of $u \otimes u'$ are given by:

$$(u \otimes u')_{ij} = pu_{ij}p^* + qu'_{ij}q^*.$$

and we see that $u \otimes u'$ is actually the internalization of the matrix:

$$\begin{pmatrix} u_{11} & 0 & u_{12} & 0 \\ 0 & u'_{11} & 0 & u'_{12} \\ u_{21} & 0 & u_{22} & 0 \\ 0 & u'_{21} & 0 & u'_{22} \end{pmatrix}$$

We are now to show that if we suppose u and u' are in types $A \multimap B$ and $A' \multimap B'$, then $u \otimes u'$ is in $A \otimes A' \multimap B \otimes B'$. For this we consider v and v' respectively in A and A' , so that $pv^*p^* + qv'q^*$ is in $A \otimes A'$, and we show that $\text{App}(u \otimes u', pv^*p^* + qv'q^*) \in B \otimes B'$.

Since u and u' are in $A \multimap B$ and $A' \multimap B'$ we have that $u_{11}v$ and $u'_{11}v'$ are nilpotent and that $\text{App}(u, v)$ and $\text{App}(u', v')$ are respectively in B and B' , thus:

$$p\text{App}(u, v)p^* + q\text{App}(u', v')q^* \in B \otimes B'.$$

But we have:

$$\begin{aligned} ((u \otimes u')_{11}(pvp^* + qv'q^*))^n &= ((pu_{11}p^* + qu'_{11}q^*)(pvp^* + qv'q^*))^n \\ &= (pu_{11}vp^* + qu'_{11}v'q^*)^n \\ &= p(u_{11}v)^n p^* + q(u'_{11}v')^n q^* \end{aligned}$$

Therefore $(u \otimes u')_{11}(pvp^* + qv'q^*)$ is nilpotent. So we can compute $\text{App}(u \otimes u', pvp^* + qv'q^*)$:

$$\begin{aligned} \text{App}(u \otimes u', pvp^* + qv'q^*) &= (u \otimes u')_{22} + (u \otimes u')_{21}(pvp^* + qv'q^*) \sum ((u \otimes u')_{11}(pvp^* + qv'q^*))^k (u \otimes u')_{12} \\ &= pu_{22}p^* + qu'_{22}q^* + (pu_{21}p^* + qu'_{21}q^*)(pvp^* + qv'q^*) \sum ((pu_{11}p^* + qu'_{11}q^*)(pvp^* + qv'q^*))^k (pu_{12}p^* + qu'_{12}q^*) \\ &= p(u_{22} + u_{21}v \sum (u_{11}v)^k u_{12})p^* + q(u'_{22} + u'_{21}v' \sum (u'_{11}v')^k u'_{12})q^* \\ &= p\text{App}(u, v)p^* + q\text{App}(u', v')q^* \end{aligned}$$

thus lives in $B \otimes B'$.

16.3.5 Other monoidal constructions

16.3.5.1 Contraposition

Let A and B be some types; we have:

$$A \multimap B = A^\perp \multimap B^\perp$$

Indeed, $u \in A \multimap B$ means that for any v and w in respectively A and B^\perp we have $u.(pvp^* + qwq^*) \in \perp$ which is exactly the definition of $A^\perp \multimap B^\perp$.

We will denote u^\perp the operator:

$$u^\perp = pu_{22}p^* + pu_{12}q^* + qu_{12}p^* + qu_{11}q^*$$

where u_{ij} is given by externalization. Therefore the externalization of u^\perp is:

$$(u^\perp)_{ij} = u_{i\bar{j}} \text{ where } \bar{1} = 2, \bar{2} = 1.$$

From this we deduce that $u^\perp \in B^\perp \multimap A^\perp$ and that $(u^\perp)^\perp = u$.

16.3.5.2 Commutativity

Let σ be the operator:

$$\sigma = ppq^*q^* + pqp^*q^* + qpq^*p^* + qqp^*p^*.$$

One can check that σ is the internalization of the operator S on $H \oplus H \oplus H \oplus H$ defined by: $S(x_1 \oplus x_2 \oplus x_3 \oplus x_4) = x_4 \oplus x_3 \oplus x_2 \oplus x_1$. In particular the components of σ are:

$$\begin{aligned} \sigma_{11} &= \sigma_{22} = 0 \\ \sigma_{12} &= \sigma_{21} = pq^* + qp^*. \end{aligned}$$

Let A and B be types and u and v be operators in A and B . Then $pup^* + qvq^*$ is in $A \otimes B$ and as $\sigma_{11} \cdot (pup^* + qvq^*) = 0$ we may compute:

$$\begin{aligned} \text{App}(\sigma, pup^* + qvq^*) &= \sigma_{22} + \sigma_{21}(pup^* + qvq^*) \sum (\sigma_{11}(pup^* + qvq^*))^k \sigma_{12} \\ &= (pq^* + qp^*)(pup^* + qvq^*)(pq^* + qp^*) \\ &= pvp^* + quq^* \end{aligned}$$

But $pvp^* + quq^* \in B \otimes A$, thus we have shown that:

$$\sigma \in (A \otimes B) \multimap (B \otimes A).$$

16.3.5.3 Distributivity

We get distributivity by considering the operator:

$$\delta = ppp^*p^*q^* + pqpq^*p^*q^* + pqqq^*q^* + qppp^*p^* + qpqp^*q^*p^* + qq^*q^*p^*$$

that is similarly shown to be in type $A \otimes (B \otimes C) \multimap (A \otimes B) \otimes C$ for any types A, B and C .

16.3.5.4 Weak distributivity

Similarly we get weak distributivity thanks to the operators:

$$\begin{aligned} \delta_1 &= pppp^*q^* + ppqp^*q^*q^* + pqq^*q^*q^* + qpp^*p^*p^* + qpqp^*p^*p^* + qq^*q^*p^* \\ \delta_2 &= ppp^*p^*q^* + pqpq^*p^*q^* + pqqq^*q^* + qppp^*p^* + qpqp^*q^*p^* + qq^*q^*p^*. \end{aligned}$$

Given three types A, B and C then one can show that:

- δ_1 has type $((A \multimap B) \otimes C) \multimap A \multimap (B \otimes C)$ and
- δ_2 has type $(A \otimes (B \multimap C)) \multimap (A \multimap B) \multimap C$.

16.3.6 Execution formula, version 2: composition

Let A, B and C be types and u and v be operators respectively in types $A \multimap B$ and $B \multimap C$.

As usual we will denote u_{ij} and v_{ij} the operators obtained by externalization of u and v , eg, $u_{11} = p^*up, \dots$

As u is in $A \multimap B$ we have that $\text{App}(u, 0) = u_{22} \in B$; similarly as $v \in B \multimap C$, thus $v^\perp \in C^\perp \multimap B^\perp$, we have $\text{App}(v^\perp, 0) = v_{11} \in B^\perp$. Thus $u_{22}v_{11}$ is nilpotent.

We define the operator $\text{Comp}(u, v)$ by:

$$\begin{aligned} \text{Comp}(u, v) &= p(u_{11} + u_{12} \sum (v_{11}u_{22})^k v_{11}u_{21})p^* \\ &\quad + p(u_{12} \sum (v_{11}u_{22})^k v_{12})q^* \\ &\quad + q(v_{21} \sum (u_{22}v_{11})^k u_{21})p^* \\ &\quad + q(v_{22} + v_{21} \sum (u_{22}v_{11})^k u_{22}v_{12})q^* \end{aligned}$$

This is well defined since $u_{11}v_{22}$ is nilpotent. As an example let us compute the composition of u and ι in type $B \multimap B$; recall that $\iota_{ij} = \delta_{ij}$, so we get:

$$\text{Comp}(u, \iota) = pu_{11}p^* + pu_{12}q^* + qu_{21}p^* + qu_{22}q^* = u$$

Similar computation would show that $\text{Comp}(\iota, v) = v$ (we use $pp^* + qq^* = 1$ here).

Coming back to the general case we claim that $\text{Comp}(u, v)$ is in $A \multimap C$: let a be an operator in A . By computation we can check that:

$$\text{App}(\text{Comp}(u, v), a) = \text{App}(v, \text{App}(u, a)).$$

Now since u is in $A \multimap B$, $\text{App}(u, a)$ is in B and since v is in $B \multimap C$, $\text{App}(v, \text{App}(u, a))$ is in C .

If we now consider a type D and an operator w in $C \multimap D$ then we have:

$$\text{Comp}(\text{Comp}(u, v), w) = \text{Comp}(u, \text{Comp}(v, w))$$

Putting together the results of this section we finally have:

Theorem 37. Let $\text{GoI}(H)$ be defined by:

- objects are types, i.e. sets A of p -isometries satisfying: $A^{\perp\perp} = A$;
- morphisms from A to B are p -isometries in type $A \multimap B$;
- composition is given by the formula above.

Then $\text{GoI}(H)$ is a star-autonomous category.

16.4 GoI for MELL: exponentials

16.4.1 The tensor product of Hilbert spaces

Recall that we work in the Hilbert space $H = \ell^2(\mathbb{N})$ endowed with its canonical hilbertian basis denoted by $(e_k)_{k \in \mathbb{N}}$.

The space $H \otimes H$ is the collection of sequences $(x_{np})_{n,p \in \mathbb{N}}$ of complex numbers such that $\sum_{n,p} |x_{np}|^2$ converges. The scalar product is defined just as before:

$$\langle (x_{np}), (y_{np}) \rangle = \sum_{n,p} x_{np} \bar{y}_{np}.$$

If $x = (x_n)_{n \in \mathbb{N}}$ and $y = (y_p)_{p \in \mathbb{N}}$ are vectors in H then their tensor is the sequence:

$$x \otimes y = (x_n y_p)_{n,p \in \mathbb{N}}.$$

We define: $e_{np} = e_n \otimes e_p$ so that e_{np} is the sequence $(e_{npij})_{i,j \in \mathbb{N}}$ of complex numbers given by $e_{npij} = \delta_{ni} \delta_{pj}$. By bilinearity of tensor we have:

$$x \otimes y = \left(\sum_n x_n e_n \right) \otimes \left(\sum_p y_p e_p \right) = \sum_{n,p} x_n y_p e_n \otimes e_p = \sum_{n,p} x_n y_p e_{np}$$

Furthermore the system of vectors (e_{np}) is a hilbertian basis of $H \otimes H$: the sequence $x = (x_{np})_{n,p \in \mathbb{N}}$ may be written:

$$\begin{aligned} x &= \sum_{n,p \in \mathbb{N}} x_{np} e_{np} \\ &= \sum_{n,p \in \mathbb{N}} x_{np} e_n \otimes e_p. \end{aligned}$$

16.4.1.1 An algebra isomorphism

Being both separable Hilbert spaces, H and $H \otimes H$ are isomorphic. We will now define explicitly an iso based on partial permutations.

We fix, once for all, a bijection from couples of natural numbers to natural numbers that we will denote by $(n, p) \mapsto \langle n, p \rangle$. For example set $\langle n, p \rangle = 2^n(2p + 1) - 1$. Conversely, given $n \in \mathbb{N}$ we denote by $n_{(1)}$ and $n_{(2)}$ the unique integers such that $\langle n_{(1)}, n_{(2)} \rangle = n$.

Remark 8. Just as it was convenient but actually not necessary to choose p and q so that $pp^* + qq^* = 1$ it is actually not necessary to have a bijection, a one-to-one mapping from \mathbb{N}^2 into \mathbb{N} would be sufficient for our purpose.

This bijection can be extended into a Hilbert space isomorphism $\Phi : H \otimes H \rightarrow H$ by defining:

$$e_n \otimes e_p = e_{np} \mapsto e_{\langle n,p \rangle}.$$

Now given an operator u on H we define the operator $!u$ on H by:

$$!u(e_{\langle n,p \rangle}) = \Phi(e_n \otimes u(e_p)).$$

Remark 9. *The operator $!u$ is defined by:*

$$!u = \Phi \circ (1 \otimes u) \circ \Phi^{-1}$$

where $1 \otimes u$ denotes the operator on $H \otimes H$ defined by $(1 \otimes u)(x \otimes y) = x \otimes u(y)$ for any x, y in H . However this notation must not be confused with the *tensor of operators* that was defined in the previous section in order to interpret the tensor rule of linear logic; we therefore will not use it.

One can check that given two operators u and v we have:

- $!u!v = !(uv)$;
- $!(u^*) = (!u)^*$.

Due to the fact that Φ is an isomorphism *onto* we also have $!1 = 1$; this however will not be used.

We therefore have that $!$ is a morphism on $\mathcal{B}(H)$; it is easily seen to be an iso (not *onto* though). As this is the crucial ingredient for interpreting the structural rules of linear logic, we will call it the *copying iso*.

16.4.1.2 Interpretation of exponentials

If we suppose that $u = u_\varphi$ is a p -isometry generated by the partial permutation φ then we have:

$$!u(e_{\langle n,p \rangle}) = \Phi(e_n \otimes u(e_p)) = \Phi(e_n \otimes e_{\varphi(p)}) = e_{\langle n,\varphi(p) \rangle}.$$

Thus $!u_\varphi$ is itself a p -isometry generated by the partial permutation $!\varphi : n \mapsto \langle n_{(1)}, \varphi(n_{(2)}) \rangle$, which shows that the proof space is stable under the copying iso.

Given a type A we define the type $!A$ by:

$$!A = \{!u, u \in A\}^{\perp\perp}$$

Chapter 17

Game semantics

This article presents the game-theoretic **fully complete model** of *MLL*. Formulas are interpreted by games between two players, Player and Opponent, and proofs are interpreted by strategies for Player.

17.1 Preliminary definitions and notations

17.1.1 Sequences, Polarities

Definition 54 (Sequences). *If M is a set of moves, a sequence or a play on M is a finite sequence of elements of M . The set of sequences of M is denoted by M^* .*

We introduce some convenient notations on sequences.

- If $s \in M^*$, $|s|$ will denote the *length* of s ;
- If $1 \leq i \leq |s|$, s_i will denote the i -th move of s ;
- We denote by \sqsubseteq the prefix partial order on M^* ;
- If s_1 is an even-length prefix of s_2 , we denote it by $s_1 \sqsubseteq^P s_2$;
- The empty sequence will be denoted by ϵ .

All moves will be equipped with a **polarity**, which will be either Player (P) or Opponent (O).

- We define $\overline{(_)} : \{O, P\} \rightarrow \{O, P\}$ with $\overline{O} = P$ and $\overline{P} = O$.
- This operation extends in a pointwise way to functions onto $\{O, P\}$.

17.1.2 Sequences on Components

We will often need to speak of sequences over (the disjoint sum of) multiple sets of moves, along with a restriction operation.

- If M_1 and M_2 are two sets, $M_1 + M_2$ will denote their disjoint sum, implemented as $M_1 + M_2 = \{1\} \times M_1 \cup \{2\} \times M_2$;
- In this case, if we have two functions $\lambda_1 : M_1 \rightarrow R$ and $\lambda_2 : M_2 \rightarrow R$, we denote by $[\lambda_1, \lambda_2] : M_1 + M_2 \rightarrow R$ their *co-pairing*;
- If $s \in (M_A + M_B)^*$, the **restriction** of s to M_A (resp. M_B) is denoted by $s \upharpoonright M_A$ (resp. $s \upharpoonright M_B$). Later, if A and B are games, this will be abbreviated $s \upharpoonright A$ and $s \upharpoonright B$.

17.2 Games and Strategies

17.2.1 Game constructions

We first give the definition for a game, then all the constructions used to interpret the connectives and operations of *MLL*

Definition 55 (Games). A game A is a triple (M_A, λ_A, P_A) where:

- M_A is a finite set of moves;
- $\lambda_A : M_A \rightarrow \{O, P\}$ is a polarity function;
- P_A is a subset of M_A^* such that
 - Each $s \in P_A$ is alternating, i.e. if $\lambda_A(s_i) = Q$ then, if defined, $\lambda_A(s_{i+1}) = \bar{Q}$;
 - A is finite: there is no infinite strictly increasing sequence $s_1 \sqsubset s_2 \sqsubset \dots$ in P_A .

Definition 56 (Linear Negation). If A is a game, the game A^\perp is A where Player and Opponent are interchanged. Formally:

- $M_{A^\perp} = M_A$
- $\lambda_{A^\perp} = \bar{\lambda}_A$
- $P_{A^\perp} = P_A$

Definition 57 (Tensor). If A and B are games, we define $A \otimes B$ as:

- $M_{A \otimes B} = M_A + M_B$;
- $\lambda_{A \otimes B} = [\lambda_A, \lambda_B]$
- $P_{A \otimes B}$ is the set of all finite, alternating sequences in $M_{A \otimes B}^*$ such that $s \in P_{A \otimes B}$ if and only if:
 1. $s \upharpoonright A \in P_A$ and $s \upharpoonright B \in P_B$;
 2. If we have $i \leq |s|$ such that s_i and s_{i+1} are in different components, then $\lambda_{A \otimes B}(s_{i+1}) = O$. We will refer to this condition as the *switching convention* for tensor game.

The *par* connective can be defined either as $A \wp B = (A^\perp \otimes B^\perp)^\perp$, or similarly to the *tensor* except that the switching convention is in favor of Player. We will refer to this as the *switching convention for par game*. Likewise, we define $A \multimap B = A^\perp \wp B$.

17.2.2 Strategies

Definition 58 (Strategies). A strategy for Player in a game A is defined as a subset $\sigma \subseteq P_A$ satisfying the following conditions:

- σ is non-empty: $\epsilon \in \sigma$
- Opponent starts: If $s \in \sigma$, $\lambda_A(s_1) = O$;
- σ is closed by even prefix, i.e. if $s \in \sigma$ and $s' \sqsubseteq^P s$, then $s' \in \sigma$;
- Determinacy: If we have $soa \in \sigma$ and $sob \in \sigma$, then $a = b$.

We write $\sigma : A$.

Composition is defined by parallel interaction plus hiding. We take all valid sequences on A, B and C which behave accordingly to σ (resp. τ) on A, B (resp. B, C). Then, we hide all the communication in B .

Definition 59 (Parallel Interaction). *If A, B and C are games, we define the set of interactions $I(A, B, C)$ as the set of sequences s over A, B and C such that their respective restrictions on $A \multimap B$ and $B \multimap C$ are in $P_{A \multimap B}$ and $P_{B \multimap C}$, and such that no successive moves of s are respectively in A and C , or C and A . If $\sigma : A \multimap B$ and $\tau : B \multimap C$, we define the set of parallel interactions of σ and τ , denoted by $\sigma || \tau$, as $\{s \in I(A, B, C) \mid s \upharpoonright A, B \in \sigma \wedge s \upharpoonright B, C \in \tau\}$.*

Definition 60 (Composition). *If $\sigma : A \multimap B$ and $\tau : B \multimap C$, we define $\sigma; \tau = \{s \upharpoonright A, C \mid s \in \sigma || \tau\}$.*

We also define the identities, which are simple copycat strategies : they immediately copy on the left (resp. right) component the last Opponent's move on the right (resp. left) component. In the following definition, let L (resp. R) denote the left (resp. right) occurrence of A in $A \multimap A$.

With these definitions, we get the following theorem:

Theorem 38 (Category of Games and Strategies). *Composition of strategies is associative and identities are neutral for it. More precisely, there is a ***-autonomous category** with games as objects and strategies on $A \multimap B$ as morphisms from A to B .*

Appendix A

Notations

A.1 Logical systems

For a given logical system such as *MLL* (for multiplicative linear logic), we consider the following variations:

Notation	Meaning	Connectives
<i>MLL</i>	propositional without units	X, \otimes, \wp
<i>MLL_u</i>	propositional with units only	$1, \perp, \otimes, \wp$
<i>MLL₀</i>	propositional with units and variables	$X, 1, \perp, \otimes, \wp$
<i>MLL₁</i>	first-order without units	$X\vec{t}, \otimes, \wp, \forall xA, \exists xA$
<i>MLL₀₁</i>	first-order with units	$X\vec{t}, 1, \perp, \otimes, \wp, \forall xA, \exists xA$
<i>MLL₂</i>	second-order propositional without units	$X, \otimes, \wp, \forall XA, \exists XA$
<i>MLL₀₂</i>	second-order propositional with units	$X, 1, \perp, \otimes, \wp, \forall XA, \exists XA$
<i>MLL₁₂</i>	first-order and second-order without units	$X\vec{t}, \otimes, \wp, \forall xA, \exists xA, \forall XA, \exists XA$
<i>MLL₀₁₂</i>	first-order and second-order with units	$X\vec{t}, 1, \perp, \otimes, \wp, \forall xA, \exists xA, \forall XA, \exists XA$

A.2 Formulas and proof trees

A.2.1 Formulas

- First order quantification: $\forall xA$ with substitution $A[t/x]$
- Second order quantification: $\forall XA$ with substitution $A[B/X]$
- Quantification of arbitrary order (mainly first or second): $\forall \xi A$ with substitution $A[\tau/\xi]$

A.2.2 Rule names

Name of the connective, followed by some additional information if required, followed by "L" for a left rule or "R" for a right rule. This is for a two-sided system, "R" is implicit for one-sided systems. For example: $\wedge_1\text{addL}$.

A.3 Semantics

A.3.1 Coherent spaces

- Web of the space X : $|X|$
- Coherence relation of the space X : $\text{large} \supset_X$ and $\text{strict} \frown_X$

A.3.2 Finiteness spaces

- Web of the finiteness space \mathcal{A} : $|\mathcal{A}|$
- Finiteness structure of the space \mathcal{A} : $\mathfrak{F}(\mathcal{A})$ (we use $\mathfrak{mathfrak{F}}$, which is consistent with the fact that $\mathfrak{P}_f(|\mathcal{A}|) \subseteq \mathfrak{F}(\mathcal{A}) \subseteq \mathfrak{P}(|\mathcal{A}|)$).

A.4 Nets

- The free ports of a net R : $\text{fp}(R)$.
- The result of the connection of two nets R and R' , given the partial bijection $f : \text{fp}(R) \leftrightarrow \text{fp}(R')$: $R \bowtie_f R'$.
- The number of loops in the resulting net: $\langle\langle R|R' \rangle\rangle_f$ (includes the loops already present in R and R').

A.5 Miscellaneous

- **Isomorphism**: $A \cong B$
- **injection**: $A \hookrightarrow B$
- **partial injection**: $A \hookrightarrow B$

Bibliography

- [Ben94] Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models (extended abstract). In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic*, volume 933 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 1994.
- [CMM10] Pierre-Louis Curien Curien and Guillaume Munch-Maccagnoni. The duality of computation under focus. In Cristian S. Calude and Vladimiro Sassone, editors, *IFIP Theoretical Computer Science*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 165–181. Springer, 2010.
- [Fio07] Marcello P. Fiore. Differential structure in models of multiplicative biadditive intuitionistic linear logic. In Simona Ronchi Della Rocca, editor, *Typed Lambda Calculi and Applications (TLCA)*, volume 4583 of *Lecture Notes in Computer Science*, pages 163–177. Springer, 2007.
- [Gir86] Jean-Yves Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45:159–192, 1986.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.
- [Gir98] Jean-Yves Girard. Light linear logic. *Information and Computation*, 43:175–204, 1998.
- [Laf04] Yves Lafont. Soft linear logic and polynomial time. *Theoretical Computer Science*, 318(1–2):163–180, 2004.
- [Mac98] Saunders MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer, second edition, 1998.
- [Mel09] Paul-André Melliès. *Interactive models of computation and program behaviour*, volume 27 of *Panoramas et Synthèses*, chapter Categorical Semantics of Linear Logic, pages 1–196. Société Mathématique de France, 2009.
- [Sch94] Harold Schellinx. *The Noble Art of Linear Decorating*. Dissertation series of the Dutch Institute for Logic, Language and Computation, University of Amsterdam, 1994. ILLC-Dissertation Series, 1994-1.