Université Paris 13

Mémoire pour l'obtention de l'habilitation à diriger les recherches
Spécialité : Informatique

Some Advances in Linear Logic

par Michele Pagani

Soutenance le 5 Décembre 2013
devant le jury composé de

Laboratoire d'Informatique de Paris-Nord (LIPN), UMR CNRS 7030,
Institut Galilée - Université Paris 13 SPC

Contents

Remerciements

- Merci infiniment à tous ceux avec qui j'ai eu la chance de publier : Pierre Boudes, Daniel de Carvalho, Alejandro Diaz-Caro, Thomas Ehrhard, Fanny He, Jim Laird, Giulio Manzonetto, Guy McCusker, Damiano Mazza, Simona Ronchi della Rocca, Alexis Saurin, Peter Selinger, Christine Tasson, Lorenzo Tortora de Falco, Paolo Tranquilli, Benoit Valiron. J'ai appris énormément en travaillant avec eux, chaque article étant une étape fondamentale de ma formation. Un merci tout particulier à Damiano, Christine et Giulio, mes compagnons historiques, avec qui j'ai grandi et partagé les recherches, les inquiétudes et les satisfactions.

- Merci aux rapporteurs (Jean Goubault-Larrecq, Simone Martini et Laurent Regnier) et aux autres membres du jury (Thomas Ehrhard, Christophe Fouqueré, Stefano Guerrini, Luke Ong) qui ont accepté de me consacrer un peu de leur temps précieux. Merci à Christophe pour m'avoir parrainé et pour son conseil avisé au long de ces années.

- Merci au LIPN et au CNRS, qui ont assuré largement les moyens et le support nécessaires à conduire mes recherches. Merci à tous mes collègues, pour l'ambiance amicale et très stimulante.

- Sopratutto, grazie ad Arianna per avermi sempre sostenuto nell'intenso lavoro di questi anni, trasformando le mie insicurezze in stimoli a progredire.

Introduction

Ce document a été conçu pour décrire mon parcours scientifique depuis la fin de ma thèse de doctorat (avril 2006) jusqu'à aujourd'hui. En conséquence, les sujets suivent à peu près l'ordre chronologique de ma recherche afin de donner une idée de son évolution.

Après ma thèse, je me suis concentré sur l'étude des propriétés de réécriture de l'élimination des coupures de la logique linéaire et de son extension différentielle. J'ai été attiré d'une part par ses relations avec des calculs avec ressources bornées et de concurrence, de l'autre part par son élégance et sa robustesse en tant que système de réécriture. C'est seulement à travers mes travaux récents que j'ai pu saisir toute la richesse des sémantiques quantitatives, qui sont à l'origine de la logique linéaire et de son extension différentielle, et qui leur donne un fondement mathématique et une allure particulière parmi d'autres systèmes logiques.

Les sémantiques quantitatives s'inscrivent dans un profond renouvellement de la correspondance de Curry-Howard, entre la sémantique formelle des langages de programmation d'un côté et la théorie de la démonstration de l'autre. L'algèbre linéaire et l'analyse fonctionnelle s'imposent comme troisième pôle de cette correspondance, en mettant au centre la notion de *ressource* du calcul. Il s'agit d'une approche très riche, qui peut potentiellement être appliquée dans différents domaines de l'informatique théorique.

Nous en sommes maintenant à un stade de la recherche où cette potentialité *peut* et *doit* être transformée en acte. Il nous faut ainsi donner des exemples concrets d'applications de cette approche : une étude de nouvelles primitives de programmation (comme la composition parallèle des processus, les bits quantiques, etc.), ou bien une étude des algorithmes non seulement par rapport à ce qu'ils calculent, mais aussi relativement à la manière dont ils calculent (avec combien de ressources, en combien de temps, et avec quelle probabilité).

Revenons sur une question posée par un rapporteur à propos d'un article sur la logique linéaire différentielle :

Why is the differential linear logic, or the quantitative semantics of any interest to computer science ?

Apporter des réponses concrètes à cette question est la motivation la plus forte de ma recherche actuelle.

L'étude des langages avec des primitives non-déterministes, comme PCF probabiliste ou le λ-calcul quantique de Selinger et Valiron ont donné des résultats encourageants. Dans ces cas, cette ligne de recherche a été fructueuse : notamment, [EPT14] (Section 3.3) donne la première démonstration d'abstraction pleine d'un modèle dénotationel probabiliste, et [PSV14] (Section 3.4) présente une sémantique concrète du λ-calcul quantique basée sur les hermitiens positifs.

Bien sûr, avoir une sémantique dénotationelle, même pleinement abstraite, ne doit pas être le but mais bien un début. Aussi, je ne pense pas que l'approche de la logique linéaire différentielle bénéficie exclusivement aux langages non-déterministes, il est permis de penser que des résultats peuvent être obtenues dans des cadres déterministes, comme des notions indépendantes de la machine du coût de calcul (temps, espaces, utilisations de certains constructeurs), ou encore des analyses en moyenne du comportement des programmes sur certaines structures de données.

Structure du mémoire. Les résultats sont regroupés en trois chapitres. Chaque chapitre contient des sections préliminaires (Sections 1.1, 1.2, 2.1, 3.1) rappelant le contexte scientifique de ma recherche. Chaque section présentant mes travaux s'ouvre avec un bref résumé (sous fond gris) donnant un premier aperçu des résultats détaillés dans la section. Afin de donner une vision d'ensemble au lecteur je liste ici, de façon informelle, les énoncés clés du mémoire.

Théorème 4 : l'élimination des coupures sur les réseaux à tranches additives de la logique linéaire satisfait la propriété de conservation (c.à-d. la réduction *non-erasing* (Définition 3) préserve l'existence des réductions infinies). Ce théorème était le lemme manquant à la démonstration de normalisation forte de l'élimination des coupures de la logique linéaire.

Théorème 5 : l'interprétation relationnelle de deux réseaux en forme normale permet de calculer le nombre de pas nécessaires à éliminer une coupure entre les deux réseaux.

Théorèmes 9 et 10 : l'*acyclicité visible* (Définition 8) caractérise l'ensemble des réseaux différentiels interprétés par des relations finitaires dans les espaces de finitude.

Conjecture 1 : l'ensemble des réseaux différentiels visiblement acycliques est le plus grand ensemble contenant les réseaux de preuve différentiels et ayant une élimination des coupures fortement normalisante.

Théorème 11 (resp. 12) : la réduction du calcul avec ressources satisfait la propriété de confluence (resp. de standardisation).

Théorème 14 (resp. 15) : la résolubilité angélique (resp. démoniaque) du calcul avec ressources (resp. calcul non-déterministe) est caractérisée aussi bien opérationnellement qu'à travers un système de types avec intersection.

Théorème 16 : la $\eta\tau$-équivalence (Figure 2.5) génère la congruence maximale non-triviale sur les formes normales du calcul avec ressources. Notamment, si deux formes normales sont $\eta\tau$-différentes alors elles sont séparables par des termes avec ressources.

Théorème 17 : l'ensemble des développements de Taylor des λ-termes est caractérisé à l'intérieur des séries des termes multi-linéaires.

Théorème 20 : la sémantique des ensembles et relations avec poids dans un semi-anneau continu donne un modèle de la logique linéaire.

Table 3.1 : quelques exemples d'observation quantitative sur les programmes de PCF avec choix non-déterministe décrits par les sémantiques relationnelles avec poids.

Théorème 25 : les espaces cohérents probabilistes donnent un modèle pleinement abstrait de PCF probabiliste.

Théorème 27 : l'extension aux bi-produits infinis de la catégorie des fonctions complètement positives avec symétries donne un modèle de la logique linéaire.

Théorème 28 : l'extension aux bi-produits infinis de la catégorie des fonctions complètement positives avec symétries donne un modèle adéquat du λ-calcul quantique avec récursion et types infinis.

Enfin, je donne aussi une liste des problèmes ouverts décrits dans le mémoire. Ces problèmes ont été pensés comme exemples de questions possibles que je pourrais proposer comme sujet de stage ou de thèse. En effet, chaque question est classée par un grade de difficulté : $\star$ est un problème simple, convenable à un sujet de stage de Master ; $\star\star$ est un problème dont j'ai le sentiment qu'il est résoluble en quelques mois de travail, mais il peut bien cacher des difficultés plus profondes : il peut être un sujet de stage de Master 2 projeté vers un travail de thèse. Enfin les problèmes classés $\star\star\star$ sont des problèmes plus complexes probablement adaptés pour un travail de thèse ou de recherche.

Problème ouvert 1 ($\star\star\star$) : est-il possible de caractériser sémantiquement le critère de switching acyclicity en présence des boîtes exponentielles ? D'autre part, quel est le sens logique de l'acyclicité visible induite par les espaces cohérents et de finitude, s'il y en a un ?

Problème ouvert 2 ($\star\star$) **:** est-ce que l'acyclicté induite par les espaces d'hypercohérence correspond au critère de correction des réseaux linéaires à tranches additives ?

Problème ouvert 3 ($\star$) **:** comment définir un calcul avec ressources correspondant au fragment appel-par-valeur des réseaux différentiels ? quel est la notion de résolubilité qui va avec ?

Problème ouvert 4 ($\star\star\star$) **:** donner un modèle pleinement abstrait de la résolubilité angélique dans le calcul avec ressources, ainsi que dans le λ-calcul non déterministe.

Problème ouvert 5 ($\star\star$) **:** est-il possible de généraliser le théorème de séparation aux termes avec ressources sans forme normale ? Est-il possible de caractériser l'équivalence induite par la résolubilité angélique avec une notion d'équivalence entre structures d'arbres ?

Problème ouvert 6 ($\star\star\star$) **:** pouvons-nous caractériser les séries des termes multi-linéaires qui sont les développements de Taylor des termes dans des langages non-déterministes, voire probabilistes ?

Problème ouvert 7 ($\star\star$) **:** est-il possible de donner des modèles relationnels avec poids d'un langage polymorphe, comme System F ? Quelles propriétés quantitatives sur les programmes peuvent être décrites par de tels modèles ?

Problème ouvert 8 ($\star\star$) **:** est-il possible de trouver une description directe du collapse extensionnel de la catégorie cartésienne fermée induite par les modèles relationnelles avec poids ?

Problème ouvert 9 ($\star\star$) **:** Quels sont les liens entre les espaces cohérents probabilistes et les modèles de PCF probabilistes basés sur une notion de monade probabiliste sur des domaines de Scott ?

Problème ouvert 10 ($\star\star$) **:** est-ce que les espaces cohérents probabilistes donnent un modèle pleinement abstrait du λ-calcul pur probabiliste ?

Problème ouvert 11 ($\star\star$) **:** Y a-t-il une notion de *clôture par double polarité* permettant de raffiner notre modèle du λ-calcul quantique et d'éviter la constante formelle ∞ ?

Problème ouvert 12 ($\star\star$) **:** pouvons nous définir des relations logiques pour le λ-calcul quantique et ainsi obtenir une démonstration plus modulaire du théorème d'adéquation ?

Chapter 1

On cut-elimination

- M. Pagani, P. Tranquilli. *The Conservation Theorem for Differential Nets*. Accepted to: Mathematical Structures in Computer Science.
- M. Pagani. *Visible Acyclic Differential Nets, Part I : Semantics*. Annals of Pure and Applied Logic, vol. 163, num. 3, 2012.
- D. de Carvalho, M. Pagani, L. Tortora de Falco. *A Semantic Measure of the Execution Time in Linear Logic*. Girard's Festschrift, special issue Theoretical Computer Science, vol. 412, num. 20, 2011.
- M. Pagani, L. Tortora de Falco. *Strong Normalization Property for Second Order Linear Logic*. Theoretical Computer Science, vol. 411, Springer 2010.

$A, B, C ::= X \mid X^\perp$	variables
$\mid \mathbf{1} \mid \perp \mid A \parr B \mid A \multimap B \mid A \otimes B$	multiplicatives
$\mid \top \mid \mathbf{0} \mid A \oplus B \mid A \,\&\, B$	additives
$\mid\, !A \mid ?A$	exponentials
$\mid \forall X.A \mid \exists X.A$	second order

Figure 1.1: grammar of formulas of second order linear logic, LL^2. MLL denotes the fragment defined by variables and multiplicatives; MALL refers to MLL plus additives; MELL refers to MLL plus exponentials. A $(\ )^2$ apex refers to a system with second order quantifiers also (e.g. MLL^2, MALL^2, MELL^2).

1.1 Preliminaries on Linear Logic

The formulas of second order linear logic (LL^2) are given in Figure 1.1 and its sequent calculus in Figure 1.2. We use a one-side sequent calculus, halving the number of sequent rules. Also, we suppose the linear arrow as a notational convention for a $\parr$:

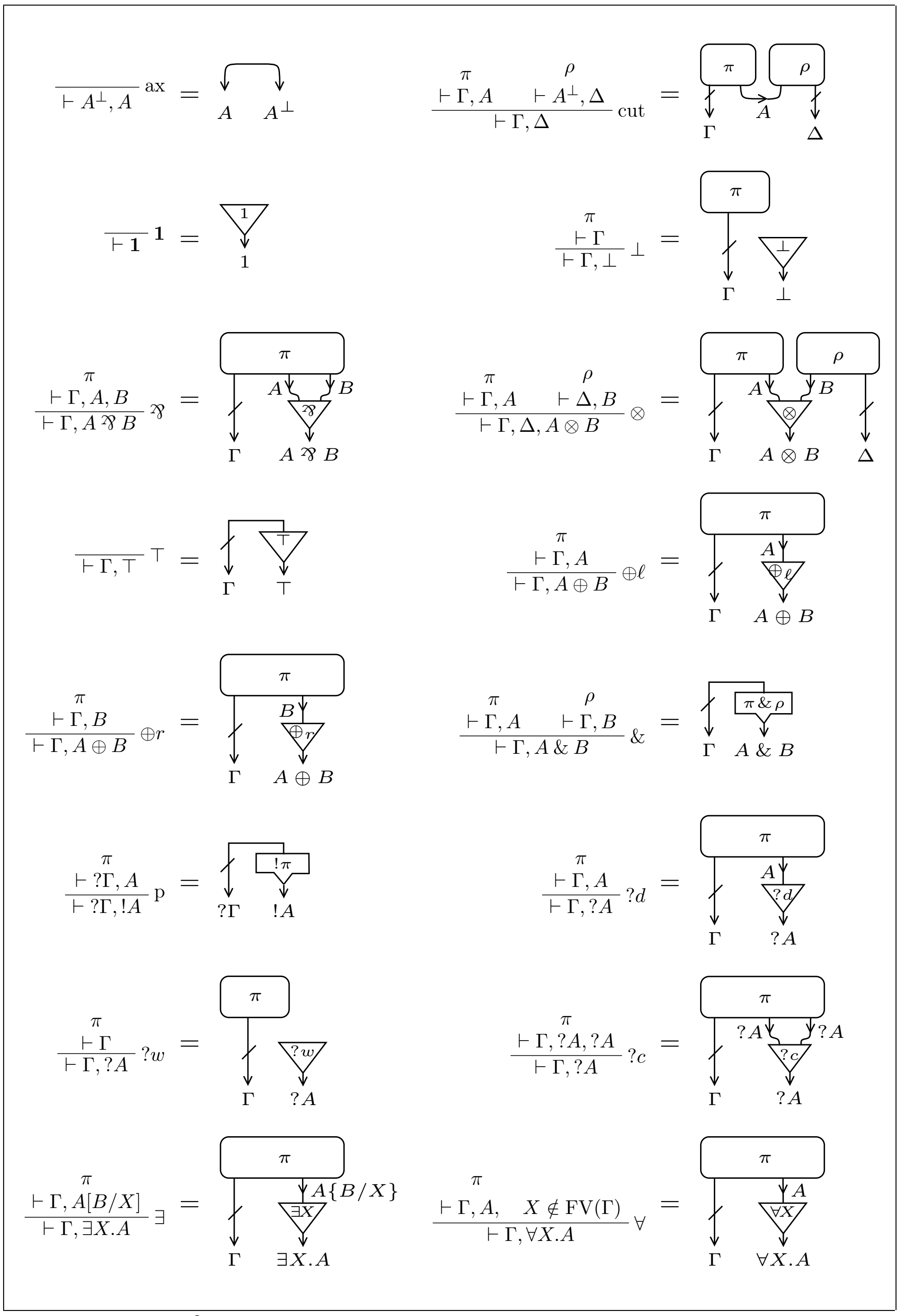
$$A \multimap B \triangleq A^\perp \parr B.$$

Linear logic has been introduced by Girard in [Gir87a] and can be seen as a refinement of classical logic, where the usual conjunction $\wedge$ and disjunction $\vee$ split into a multiplicative version (resp. tensor $\otimes$ and par $\parr$) and an additive version (resp. with $\&$ and plus $\oplus$). Two new modalities of course ! and why not ? are introduced, handling the structural rules – only ?-formulas admit weakening and contraction. The rule introducing a !-formula (resp. ?-formula) is called promotion (resp. dereliction).

Negation $A^\perp$ is defined as a primitive connective on variables and then extended to the whole grammar by notational convention, following De Morgan and involutive laws:

$$\begin{array}{ll} \mathbf{1}^\perp \triangleq \perp, & (A \otimes B)^\perp \triangleq A^\perp \parr B^\perp \triangleq A \multimap B^\perp, \\ \mathbf{0}^\perp \triangleq \top, & (A \oplus B)^\perp \triangleq A^\perp \,\&\, B^\perp, \\ (\forall X.A)^\perp \triangleq \exists X.A^\perp, & A^{\perp\perp} \triangleq A. \end{array}$$

There are various motivations behind the introduction of linear logic. The first one sinks its roots in denotational semantics, and will be mentioned in Section 3, when presenting my most recent research. A syntactical motivation relies on the fact that linear logic expresses the computational



$\dfrac{}{\vdash A^\perp, A}\,\text{ax}$ =
A $A^\perp$
$\dfrac{\overset{\pi}{\vdash \Gamma, A} \qquad \overset{\rho}{\vdash A^\perp, \Delta}}{\vdash \Gamma, \Delta}\,\text{cut}$ =
π ρ Γ A Δ
$\dfrac{}{\vdash \mathbf{1}}\,\mathbf{1}$ =
1 1
$\dfrac{\overset{\pi}{\vdash \Gamma}}{\vdash \Gamma, \bot}\,\bot$ =
π $\bot$ Γ $\bot$
$\dfrac{\overset{\pi}{\vdash \Gamma, A, B}}{\vdash \Gamma, A \parr B}\,\parr$ =
π A B $\parr$ Γ $A \parr B$
$\dfrac{\overset{\pi}{\vdash \Gamma, A} \qquad \overset{\rho}{\vdash \Delta, B}}{\vdash \Gamma, \Delta, A \otimes B}\,\otimes$ =
π ρ A B $\otimes$ Γ $A \otimes B$ Δ
$\dfrac{}{\vdash \Gamma, \top}\,\top$ =
$\top$ Γ $\top$
$\dfrac{\overset{\pi}{\vdash \Gamma, A}}{\vdash \Gamma, A \oplus B}\,\oplus\ell$ =
π A $\oplus\ell$ Γ $A \oplus B$
$\dfrac{\overset{\pi}{\vdash \Gamma, B}}{\vdash \Gamma, A \oplus B}\,\oplus r$ =
π B $\oplus r$ Γ $A \oplus B$
$\dfrac{\overset{\pi}{\vdash \Gamma, A} \qquad \overset{\rho}{\vdash \Gamma, B}}{\vdash \Gamma, A \,\&\, B}\,\&$ =
$\pi \,\&\, \rho$ Γ $A \,\&\, B$
$\dfrac{\overset{\pi}{\vdash ?\Gamma, A}}{\vdash ?\Gamma, !A}\,\text{p}$ =
$!\pi$ $?\Gamma$ $!A$
$\dfrac{\overset{\pi}{\vdash \Gamma, A}}{\vdash \Gamma, ?A}\,?d$ =
π A $?d$ Γ $?A$
$\dfrac{\overset{\pi}{\vdash \Gamma}}{\vdash \Gamma, ?A}\,?w$ =
π $?w$ Γ $?A$
$\dfrac{\overset{\pi}{\vdash \Gamma, ?A, ?A}}{\vdash \Gamma, ?A}\,?c$ =
π $?A$ $?A$ $?c$ Γ $?A$
$\dfrac{\overset{\pi}{\vdash \Gamma, A[B/X]}}{\vdash \Gamma, \exists X.A}\,\exists$ =
π $A\{B/X\}$ $\exists X$ Γ $\exists X.A$
$\dfrac{\overset{\pi}{\vdash \Gamma, A, \quad X \notin \mathrm{FV}(\Gamma)}}{\vdash \Gamma, \forall X.A}\,\forall$ =
π A $\forall X$ Γ $\forall X.A$

Figure 1.2: LL^2 sequent calculus rules and their translation into proof nets. A struck wire denotes a bunch of wires. In the proof net associated with the $\forall$ rule, we suppose that no other $\forall$ cell in π uses X as eigenvariable.

content of cut-elimination in more evidence than classical and intuitionistic logic. For example, there are two different versions of the sequent rule associated with the classical disjunction:

$$\begin{array}{ccc} \text{additive version:} & & \text{multiplicative version:} \\ \dfrac{\vdash \Gamma, A}{\vdash \Gamma, A \vee B} \quad \dfrac{\vdash \Gamma, B}{\vdash \Gamma, A \vee B} & & \dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B} \end{array} \tag{1.1}$$

They are equivalent from the point of view of provability, however they have a fairly different behavior at the level of cut-elimination. Reducing a cut between an additive $\vee$-rule and a $\wedge$-rule consists in choosing one premise of the $\wedge$-rule, while, in the multiplicative case, consists in establishing a communication between the two premises of the $\wedge$-rule. Linear logic gives a logical status to such different computational behaviors: branching (ruled by $\&$ and $\oplus$) and inter-communication (ruled by $\otimes$ and $⅋$). Similar reasoning can be done with respect to other groups of connectives: exponentials express duplication and erasing, linear negation continuations, etc., see e.g. [Abr93].

In some sense, linear logic grew out of cut-elimination. This conviction mainly motivates my study on the properties of linear logic as a proof rewriting system, done right after my PhD, during a post-doc leaded by Tortora de Falco (professor at Roma 3).

Proof Nets

In sequent calculus, the logical steps of cut-elimination are buried under a plethora of commutative steps, needed to move a cut right below the two rules introducing the cut formulas. This hides the computational content of the logical rules, in particular it lacks an account of the asynchrony among the various steps of the reduction.

Proof nets have been introduced in [Gir87a] in order to overcome such deficiencies. The idea is to give a context-free representation of sequent rules by using graphs more complex than derivation trees. The number of commutative cuts then decreases significantly, the logical rules disclosing their computational meaning in all its beauty.

Proof nets are defined inductively by translating sequent proofs, following Figure 1.2. A proof net can be seen as a kind of circuit made of cells and wires[1]. A cell is labelled by a sequent rule or by boxes (i.e. other proof nets), the wire incident to the tip of a cell corresponds to the active formula in the conclusion of the sequent rule and is called principal wire of the cell, while the wires (if any) incident to the opposite side of a tip are the auxiliary wires and correspond to the premises of the associated rule or, in case of boxes,

[1] The precise definition of proof nets for the whole system of linear logic hides endlessly technicalities. Since the goal here is to give an intuitive account of my results, I stay at an informal level, referring to the literature for more detailed discussions.

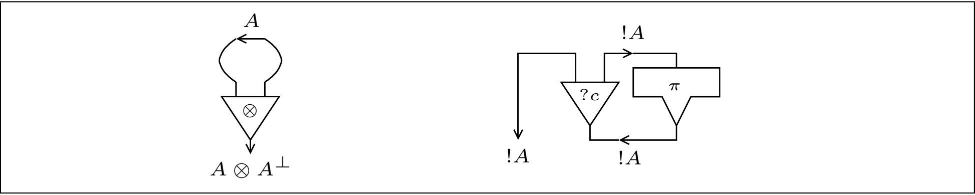


Figure 1.3: examples of incorrect net.

to the formulas in the context of the sequent calculus conclusion. Boxes are of two kinds: exponential boxes (associated with the promotion rule) and additive boxes (associated with the &). They allow for expressing the context conditions of the corresponding sequent calculus rules.

Oriented wires are labelled by formulas, in such a way that if one orientation is associated with a formula A, the opposite orientation is associated with the dual formula $A^{\perp}$. Indeed, such convention allows for representing axioms and cuts as simple wires: an axiom is a wire which is not principal of any cell, nor auxiliary of a box, while a cut is a wire which is either principal of two cells, or principal of one cell and auxiliary of a box, or auxiliary of two additive boxes.

Some wires may have extremities pending out of the net: they correspond to the conclusions of the proof. By extension of the language, we can speak about a wire or a cell at exponential depth n of a proof net π, referring to an wire or a cell occurring in a proof net "inside" a nesting of n exponential boxes of π.

Correctness Criteria

The syntax of proof nets introduces new objects in proof theory: *wrong* proofs. In fact, *proof nets* belong to a wider class of circuits, that of *nets* (or *proof structures* in Girard's terminology). Not every net is a proof net, i.e. can be defined inductively following the rules of Figure 1.2 for LL^2. Some nets represent proofs with errors, intuitively they are argumentations using part of the thesis as hypothesis. For example, the circuits in Figure 1.3 have no correspondent sequent proof. We call net any circuit made by LL^2 cells.

A challenging problem is to give graph-theoretical characterizations of the set of proof nets, i.e. of the nets that are associated with sequent proofs. These characterizations are called correctness criteria. Two notable examples of correctness criteria are Girard's *longtrip condition* [Gir87a] and Danos and Regnier's *switching acyclicity and connectivity* [DR89], characterizing the set of proof nets in the fragment MLL without units ($\mathbf{1}$, $\perp$). Just to have an idea of how challenging the general problem is, remark that, as far as I know, no acceptable correctness criterion for MLL with units has been found so far.

In the sequel, we consider Danos and Regnier's switching acyclicicty criterion, characterizing the nets associated with the sequent rules of MELL plus

$$\frac{}{\vdash}\text{empty} \qquad \frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\text{mix}$$

Figure 1.4: empty and mix sequent rules.

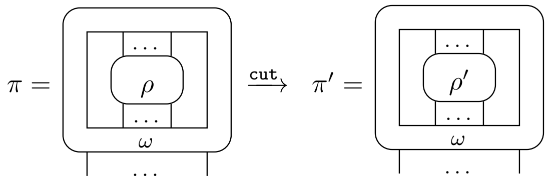
the rules empty and mix defined in Figure 1.4. The empty rule is associated with the empty net, while the mix is translated into the juxtaposition of the proof nets associated with the two premises.

A path[2] in a net π is called switching whenever it does not contain both premises of a cell of type $⅋$. A net is switching acyclic whenever it contains no switching cycle and all its boxes are switching acyclic. We have:

Proposition 1 (From [DR89]). *A* MELL *net π is switching acyclic iff it is the translation of a sequent proof of* MELL *plus empty and mix.*

Cut-elimination

Proof nets implement the cut-elimination as a graph rewriting relation defined by a set of elementary reduction steps. One elementary reduction step (ers for short) is a pair $\pi \to \pi'$ of two nets, such that π' is obtained by replacing a specific sub-circuit ρ with another circuit ρ' having the same conclusions as ρ, i.e. the same number and types of pending wires[3]:



The circuits ρ and ρ' active in the ers are called, respectively, redex and contractum of the ers, while the sub-circuit ω remained inactive is the context of the ers. Table 1.5 sketches the pairs of redex/contractum defining the cut-elimination in LL^2.[4] Because of the boxes, we need also to be closed under the following inductive rules: if $\rho \to \rho'$, then $!\rho \to !\rho'$, and $\rho \,\&\, \rho'' \to \rho' \,\&\, \rho''$, and $\rho'' \,\&\, \rho \to \rho'' \,\&\, \rho'$.

Notice that propositional ers do not depend on the formulas labeling wires, but just on the type of the cells. This means that one can study the

[2]We omit here the formal definition of a path in a net: think just of an alternating sequence of distinct cells and wires $(c_1, w_1, c_2, w_2, \dots, c_n)$ such that w_i is a wire between c_i and c_{i+1}. We refer to [Pag12] for more details.

[3]Actually, this operation is quite subtle, because of the possible merging of several wires into one single wire. We refer to [dF10] for a precise definition in the general setting of Lafont's interaction nets.

[4]In the case of the $\forall/\exists$ ers, one must suppose that the eigenvariable of the $\forall$-cell is not free in the context ω.

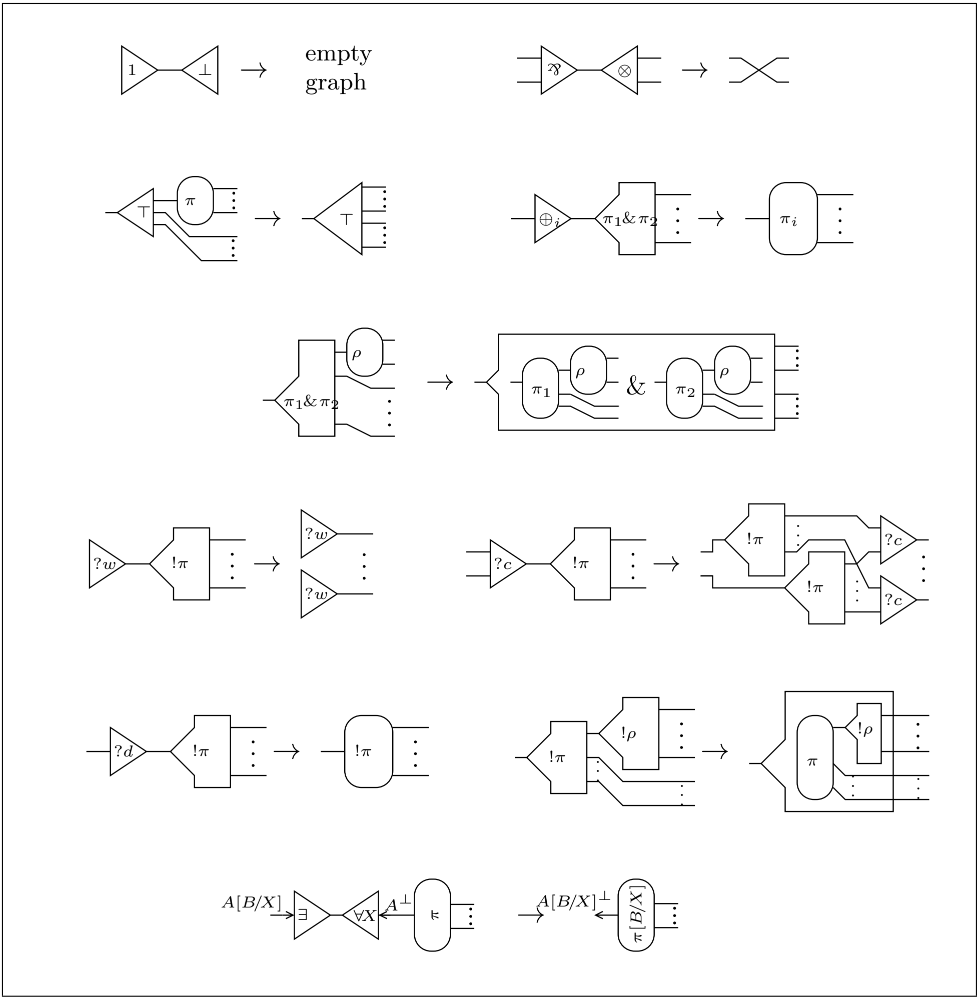


Figure 1.5: elementary reduction steps (ers) of LL^2 cut-elimination.

rewriting system on completely untyped nets, or, less radically, on nets labeled by some form of recursive types. The most notable example are Danos and Regnier's pure nets [Dan90, Reg92], defined on a set of four formulas $\{o, i, ?i \parr o, !o \otimes i\}$, enjoying the equations

$$o = ?i \parr o \; (= !o \multimap o), \qquad o^{\perp} = i. \tag{1.2}$$

This system allows us to represent untyped λ-terms as proof nets, since the first equation gives $o = o \to o$ via Girard's translation of the functional type $o \to o$ into $!o \multimap o$. Pure nets have been at the origin of a fruitful exchange of tools and issues between λ-calculus and linear logic.

The rewriting rules of cut-elimination are also independent from the correctness of the net. However, in presence of switching cycles, even simply typed nets can yield infinite cut-elimination reductions. For example, the net at the right-hand side of Figure 1.3 yields an infinite cut-elimination sequence. This can be used to model fix-point combinators, allowing for encoding full recursion within propositional linear logic, see e.g. [Mon03].

1.1.1 The Strong Normalization Theorem of LL^2 (eventually) (2007-2009)

One among the main theorems in [Gir87a] is the following:

Theorem 2. *Cut-elimination is strongly normalizing on* LL^2 *proof nets.*

The proof adapts to LL^2 the notion of reducibility candidate, developed in [Gir72] for achieving the strong normalization (SN) of β-reduction in System F. Unfortunately, the reducibility technique is not enough for proof nets: another crucial ingredient is the following conservation property.

Conservation[5]: if $\pi \to \pi'$ is not an *erasing ers* and π' is SN, then π is SN.

The proof of this property is missing in [Gir87a], and even the given definition of *erasing* ers ([Gir87a, Definition 4.24, p. 72]) is incomplete. The main goal of [PTdF10], written in collaboration with Tortora de Falco, was to fill this (rather big) gap in the linear logic literature, giving a correct definition of erasing ers (Definition 3) and achieving a conservation theorem (Theorem 4) in the setting of untyped sliced proof nets.

In λ-calculus, a β-redex $(\lambda x.M)N$ is not erasing whenever the variable x occurs free in M. In this case, the proof that any infinite reduction sequence from $(\lambda x.M)N$ induces an infinite reduction sequence starting from $M\{N/x\}$, is an easy consequence of the fact that the reduction of the β-redex $(\lambda x.M)N$ can always be anticipated with respect to the reduction of a redex in M or in N.

This reasoning however cannot be adapted to the syntax of proof nets for two main issues.

(i) The ers of Figure 1.5 do not easily split into erasing and non-erasing ers. In particular, $\&/\oplus_i$ mixes both an erasing feature (removing one component of the $\&$-box) with a non erasing feature (moving the cut up to the other component of the $\&$-box and the premise of the $\oplus_i$ link).

(ii) There is no clear "head" cut link whose reduction can always be anticipated. More precisely, proof nets yield a non orthogonal rewriting since they have non-trivial critical pairs.

In [Gir87a, Definition 4.24, p. 72] an ers $\pi \to \pi'$ is defined to be non-erasing (*standard* in Girard's terminology) whenever it does not erase other

[5]Girard call this lemma *standardization lemma* [Gir87a, Th. 4.24, p. 72]. We consider this nomenclature quite misleading when related to λ-calculus. We reserve the name "standardisation" for a result much closer to the renowned standardisation theorem of λ-calculus (see also Theorem 12 on resource calculus).

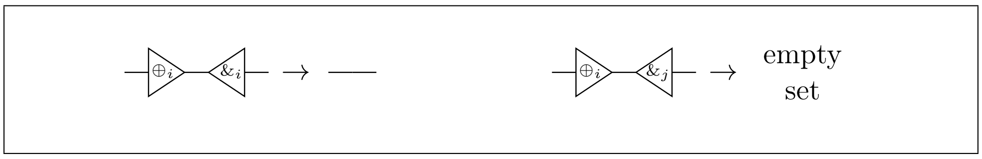
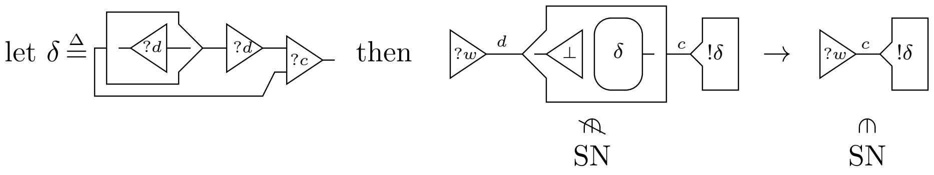


Figure 1.6: additive ers on slices (supposing $i \neq j$).

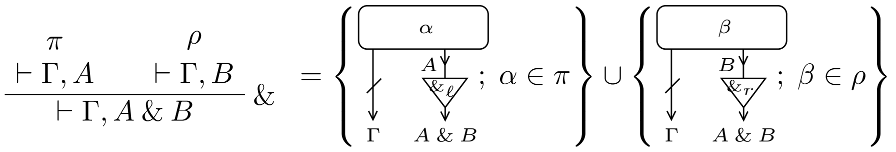
cuts than the fired one. This definition does not work in an untyped setting, as for example:



The reduction of the cut c in the net at the middle moves the box $!\delta$ inside the other box, creating a cut between δ and $!\delta$, that can be easily checked to loop. On the contrary, reducing the weakening cut d in front of the box erases such potential loop, even if it does not erase the cut c.

In fact, one should forbid that $\pi \to \pi'$ erases not just the cuts occurring in π, but also those that can be produced in the future by some cut-elimination sequence (in the example the potential cut between δ and $!\delta$). Danos emended Girard's definition in MELL^2, by defining *erasing* any ers of type $!\pi/?w$. In this way the conservation property can be achieved[6] and hence the strong normalization for MELL^2 [Dan90].

In presence of additives, the same phenomenon occurs, but the solution cannot be that simple, because of the double erasing/non-erasing nature of the ers $\&/\oplus_i$, mentioned in item (i). Our solution consists in using a different syntax for the additives, called sliced proof nets (already introduced in [Gir87a]). The idea is to represent proofs as sets of simple nets, so that a &-rule can be sliced into the disjoint union of the nets associated with its premises:



The $\top$-rule is represented by the empty set of slices.

This change has two main consequences with respect to cut-elimination: first, the $\&/\oplus_i$ rule splits in the two different ers depicted in Figure 1.6, where $\&_j/\oplus_i$ (for $i \neq j$) expresses the erasing feature of $\&/\oplus_i$, and $\&_i/\oplus_i$ the non-erasing one; second, there is no need for the commutative additive

[6]Notice that the cut link d in the above example is in fact of type $!\delta/?w$.

ers (i.e. the ers focusing on a cut which is an auxiliary wire of an additive box or a $\top$-cell). In fact we prove that the cut-elimination becomes confluent in this system (even in an untyped setting). We can then give the correct notion of erasing ers:

Definition 3 ([PTdF10])**.** The *erasing* ers of sliced proof nets are: $\&_j/\oplus_i$ with $i \neq j$ and $!\pi/?w$.

This allows us to state correctly the conservation property:

Theorem 4 ([PTdF10, Th. 4.2])**.** *Given two untyped sliced proof nets π and π', if $\pi \to \pi'$ is not erasing and π' is* SN*, then π is* SN*.*

Because of the previously mentioned problem (ii), the proof is not trivial. We use what Girard refers to as Gandy's method ([Gir87b], referring to [Gan80]), while called Nederpelt lemma by Bezem and Klop ([Ter03], referring to [Ned73]). Namely, we define an increasing measure on non-erasing cut-elimination which together with confluence yields that weak non-erasing normalization implies SN. This is equivalent to Theorem 4.

By reducibility candidates we then show the strong normalization on sliced proof nets, and, from this, we achieve Theorem 2, i.e. the strong normalization of cut-elimination on standard Girard's proof nets.

1.1.2 Experiments in denotational semantics (2007-2008)

During my post-doc in Roma (2007) I was also interested in a study of cut-elimination via the relational semantics of linear logic. The results are published in [dCPTdF11], co-authored with de Carvalho (Ph.D. student at Marseille at that time) and Tortora de Falco.

Relational semantics yields a model of propositional linear logic (see Appendix A, Figure A.1), where formulas are associated with sets and proof nets with relations. Basically, our result shows that the relational interpretations of two cut-free proof nets give an exact account of the number of ers needed to reduce a cut between dual conclusions of the proof nets (Theorem 5).

The result paves the way to a denotational approach to computational complexity — a research direction which I now lead in the richer setting of quantitative semantics (Chapter 3).

The category REL of sets and relations yields a denotational semantics of propositional LL, but not of second order quantifiers (see [BBE06, Tas06]).

REL associates a propositional formula with a set and a proof with a relation over the interpretation of its conclusions. Given a mapping v from propositional variables to sets, the interpretation $[\![A]\!]^v$ is defined by structural induction on A, following the structure of REL described in Figure A.1, Appendix A. Hereafter, we fix an interpretation v of the atoms, and we will omit to explicit it in the notation of $[\![\,]\!]$.

The interpretation of a proof net π with conclusions $A_1, \dots, A_n$ is a relation:

$$\left[\!\!\left[\overset{\pi}{A_1 \dots A_n} \right]\!\!\right] \subseteq [\![A_1]\!] \times \dots \times [\![A_n]\!]$$

which can be defined by structural induction following the denotational interpretation of a sequent proof. In particular, the interpretation of the axiom is the identity of REL and the interpretation of a cut between two proof nets $\pi \vdash \Gamma, A$ and $\rho \vdash A^\perp, \Delta$ is the composition of the relation $[\![\pi]\!] \subseteq [\![\Gamma]\!] \times [\![A]\!]$ and $[\![\rho]\!] \subseteq [\![A^\perp]\!] \times [\![\Delta]\!]$ (recall that $[\![A]\!] = [\![A^\perp]\!]$):

$$[\![\pi]\!] \,;\, [\![\rho]\!] \triangleq \{(b,c) \mid \exists a \in A, (b,a) \in [\![\pi]\!], (a,c) \in [\![\rho]\!]\} \tag{1.3}$$

The main property of a denotational model is to be invariant under cut-elimination. This can be proven either by "brute force", checking that no ers alters the interpretation[7], or by using Girard's notion of experiment [Gir87a, Def. 3.17, p. 58].

Let us restrict to the case of propositional MELL. Experiments are a way of computing the relational semantics of a proof net point-wise, independently from the associated sequent proof. Basically, an experiment e on a

[7]This reasoning is indeed decomposed in more atomic steps showing that REL enjoys the various diagrams defining a $\star$-autonomous Lafont category (or any other categorical axiomatization of a model of LL).

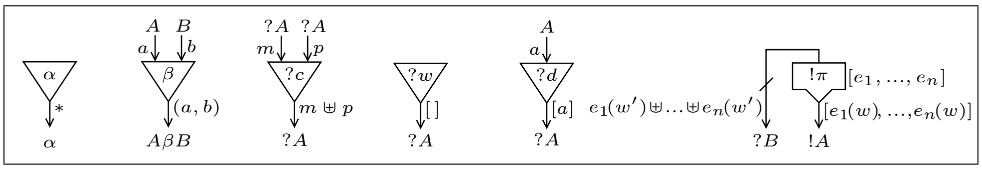


Figure 1.7: labeling of MELL experiments. The symbol α (resp. β) varies over $\{\bot, 1\}$ (resp. $\{\parr, \otimes\}$). In the case of the box $!\pi$, $[e_1, \dots, e_n]$ is a multiset of experiment on the net π associated with the box. Also, we suppose that w (resp. w') is the conclusion of π associated with the principal wire (resp. with an auxiliary wire) of the box, of type A (resp. $?B$).

proof net π, denoted $e : \pi$, is a labeling of the wires of π with elements belonging to the interpretation of the type of the wires (notice that $[\![A]\!] = [\![A^\perp]\!]$, hence the interpretation is independent from the wire orientation), and of the boxes with multisets of experiments over the proof nets inside the boxes. Each cell defines some conditions between the labels associated with the wires incident to it. These conditions are sketched in Figure 1.7, for the fragment of propositional MELL. The result of the experiment $e : \pi$ is the sequence of labels associated with the conclusions of π. In fact, an easy structural induction shows that the semantics of π given by REL corresponds to the set of results of the experiments on π:

$$\left[\!\!\left[\begin{array}{c} \pi \\ a_1 : A_1 \ \dots\ a_n : A_n \end{array} \right]\!\!\right] = \{(e(a_1), \dots, e(a_n)) \mid e : \pi\}. \tag{1.4}$$

One can then prove the invariance of $[\![\pi]\!]$ under cut-elimination by showing that, for any ers $\pi \to \pi'$, for any experiment $e : \pi$ there is an experiment $e' : \pi'$ with the same result as e, and, vice versa, for any $e' : \pi'$ there is one $e : \pi$ with the same result as e'. This correspondence can be checked by an immediate inspection of the rules of Figure 1.5.

The reasoning is actually defining a relation[8] between the experiments in π and those in π'. The starting point of [dCPTdF11] is the remark that one can define a notion of *size of an experiment* such that the size of $e' : \pi'$ decreases constantly with respect to the size of an associated $e : \pi$ when we perform a logical ers, i.e. an ers that creates at most cuts of smaller type, i.e. $\mathbf{1}/\bot$, $\otimes/\parr$, $!\pi/?d$, $!\pi/?w$. This means that the relational interpretation is carrying some quantitative information about the length of a reduction sequence. The goal of [dCPTdF11] has been to show how much precise this information can be.

A proof net is η-long if it has only axioms introducing propositional variables. A stratified cut-elimination sequence is a cut-elimination sequence reducing cuts of increasing exponential depth.

[8]This relation is not even a function. However, if we consider experiments in the category of coherence spaces, this relation becomes even a bijection. See the discussion in [dCPTdF11, Figure 7].

The size $s(a)$ of an element $a \in [\![A]\!]$ is defined inductively as:[9]

$$s(a) \triangleq 0, \text{ if } a \in [\![X]\!] = \left[\!\!\left[X^{\perp}\right]\!\!\right], \qquad s(\star) \triangleq 1,$$
$$s\big((a,b)\big) \triangleq 1 + s(a) + s(b), \qquad s([a_1,\dots,a_n]) \triangleq 1 + \sum_i s(a_i).$$

The set of exhaustive elements of $[\![A]\!]$ is the subset $[\![A]\!]^{ex}$ of those elements having no occurrence of the empty multiset at a position corresponding to a sub-formula of A of !-type. More formally, $[\![A]\!]^{ex}$ can be inductively defined by interpreting $!A$ with $[\![!A]\!]^{ex} \triangleq \mathcal{M}_{\mathrm{f}}\left([\![A]\!]^{ex}\right) \setminus \{[\,]\}$ (but keeping $[\![?A]\!]^{ex} \triangleq \mathcal{M}_{\mathrm{f}}\left([\![A]\!]^{ex}\right)$).

I think that the following theorem gives an clear idea on the precision of REL in describing the cost of eliminating a cut in a proof net.

Theorem 5 (from [dCPTdF11])**.** *Given two cut-free and η-long proof nets $\pi \vdash \Gamma, A$ and $\pi' \vdash A^{\perp}, \Delta$, let us denote by $\langle \pi|_A\pi' \rangle$ the proof net obtained by cutting the A conclusion of π with the $A^{\perp}$ conclusion of π'. The number of logical ers in a stratified cut-elimination sequence from $\langle \pi|_A\pi' \rangle$ to a normal form is*

$$\inf\{s(a) \mid \exists c \in [\![\Gamma]\!]^{ex}, \exists d \in [\![\Delta]\!]^{ex}, (c,a) \in [\![\pi]\!], (a,d) \in \left[\!\!\left[\pi'\right]\!\!\right]\}.$$

The result has not much to do with propositional formulas. Indeed, the paper [dCPTdF11] deals with MELL untyped proof nets. We define experiments on a set D which is a solution in REL to the recursive equation

$$D = \mathcal{A} \oplus (D \otimes D) \oplus (D \parr D) \oplus !D \oplus ?D,$$

where $\mathcal{A}$ is a set of atoms. One then gets a statement similar to Theorem 5, but with the difference that now π and π' cannot be supposed to be η-long (a meaningless notion in an untyped setting). This makes the semantic quantity more complex to define, in particular one has to introduce a notion of atomic substitution, expressing the intuition that any element of $\mathcal{A}$ is just a marker for a kind of infinite η-expansion (analogous to the notion of *fax* in [Gir01]). See [dCPTdF11, Theorem 38], for more details.

On intersection types

This analysis generalizes to MELL the study achieved by De Carvalho about the characterization via non-idempotent intersection types of the number of steps needed by Krivine machine to evaluate a λ-term [dC07, dC09]. In fact,

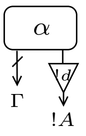
[9]The definition of $s(a)$ is different from the original one in [dCPTdF11] in the case a is an atom, i.e. $a \in [\![X]\!]$. This is due to the fact that [dCPTdF11] uses a syntax with explicit axiom ers.

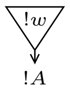
an element of the REL interpretation of the formula $!A \multimap B$ (expressing in MELL the functional type of λ-calculus) can be seen as an intersection type:

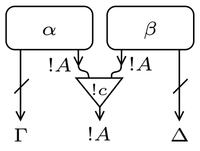
$$([a_1, \dots, a_n], b) \text{ can be written as } a_1 \wedge \cdots \wedge a_n \to b \tag{1.5}$$

An experiment, then, is a type derivation, i.e. a decoration of the syntactic tree of a term by intersection types.

The intersection is non-idempotent (i.e. $a \wedge a \neq a$) because REL denotes $!A$ as a set of finite multisets, hence $[a, a] \neq [a]$. The lack of idempotency is the key ingredient to model resource sensitiveness — while the usual idempotent intersection $M : a \wedge b$ stands for "M can be used either as data of type a or as data of type b", when the intersection is not idempotent the meaning of $M : a \wedge b$ becomes "M will be called *once* as data of type a and *once* as data of type b". Hence, types should no longer be understood as *sets of terms*, but rather as *sets of calls* to terms.

co-dereliction: $$\dfrac{\overset{\pi}{\vdash \Gamma, A}}{\vdash \Gamma, !A}\,!d \quad = \sum_{\alpha\in\pi}$$ 

co-weakening: $$\dfrac{}{\vdash !A}\,!w \quad =$$ 

co-contraction: $$\dfrac{\overset{\pi}{\vdash \Gamma, !A} \qquad \overset{\rho}{\vdash !A, \Delta}}{\vdash \Gamma, !A, \Delta}\,!c \quad = \sum_{\alpha\in\pi}\sum_{\beta\in\rho}$$ 

zero: $$\dfrac{}{\vdash \Gamma}\,0 \quad = \mathbf{0}$$

sum: $$\dfrac{\overset{\pi}{\vdash \Gamma} \qquad \overset{\rho}{\vdash \Gamma}}{\vdash \Gamma}\,+ \quad = \pi+\rho$$

Figure 1.8: sequent rules defining the differential extension of linear logic, together with their translation into differential proof nets.

1.2 Preliminaries on Differential Linear Logic

The differential extension of linear logic (DILL, for short) has been introduced in [ER06b]. I refer also to [Ehr11] for an up-to-date account of DILL. Like linear logic, this extension is mainly motivated by denotational semantics. Namely, DILL lifts to the syntax constructions that are natural in Köthe sequences spaces and finiteness spaces — two vectorial based semantics introduced by Ehrhard in [Ehr02, Ehr05].

DILL adds to linear logic the sequent rules of Figure 1.8: three further rules are associated with the ! modality (co-dereliction, co-weakening, co-contraction), dual with respect to the rules associated with the ? modality; two other rules (zero, sum) are added in order to linearly combining proofs over the semi-ring of natural numbers[10].

In this chapter, I will focus on the propositional multiplicative exponential fragment, so that DILL will denote henceforth the MELL rules of Figure 1.2, and the rules of Figures 1.4 and 1.8.

DILL cannot be thought of as a "new logic", in fact it collapses at the level of provability: any formula is provable by a zero rule, or (in case of a ! formula) by a co-weakening. Besides denotational semantics, the interest of DILL relies in its cut-elimination and in the Curry-Howard correspondence

[10]One can imagine to consider a different semi-ring $\mathcal{R}$ by adding a scalar multiplication rule. This will in fact happen in Section 3.2, in the grammar defining $\mathcal{R}$-weighted PCF.

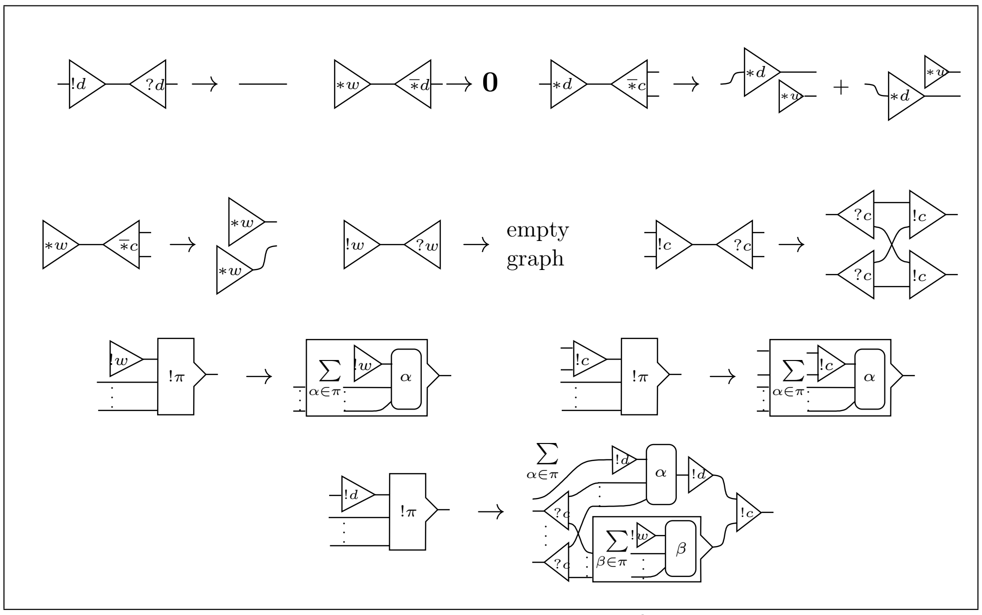


Figure 1.9: the elementary reduction steps specific to DILL cut-elimination.

with non-deterministic programming primitives.

As it should be clear to the reader now, the favorite way for describing cut-elimination is via a proof net syntax. I call the nets corresponding to the sequent calculus of DILL differential proof nets, they can be defined by extending the definition of MELL proof nets (Figure 1.2) to the new rules of DILL, Figure 1.8. The main difference is that now nets are finite sums of simple nets with the same conclusions, modeling the sequent rules 0 and + in a way analogous to how additive slices represent the $\top$ and & rules of linear logic (Section 1.1.1).

Correctness criteria for differential nets are quite easy. In fact, DILL sequent calculus is so liberal that its image within the set of nets can be characterized via a simple adaptation of Danos and Regnier's *switching acyclicity*.

A path in a differential net π is a path in one simple net of π. A path is called switching whenever it does not contain both auxiliary wires of a cell of type $\otimes$ or $!c$. A net is switching acyclic whenever it contains no switching cycle and all its boxes are switching acyclic.

Proposition 6 ([ER06b])**.** *A differential net π is switching acyclic iff it is a proof net of* DILL*.*

The cut-elimination procedure of the new cuts is defined by the ers given in Figure 1.9. Let me give some computational and mathematical intuitions.

Co-dereliction can be seen both as a constructor making a *one-use* resource of type A able to interact with a query of type $!A$, i.e. a query of a number of resources of type A, as well as, semantically, co-dereliction is the derivation at 0 of a smooth function of domain A. Hence, we have:

$!d/?d$
- expresses the case of a single query meeting a one-use resource: simply, they communicate;
- is the derivation of a linear function, which is the function itself;

$!d/?w$
- reduces to the empty sum, representing a deadlock arisen from a resource that has to be used once but which is not asked;
- is the derivation of a constant function, which is 0;

$!d/?c$
- describes what happens when a one-use resource is contended by two queries: only one query will get the resource, the other one will get a co-weakening, i.e. the absence of resources. In fact, the contractum is the sum gathering the two possibilities;
- expresses the equality between the derivation of $f(x,x)$ and the derivation of $f(x,0) + f(0,x)$;

The ers dealing with dereliction are symmetric and carry similar intuitions. The ers between (co)-weakenings and (co)-contractions ($!w/?w$, $!w/?c$, $!c/?w$, $!c/?c$) describe a kind of non-deterministic routing, obeying to the bi-algebra laws.

The ers $!d/!\pi$ is quite delicate from a rewriting theory point of view, but it expresses what happens when a one-use resource is asked by a reusable one (the box $!\pi$): just one copy of $!\pi$ will get it. Also, this rule corresponds to the usual chain rule of differentiation for computing the derivative of the composition of two functions (one represented by π and the other one by what will be wired to the principal wire of $!\pi$).

DILL cut-elimination modifies the viewpoint on linear logic exponentials: not only these connectives yields a logical status to the operations of duplication and deletion, but also they model, thanks to the differential rules, a kind of communication between proofs which is similar to the one described in process calculi ([EL07]). What makes DILL special with respect to other process algebra is to relate such computational intuitions with the mathematical notion of derivative.

1.2.1 The question of semantic correctness (2008-2009)

Experiments (Section 1.1.2) give a relational interpretation to any net, even if not associated with a correct proof. May we express logical correctness by a semantic criterion on REL, adding some structure on its objects?

Various semantics can be presented on top of REL in order to answer such a question. In particular, finiteness spaces [Ehr05] can be presented as a refinement of REL providing a notion of *finitary relation*. During my stay in Paris 7 (2008-2009), I studied the link between finitary relations and logical correctness. I showed that the property of having a finitary interpretation is strictly weaker than the property of being a correct differential net. Moreover, I gave a syntactic characterization of the former, by relaxing Danos and Regnier's switching acyclicity into the notion of *visible acyclicity* (Definition 8 and Theorems 9 and 10). The result has been published in [Pag12] and extends to DILL a similar result achieved during my Ph.D. in the setting of (non-uniform) coherence spaces and MELL nets ([Pag06]).

Let me underline that this issue is related with Abramsky and Jagadeesan's notion of full completeness [AJ94]. A model M is fully complete for a logic L whenever every morphism in the model denotes some proof in L. Full completeness is a very strong property, meaning that the model M expresses a true syntax-independent description of the proofs spanned by L. The syntax of nets allows us to decompose full completeness in two independent properties: (i) a characterization of the denotations of nets, and (ii) a semantical description of logical correctness.

I recall the multiplicative exponential structure of the category FIN of finiteness spaces and finitary relations, see [Ehr05] for more details.

Let A be a set and S be a set of subsets of A, i.e. $S \subseteq \mathcal{P}(A)$. The polar of S is defined by:

$$S^{\perp} \triangleq \{u \subseteq A \,|\, \forall v \in S, u \cap v \text{ is a finite set}\}. \tag{1.6}$$

Definition 7 (The category FIN, [Ehr05])**.** The objects are finiteness spaces, i.e. pairs $\mathcal{A} = (|\mathcal{A}|, \mathrm{F}(\mathcal{A}))$ of a set $|\mathcal{A}|$, called the web, and a set $\mathrm{F}(\mathcal{A})$ of subsets of $|\mathcal{A}|$ such that $\mathrm{F}(\mathcal{A})^{\perp\perp} = \mathrm{F}(\mathcal{A})$. A morphism from a finiteness space $\mathcal{A}$ to a finiteness space $\mathcal{B}$ is a relation f between the webs of the spaces which is finitary, i.e. such that

(i) $\forall u \in \mathrm{F}(\mathcal{A})$, $f(u) \triangleq \{b \,|\, \exists a \in u, (a, b) \in f\} \in \mathrm{F}(\mathcal{B})$, and

(ii) $\forall v' \in \mathrm{F}(\mathcal{B})^{\perp}$, $f^t(v') \triangleq \{a \,|\, \exists b \in v', (a, b) \in f\} \in \mathrm{F}(\mathcal{A})^{\perp}$.

FIN yields a model of DILL (hence of LL). Formulas are interpreted as finiteness spaces (once fixed an interpretation on the variables) and correct

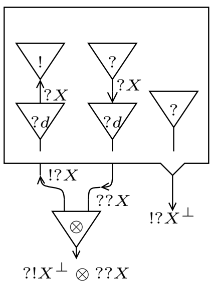


Figure 1.10: example of logically incorrect net associated with a finitary relation in FIN. Notice that the cycle crossing the $\otimes$ cell is switching but not visible (Definition 8).

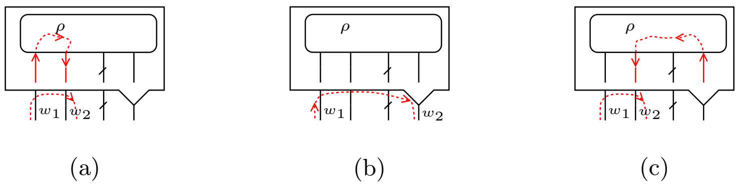


Figure 1.11: the three cases defining a visible passage through a box.

proofs as *finitary* relations on the interpretation of their conclusions. Actually, FIN can be seen as a refinement of REL, namely any DILL proof is associated with exactly the same relation in both models, the difference being that FIN can say moreover that such relation is finitary.

Given a formula A, the definition of the web of $[\![A]\!]$ is defined as in REL (see Appendix A, Figure A.1), while the definition of the set $\mathrm{F}([\![A]\!])$ is given as follows: $\mathrm{F}([\![\mathbf{1}]\!]) \triangleq \{*\}$, $\mathrm{F}([\![A \otimes B]\!]) \triangleq \{u \times v \mid u \in \mathrm{F}([\![A]\!])\,, v \in \mathrm{F}([\![B]\!])\}^{\perp\perp}$, $\mathrm{F}([\![!A]\!]) \triangleq \{\mathcal{M}_{\mathrm{f}}(u) \mid u \in \mathrm{F}([\![A]\!])\}^{\perp\perp}$, otherwise $\mathrm{F}\big([\![A^{\perp}]\!]\big) \triangleq \mathrm{F}([\![A]\!])^{\perp}$.

Experiments (Section 1.1.2, Figure 1.7) are extended to differential nets by associating with the new cells of DILL (co-weakening, co-dereliction, co-contraction) the same conditions as for the corresponding dual cells (weakening, dereliction and contraction, Figure 1.7). The interpretation $[\![\pi]\!]$ of a differential net π is then defined as the set of the results of the experiments on any simple net in π.

Ehrhard proves that the property of being finitary is preserved by the sequent calculus rules of DILL. Hence, $[\![\pi]\!]$ is a finitary relation whenever π is a DILL proof net. Unfortunately, the converse implication fails: there are DILL nets which are finitary but not proof nets (e.g. Figure 1.10). The following definition lifts then to the syntax the property of being finitary.

Definition 8 ([Pag12, Def. 2.5])**.** The set of visible paths in a net π is defined by induction on the exponential depth of π. If π has no exponential box, then a visible path is a switching path in a simple net of π. Otherwise, a path is visible if it is switching and whenever it crosses an exponential box $!\rho$ from a wire w_1 to a wire w_2, one of the following conditions holds (see Figure 1.11):

(a) either there is a visible path in ρ from the conclusion associated with w_1 to the conclusion associated with w_2,

(b) or w_2 is the principal wire of the box $!\rho$,

(c) or there is a visible path in ρ from the conclusion associated with the principal conclusion of the box $!\rho$ to w_2.

A net is visibly acyclic whenever it has no visible cycle nor a box containing a visibly cyclic net.

Theorem 9 ([Pag12, Th. 3.3])**.** *Let π be a* DILL *net. If π is visibly acyclic, then $[\![\pi]\!]$ is a finitary relation for any interpretation of the variables.*

Theorem 10 ([Pag12, Th. 4.5])**.** *Let π be a* DILL *cut-free net*[11]*. If π is a finitary relation for any interpretation of the variables, then π is visibly acyclic.*

Let me give a glance at the proofs of the theorems. Consider a net π with two conclusions: $a : A$ and $b : B$. Suppose that π is visibly acyclic, and let us prove that $[\![\pi]\!]$ is finitary. That means that we must prove the above two conditions (i), (ii) defining finitary relations (Definition 7). The two conditions are symmetric, and they can be rephrased in: for any family of experiments $\{e_i\}_{i\in I}$ on π, if the set of their values on one conclusion, say A, is anti-finitary in $[\![A]\!]$, i.e. $\{e_i(a)\}_{i\in I} \in \mathrm{F}([\![A]\!])^{\perp}$, then the set of its values on the other conclusion is finitary in $[\![B]\!]$, i.e. $\{e_i(b)\}_{i\in I} \in \mathrm{F}([\![B]\!])$. So, suppose $\{e_i(a)\}_{i\in I} \in \mathrm{F}([\![A]\!])^{\perp}$, the proof proceeds by defining a path ϕ in π, starting from a and following the values of the family $\{e_i\}_{i\in I}$ on the wires, going upward whenever the values of $\{e_i\}_{i\in I}$ are anti-finitary, or going downward whenever the values of $\{e_i\}_{i\in I}$ are finitary. In fact, this path is visibly acyclic and hence cannot be a cycle, so it must stop into b, proving then $\{e_i(b)\}_{i\in I} \in \mathrm{F}([\![B]\!])$.

The proof of Theorem 10 basically uses the inverse reasoning, constructing from a visible cycle in π a family of experiments witnessing that $[\![\pi]\!]$ is not finitary.

The two proofs adapt to DILL techniques developed by Girard [Gir87a] and Retoré [Ret97] in order to prove a correspondence between switching acyclicity of MLL nets and cliques of coherence spaces.

[11] Actually, the right hypothesis supposes π to be a value, i.e. a cut-free net without some very pathological cycles, called *vicious cycles* in [Laf90].

Open problem 1 ($\star\star\star$)**.** *Moving from switching acyclicity to visibly acyclicity is already needed in the framework of linear logic nets, as one considers exponential boxes: the* MELL *nets corresponding to cliques in (non-uniform) coherence spaces are the visibly acyclic nets, as I proved in [Pag06]. Is it possible to refine coherence and/or finiteness spaces in order to capture logical correctness (at least of* MELL *or* DILL *plus empty and mix rules)? Conversely, do we have a sequent calculus generating the set of visibly acyclic nets? Does visibly acyclicity have any sense from a logical point of view?*

Open problem 2 ($\star\star$)**.** *In presence of additives, let me mention Tranquilli's* hypercorrectness *[Tra08], a fine criterion on* MALL *sliced nets corresponding to the semantic correctness of hypercoherence spaces [Ehr95] (a refinement of coherence spaces able to catch the strongly stable functions) — it remains an open question whether hypercorrectness is equivalent to the correctness induced by* MALL *sequent calculus. Precisely, the question is whether Hughes and van Glabbeek's correctness criterion on additive slices [HvG03] is equivalent to hypercorrectness, or not.*

1.2.2 Finiteness spaces and safe interaction (since 2009)

Finiteness spaces do not model fix-point combinators, namely FIN is not cpo-enriched. Indeed, Ehrhard supposes strict links between the property of being finitary and the termination of cut-elimination. I think that visibly acyclic differential nets are the right framework to study and formalize this intuition. More precisely, define a set S of DILL cut-free nets to have a safe interaction, if any cut between two nets in S is strongly normalizable.

Conjecture 1. *The set of simply typed visibly acyclic nets is the maximum set of differential nets containing* DILL *proof nets and having a safe interaction.*

Together with Theorems 9 and 10, this conjecture would show that finiteness spaces define the "closure" of DILL with respect to safe interaction.

An easy consequence of [Pag09] and [PT11] is that the set of visibly acyclic nets enjoys safe interaction. Namely, the two papers achieve, respectively, a weak normalization theorem and a conservation theorem of DILL proof nets. The proof of the maximality of visible acyclicity is a work in progress.

Conjecture 1 can be obtained in two steps:

(i) by proving that cut-elimination is strongly normalizing on simply typed visibly acyclic nets;

(ii) by showing the maximality of the set of visibly acyclic nets, i.e. by showing that for any simply typed visibly cyclic π, there exists a simply typed visibly acyclic net ρ such that the cut between π and ρ yields an infinite reduction sequence.

Item (i) amounts in fact to extend to differential nets the (already complex) techniques developed in [PTdF10] for achieving the strong normalization theorem of LL^2 (see Section 1.1.1). In collaboration with Tranquilli (PhD student at Paris 7 at that time), we succeeded in this feat. The results have been published in [Pag09, PT11] for the restricted set of DILL proof nets (i.e. switching acyclic nets), but the proofs directly generalize to visibly acyclic nets.

In [Pag09], I prove the weak normalization theorem for propositional DILL. The reasoning is quite standard and amounts to define a measure and a reduction strategy making this measure to decrease. A different technique based on reducibility candidates is given by Gimenez in [Gim11].

The paper [PT11] is much more involved, proving a conservation theorem for untyped differential proof nets: weak non-erasing normalization implies strong normalization [PT11, Th. 32]. The proof is considerably subtler than the one for linear logic proof nets. In particular, cut-elimination in DILL fails to give a confluent rewriting, not even locally. One needs to introduce some equivalences, namely associativity of contractions and co-contractions (see [Tra09]), and working in an equational rewriting system [Ter03, §14.3]. The strong normalization theorem concludes the paper ([PT11, Th. 35]), and allows us to achieve item (i) in the proof of Conjecture 1.

As for item (ii), the proof should adapt a technique used by Béchet for proving a similar safeness property of MLL proof nets [Béc98]. The technique consists in defining a DILL proof net ρ by looking at a visible cycle in π such that the cut between π and ρ yields an infinite reduction sequence. It is in some sense a syntactical version of the reasoning used in the proof of Theorem 10. This is a work in progress.

Chapter 2

Linear resources in a functional setting

- P. Boudes, F. He, M. Pagani. *A Characterization of the Taylor Expansion of the Lambda-Terms.* Proceedings of the 22nd EACSL Annual Conference Computer Science Logic (CSL 2013), Ronchi della Rocca ed., LIPICS, 2013.
- A. Diaz-Caro, G. Manzonetto, M. Pagani. *Call-by-Value Non-determinism in a Linear Logic Type Discipline.* International Symposium on Logical Foundations of Computer Science (LFCS 2013), Artemov and Nerode ed., LNCS, pp. 164-178, 2013.
- G. Mazonetto, M. Pagani. *Böhm's Theorem for Resource Lambda Calculus through Taylor Expansion.* 10th International Conference on Typed Lambda Calculi and Applications (TLCA 2011), Ong ed., Springer ARCoSS, pp. 153-168, 2011.
- M. Pagani, S. Ronchi Della Rocca. *Linearity, Non-determinism and Solvability.* Fundamenta Informaticae, vol. 103, num. 1-4, 2010.
- M. Pagani, S. Ronchi della Rocca. *Solvability in Resource Lambda-Calculus.* 13th International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2010, a member conference of ETAPS 2010), Ong ed., Springer LNCS 6014, 2010.
- M. Pagani, P. Tranquilli. *Parallel Reduction in Resource Lambda-Calculus.* 7th Asian Symposium on Programming Languages and Systems (APLAS 2009), Hu ed., Springer LNCS 5904, pp. 226-242, 2009.
- M. Pagani. *The Cut-Elimination Theorem for Differential Nets with Boxes.* 9th International Conference on Typed Lambda Calculi and Applications (TLCA 2009), Curien ed., Springer LNCS 5608, pp. 219-233, 2009.

- M. Pagani, C. Tasson. *The Taylor Expansion Inverse Problem in Linear Logic*. 24th Annual IEEE Symposium on Logic in Computer Science (LICS 2009), Pitts ed., IEEE, pp. 222-232, 2009.

$$\begin{array}{rll}\text{terms:} & M, N, L & ::= x \mid \lambda x.M \mid MP \\ \text{bags:} & P, Q, R & ::= 1 \mid P \cdot Q \mid [M] \mid [\mathfrak{M}^!] \\ \text{sums of terms:} & \mathfrak{M}, \mathfrak{N}, \mathfrak{L} & ::= 0 \mid M \mid \mathfrak{M} + \mathfrak{N}\end{array}$$

Figure 2.1: syntax of the resource calculus Λ^r.

2.1 Preliminaries

The minimal fragment of DILL (Section 1.2) expressing the intuitionistic arrow (i.e. $A \to B = !A \multimap B$) can be described by a term calculus, called resource calculus. The resource calculus, Λ^r for short, is an extension of untyped λ-calculus defined by the grammar in Figure 2.1. We have the variable abstraction, corresponding to the introduction of $\to$, and the application, corresponding to the elimination of $\to$. However, the argument of an application in Λ^r is a bag, i.e. a finite multiset of resources, that are terms either linear (to be used exactly once) or not (can be copied or erased).

Bags express the differential rules of DILL: the empty bag 1 is co-weakening, the disjoint union $P \cdot Q$ of two bags P and Q is co-contraction, the singleton $[M]$ of a linear resource is co-dereliction and the singleton $[\mathfrak{M}^!]$ of a reusable resource is promotion. Bags are considered as multisets in multiplicative notation, hence $\cdot$ is supposed commutative and associative, with 1 as neutral element.

The evaluation of a resource redex $(\lambda x.M)P$ gives rise to different possible choices, because of the different possibilities of distributing the resources in P among the occurrences of x in M. Like DILL cut-elimination, the operational semantics expresses this non-determinism internally, by reducing a single term to a finite formal sum of terms, representing all possible results of a computation. Hence, Λ^r has sums of terms as a third syntactical sort. For technical convenience, the grammar prevents terms from having sums within its syntactic tree except under the scope of a bang. However, as syntactic sugar, all the constructors can be extended to sums by (multi)linearity (Figure 2.2(c))[1].

The reduction $\to$ is defined by the context closure of the elementary reduction steps in Figure 2.2(a), where $\{0/x\}$ and $\{N + x/x\}$ refers to the usual λ-calculus substitution extended to sums, whilst $\langle N/x \rangle$ is the linear substitution, i.e. a substitution replacing non-deterministically one *linear occurrence* of x, if it exists, otherwise returning 0. The definition of $\langle N/x \rangle$ is in Figure 2.2(b), the crucial case being $[\mathfrak{M}^!]\langle N/x \rangle = [\mathfrak{M}\langle N/x \rangle, \mathfrak{M}^!]$: in order to assure that an occurrence of x will be used linearly in a reusable

[1]Actually, in the original papers here discussed, we distribute sums also under the bang, by the equation $[(M + N)^!] = [M^!] \cdot [N^!]$. I think however that the present syntax gives a more uniform and direct presentation of the results listed in the chapter. Anyway, this is a matter of notation, and we should not give it too much attention here.

$$(\lambda x.M)1 \to M\{0/x\}, \qquad (\lambda x.M)[N]\cdot P \to (\lambda x.M\langle N/x\rangle)P,$$
$$(\lambda x.M)[\mathfrak{N}^!]\cdot P \to (\lambda x.M\{\mathfrak{N}+x/x\})P,$$

(a) Elementary reduction steps (ers) of Λ^r (assuming x not free in N).

$$y\langle N/x\rangle \triangleq \begin{cases} N & \text{if } y = x, \\ 0 & \text{otherwise,} \end{cases} \qquad (\lambda y.M)\langle N/x\rangle \triangleq \lambda y.(M\langle N/x\rangle),$$
$$(MP)\langle N/x\rangle \triangleq M\langle N/x\rangle P + M(P\langle N/x\rangle),$$
$$[M]\langle N/x\rangle \triangleq [M\langle N/x\rangle], \qquad 1\langle N/x\rangle \triangleq 0,$$
$$[\mathfrak{M}^!]\langle N/x\rangle \triangleq [\mathfrak{M}\langle N/x\rangle, \mathfrak{M}^!], \qquad (P\cdot R)\langle N/x\rangle \triangleq P\langle N/x\rangle\cdot R + P\cdot R\langle N/x\rangle,$$
$$(\textstyle\sum_i M_i)\langle N/x\rangle \triangleq \sum_i (M_i\langle N/x\rangle).$$

(b) Linear substitution, in the abstraction case we suppose $y \notin \mathrm{FV}(N)\cup\{x\}$.

$$\lambda x.(\textstyle\sum_i M_i) \triangleq \sum_i \lambda x.M_i \qquad (\sum_i M_i)P \triangleq \sum_i M_i P \qquad M(\sum_i P_i) \triangleq \sum_i MP_i$$
$$[(\textstyle\sum_i M_i)]\cdot P \triangleq \sum_i [M_i]\cdot P$$

(c) Distribution of sums as syntactic sugar.

Figure 2.2: operational semantics, linear substitution and notational conventions of Λ^r.

resource $\mathfrak{M}^!$, one has to extract a linear copy of $\mathfrak{M}$. Notice the correspondence with the $!d/!\pi$ ers in differential nets (Figure 1.9).

The resource calculus is a slight variant of Ehrhard and Regnier's *differential λ-calculus* [ER03], introduced by Tranquilli in [Tra11] in order to have a precise Curry-Howard correspondence with DILL proof nets.[2] The idea of extending λ-calculus with a resource sensitive application actually

[2]In short, differential λ-calculus is characterized by allowing a linear application $\mathtt{D}M\cdot N$ of the derivative of a term M to another term N. This application can be expressed in Λ^r via an η-expansion and a linear bag, vice versa, linear bags can be represented via a derivative at 0:

differential λ-calculus		resource calculus	
$\mathtt{D}M\cdot N$	$=$	$\lambda x.M[N,x^!]$,	$x\notin FV(M)\cup FV(N)$
$(\mathtt{D}M\cdot N)0$	$=$	$M[N]$.	

The difference between the two calculi is mainly a matter of notation: the resource notation underlines the computational behavior of co-dereliction, whilst the differential λ-calculus focuses on the correspondence with the derivative in calculus.

dates back to, at least, Boudol's [Bou93], where a resource calculus is introduced in order to study Milner's encoding of the lazy λ-calculus into the π-calculus [BL96, BCL99]. The fact that two different research lines (linear logic semantics on the one side and process algebras on the other side) meets so naturally in this point, is a clear clue of a common framework.

What can we program within resource calculus?

For sure, we have the untyped λ-calculus, the usual application MN being represented by $M[N^!]$. For example, we have loops, like the typical $\mathbf{\Omega} \triangleq \mathbf{\Delta}[\mathbf{\Delta}^!]$, with $\mathbf{\Delta} \triangleq \lambda x.x[x^!]$:

$$\begin{aligned}\mathbf{\Omega} &\to (\lambda x.(\mathbf{\Delta}+x)[(\mathbf{\Delta}+x)^!])1 = (\lambda x.\mathbf{\Delta}[(\mathbf{\Delta}+x)^!])1 + (\lambda x.x[(\mathbf{\Delta}+x)^!])1\\ &\twoheadrightarrow \mathbf{\Delta}[(\mathbf{\Delta}+0)^!] + 0[(\mathbf{\Delta}+0)^!] = \mathbf{\Delta}[\mathbf{\Delta}^!] = \mathbf{\Omega},\end{aligned}$$

as well as fix-point operators, like $\mathbf{\Theta} \triangleq \lambda f.\mathbf{\Theta}_f[\mathbf{\Theta}_f^!]$, with $\mathbf{\Theta}_f \triangleq \lambda x.f[(x[x^!])^!]$:

$$\mathbf{\Theta}[g^!] \twoheadrightarrow \mathbf{\Theta}_g[\mathbf{\Theta}_g^!] \to (\lambda x.g[((\mathbf{\Theta}_g+x)[(\mathbf{\Theta}_g+x)^!])^!])1 \to g[(\mathbf{\Theta}_g[\mathbf{\Theta}_g^!])^!].$$

Since bags can be composed by disjoint union, we can define non-deterministic choice operators, such as:

$$M \oplus N \triangleq (\lambda x.x)[(M+N)^!] \tag{2.1}$$

Indeed, we have: $(\lambda x.x)[(M+N)^!] \to (\lambda x.(M+N+x))1 \twoheadrightarrow\!\!\to M+N$.

We denote by $\Lambda^{\oplus}$ the <u>purely non-deterministic calculus</u>, obtained by forbidding in the grammar of Figure 2.1 the empty and linear bags (1 and $[M]$) and the empty sum 0.

Notice that the operational semantics is call-by-name, hence choices can be duplicated before being evaluated:

$$\begin{aligned}\mathbf{\Delta}[(M\oplus N)^!] &\to (\lambda x.(x+M\oplus N)[(x+M\oplus N)^!])1\\ &= (\lambda x.x[(x+M\oplus N)^!])1 + (\lambda x.(M\oplus N)[(x+M\oplus N)^!])1\\ &\twoheadrightarrow 0[(0+M\oplus N)^!] + (M\oplus N)[(0+M\oplus N)^!]\\ &= (M\oplus N)[(M\oplus N)^!].\end{aligned}$$

The most striking feature of Λ^r is the presence of linear resources. This introduces potential deadlocks in the evaluation of a term, represented by the empty sum 0. For example, taking $\mathbf{I} = \lambda y.y$:

$$\begin{aligned}\mathbf{\Delta}[\mathbf{I}] &\to (\lambda x.(x[x^!])\langle \mathbf{I}/x\rangle)1 = (\lambda x.\mathbf{I}[x^!])1 + (\lambda x.x[\mathbf{I},x^!])1\\ &\to \mathbf{I}1 + (\lambda x.x[\mathbf{I},x^!])1 \to 0 + (\lambda x.x[\mathbf{I},x^!])1 = (\lambda x.x[\mathbf{I},x^!])1 \to 0[\mathbf{I}] = 0.\end{aligned}$$

Open problem 3 ($\star$)**.** *The operational semantics is call-by-name because we are considering the call-by-name translation of the intuitionistic arrow* *into linear logic: $A \to B = !A \multimap B$. A translation associated with the call-by-value evaluation strategy has been also given in [Gir87a, §5.1]: $A \to B = !(A \multimap B)$. What is the term calculus associated with the minimal* DILL *fragment of $!(A \multimap B)$? How is the grammar of terms and bags? How do sums distribute? Which kind of solvability (see Section 2.3) does this calculus express? The $!$-free fragment of this calculus should be similar to the one given in [Ehr12a].*

2.2 A standard operational semantics (2009)

A noteworthy feature of resource calculus (as well as of the differential λ-calculus) is to enjoy many rewriting properties of the regular β-reduction, although the calculus is much more expressive than λ-calculus (it has non-determinism and linear resources). During my post-doc in Turin (2009), I started to work on resource calculus by proving, in collaboration with Tranquilli (PhD student at Paris 7 at that time), the properties of confluence and of a form of standardization [PT09b].

Theorem 11 ([PT09b])**.** *The reduction $\to$ is confluent, i.e. for every $\mathfrak{M}' \overset{*}{\leftarrow} \mathfrak{M} \overset{*}{\to} \mathfrak{M}''$ there is $\mathfrak{M}' \overset{*}{\to} \mathfrak{M}''' \overset{*}{\leftarrow} \mathfrak{M}''$.*

Confluence does not contradict non-determinism because the result of a non-deterministic reduction is a sum of terms. It remains meaningful, as it states that non-determinism is really internal and not caused by what an evaluator chooses to reduce. The theorem is achieved by adapting the technique by Tait and Martin-Löf, using a suitable notion of parallel reduction. A similar result is in [ER03], where confluence of differential λ-calculus is proven. However our proof is somewhat simpler, using a notion of *development* as defined by Takahashi [Tak95] for λ-calculus.

As for standardization, an ers is inner $\to_i$ if the redex to be contracted is under the scope of a bang, otherwise it is outer $\to_o$. Standardization then states that every reduction sequence can be split into a concatenation of a sequence of outer steps and a sequence of inner steps.

Theorem 12 ([PT09b])**.** *For every $\mathfrak{M} \overset{*}{\to} \mathfrak{M}'$, there exists $\mathfrak{M} \overset{*}{\to}_o \mathfrak{M}'' \overset{*}{\to}_i \mathfrak{M}'$.*

Our proof of standardization adapts the λ-calculus one given by Takahashi and based on parallel reduction and inner parallel reduction [Tak95]. Actually our notion of inner parallel reduction is quite peculiar, possibly yielding a slight generalization of the technique by Takahashi.

2.3 Solvability (2009, 2013)

A closed λ-term is solvable whenever there are $P_1, \ldots, P_n$ such that:

$$MP_1 \ldots P_n \xrightarrow{*}_\beta \lambda x.x. \tag{2.2}$$

This notion has been defined by Barendregt in [Bar71], in order to have a satisfactory representation of partial recursive functions. An equivalent definition has been given independently by Wadsworth in [Wad71] with the notion of head-normalizable[3].

My post-doc in Torino (2009) was dedicated to study the solvability in resource calculus. Because of the non-deterministic behavior of Λ^r, two different notions of solvability arise, one optimistic (angelical, may) and one pessimistic (demoniac, must).

The study was developed in collaboration with Ronchi della Rocca (professor at Torino) and yielded the extended abstract [PRDR10b] and its long version [PRDR10a]. Basically, the goal was to characterize the may and must solvability both operationally, with suitable variants of Wadsworth's notion of head-normalizable term, and semantically, with intersection type systems giving a type to all and only the (may, must) solvable terms.

Finally, in the more recent [DCMP13], written in collaboration with Diaz-Caro (at that time post-doc at Paris 13) and Manzonetto (assistant professor at Paris 13), we give a similar characterization of solvability in a call-by-value extension of λ-calculus with both a may-convergent sum and a must-convergent parallel operator.

We detail here the results in [PRDR10b, PRDR10a].

A closed resource term M is may-solvable if there is a sequence of bags $P_1, \ldots, P_n$ such that $MP_1 \ldots P_n$ reduces to a sum in which *at least one* term of the sum is the identity $\lambda x.x$, while M is must-solvable when the final sum is *non-empty* and *all* its terms are the identity.

In [PRDR10b] we characterize the may-solvability from an operational and semantical point of view. We extend the λ-calculus notion of head-normal form into a notion of outer-normal form.

[3]The original definition of λ-definable function is due to Church: a function $\phi : \mathbb{N} \to \mathbb{N}$ is λ-definable whenever there is a closed λ-term F such that for any Church numeral $\underline{n}$:

$$\begin{array}{ll} F\underline{n} \xrightarrow{*}_\beta \underline{m} & \text{if } \phi(n) = m, \\ F\underline{n} \text{ has no } \beta\text{-normal form} & \text{if } \phi(n) \text{ undefined.} \end{array}$$

This definition has however several disadvantages, especially it does not compose. The solution proposed by Barendregt was to replace the notion of "having no β-normal form" with that of "being unsolvable" [Bar71]. Independently, Wadsworth proposed the notion of "having no head-normal form" in [Wad71]. Barendregt's and Wadsworth's solutions have been proved to be equivalent in [Wad76].

Grammar of types ($\wedge$ commutative and associative, ω neutral element, $*$ a constant):

$$\text{term types:}\quad a,b ::= * \mid m \to a \qquad \text{bag types:}\quad m,p ::= \omega \mid a \mid m \wedge p$$

Inference rules ($\Gamma \wedge \Delta$ defined point-wise):

$$\frac{}{\vec{y}:\omega, x:a \vdash_{\mathtt{m}} x:a} \qquad \frac{\Gamma, x:m \vdash_{\mathtt{m}} M:a}{\Gamma \vdash_{\mathtt{m}} \lambda x.M : m \to a} \qquad \frac{\Gamma \vdash_{\mathtt{m}} M : m \to a \quad \Delta \vdash_{\mathtt{m}} P : m}{\Gamma \wedge \Delta \vdash_{\mathtt{m}} MP : a}$$

$$\frac{}{\vec{y}:\omega \vdash_{\mathtt{m}} 1:\omega} \qquad \frac{\Gamma \vdash_{\mathtt{m}} M:a}{\Gamma \vdash_{\mathtt{m}} [M]:a} \qquad \frac{\Gamma_1 \vdash_{\mathtt{m}} \mathfrak{M} : a_1 \ \ldots \ \Gamma_n \vdash_{\mathtt{m}} \mathfrak{M} : a_n, \quad \text{for } n \in \mathbb{N}}{\bigwedge_i \Gamma_i \vdash_{\mathtt{m}} [\mathfrak{M}^!] : a_1 \wedge \cdots \wedge a_n}$$

$$\frac{\Gamma \vdash_{\mathtt{m}} P:m \qquad \Delta \vdash_{\mathtt{m}} Q:p}{\Gamma \wedge \Delta \vdash_{\mathtt{m}} P \cdot Q : m \wedge p} \qquad \frac{\Gamma \vdash_{\mathtt{m}} M:a}{\Gamma \vdash_{\mathtt{m}} M + \mathfrak{N} : a}$$

Figure 2.3: intersection type system characterizing may-solvability in Λ^r.

Definition 13 ([PRDR10b]). A resource term is an outer-normal form if it has no redex but under the scope of a $(\ )^!$.

Theorem 14 ([PRDR10b]). *Given $M \in \Lambda^r$, the following are equivalent:*

1. *M is may-solvable,*
2. *M is reducible to a sum of terms containing at least one outer-normal form,*
3. *M is typable in the intersection type system of Figure 2.3.*

Notice that, by standardization (Theorem 12), condition 2 is equivalent to: there is an outer reduction of M to a sum of terms containing an outer-normal form.

The paper [PRDR10a] gives an extended account of may-solvability and also addresses the question of must-solvability. Unfortunately, the characterization of this last one is a tough problem because we have deadlocks (i.e. the empty sum 0) and diverging terms (like $\mathbf{\Omega}$). They are both considered unsolvable since they have a constant behavior with the environment, but they have a crucial difference: 0 is the neutral element of the sum, hence it disappears when added to a solvable term, $\mathbf{\Omega}$ does not. This difference is harmless when considering may-solvability but becomes relevant for the must-solvability. Recall the notation $\oplus$ of Equation 2.1, and define:

$$M_1 = \lambda x.(\mathbf{\Omega} \oplus x) \xrightarrow{*} \lambda x.\mathbf{\Omega} + \lambda x.x, \quad M_2 = \lambda x.(\mathbf{\Omega}[x] \oplus x) \xrightarrow{*} \lambda x.\mathbf{\Omega}[x] + \lambda x.x.$$

The term M_1 is must-solvable whilst M_2 is not. In fact, both $\lambda x.\mathbf{\Omega}$ and $\lambda x.\mathbf{\Omega}[x]$ are not may-solvable (by Theorem 14, they have no outer-normal

Grammar of types ($*$ and ω different constants):

$$v, u ::= * \mid v \to u \mid v \wedge u$$

Pre-order $\preceq$ (smallest pre-order satisfying the following rules):

$$v \preceq \omega \qquad v \preceq v \wedge v \qquad v_1 \wedge v_2 \preceq v_i \ (i = 1, 2) \qquad \omega \to u \preceq u \preceq \omega \to u$$

$$\omega \preceq \omega \to \omega \qquad v \to \omega \preceq \omega \to \omega \qquad (v \to u') \wedge (v \to u'') \preceq v \to (u' \wedge u'')$$

$$\frac{v \preceq v' \quad u \preceq u'}{v \wedge u \preceq v' \wedge u'} \qquad \frac{v' \preceq v \quad u \preceq u'}{v \to u \preceq v' \to u'}$$

Inference rules:

$$\frac{}{\Gamma, x : v \vdash_{\mathtt{M}} x : v} \qquad \frac{}{\Gamma \vdash_{\mathtt{M}} M : \omega} \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : v \quad \Gamma \vdash_{\mathtt{M}} \mathfrak{N} : v}{\Gamma \vdash_{\mathtt{M}} M + \mathfrak{N} : v}$$

$$\frac{\Gamma, x : v \vdash_{\mathtt{M}} M : u}{\Gamma \vdash_{\mathtt{M}} \lambda x.M : v \to u} \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : v \to \tau \quad \Gamma \vdash_{\mathtt{M}} P : v}{\Gamma \vdash_{\mathtt{M}} MP : \tau}$$

$$\frac{\Gamma \vdash_{\mathtt{M}} \mathfrak{M} : v}{\Gamma \vdash_{\mathtt{M}} [\mathfrak{M}^!] : v} \qquad \frac{\Gamma \vdash_{\mathtt{M}} \mathfrak{M} : v \quad \Gamma \vdash_{\mathtt{M}} P : v}{\Gamma \vdash_{\mathtt{M}} [\mathfrak{M}^!] \cdot P : v}$$

$$\frac{\Gamma \vdash_{\mathtt{M}} M : v \quad \Gamma \vdash_{\mathtt{M}} M : u}{\Gamma \vdash_{\mathtt{M}} M : v \wedge u} \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : u \quad u \preceq v}{\Gamma \vdash_{\mathtt{M}} M : v}$$

Figure 2.4: intersection type system characterizing must-solvability in $\Lambda^{\oplus}$.

form). Hence the only possibility for having a sum of identities from M_1 (resp. M_2) is to find an applicative context reducing $\lambda x.\mathbf{\Omega}$ (resp. $\lambda x.\mathbf{\Omega}[x]$) to 0 and keeping $\lambda x.x$ different from 0. Such a context exists for M_1, e.g. $(\_)[\mathbf{I}]$, but not for M_2, both $\lambda x.\mathbf{\Omega}[x]$ and $\lambda x.x$ using x linearly.

This example shows: first, that the implication *must-solvable* $\Rightarrow$ *sum of outer-normal forms* does not hold in Λ^r, M_1 being a contra-example; second, that in order to characterize whether a sum is must-solvable, one should know if there are contexts reducing the diverging terms of a sum to 0 and keeping at least one among the others different from 0.

Because of such kinds of problems (see [PRDR10a, §5] for further details), we restricted our study to the purely non-deterministic calculus $\Lambda^{\oplus}$. Notice that outer-normal forms of $\Lambda^{\oplus}$ coincide with head-normal forms.

Theorem 15 ([PRDR10a])**.** *Given $M \in \Lambda^{\oplus}$, the following are equivalent:*

1. *M is must-solvable,*

2. *M is reducible to a sum of outer-normal forms,*

3. M has a non-trivial[4] *type in the intersection type system of Figure 2.4.*

The system $\vdash_{\mathtt{M}}$ of Figure 2.4 is different from $\vdash_{\mathtt{m}}$ of Figure 2.3. Essentially, $\vdash_{\mathtt{m}}$ has a quantitative aspect missing in $\vdash_{\mathtt{M}}$. This is reflected in the fact that the intersection is idempotent in the latter and not in the former. Indeed, in $\vdash_{\mathtt{m}}$ the non-idempotency of the intersection is crucial to account for the consumption of linear resources, while in $\vdash_{\mathtt{M}}$ we do not need to count the number of resources in a bag since $\Lambda^{\oplus}$ allows only reusable resources. Moreover, in order to type a sum, all elements of the sum need to be typed by the same type, and this allows us to characterize must-solvability.

Indeed, the two systems provide logical descriptions of two models in different ambient categories. The system $\vdash_{\mathtt{m}}$ gives a model of Λ^{r} in the relational category MREL (see Appendix A and Figure A.2), while $\vdash_{\mathtt{M}}$ is describing the model $\mathcal{D}_{\infty}$ of $\Lambda^{\oplus}$ in the category of Scott domains. Moving from MREL to Scott domains is not necessary for characterizing must-solvability: we could give a non-idempotent intersection type system based on MREL, but we did not. This is a flaw in [PRDR10a], in fact the relationship between the two categories was not clear to me at the time I worked on that project. Ehrhard clarifies this relation in [Ehr12b].

The question of full-abstraction

The intersection type system of Figure 2.4 extends to resource calculus de Carvalho's system $\mathcal{R}$, the latter one providing a logical presentation to a relational model of the untyped λ-calculus ([dC07, dC09], see also Equation (1.5)). Indeed, if we suppose the equation $\omega \to * = *$, our type system describes the MREL model $\mathcal{M}_{\infty}$ of the resource calculus, studied in [Man12]. The denotation of a closed term is equal to the set of its types:

$$[\![M]\!]^{\mathcal{M}_{\infty}} = \{a \text{ s.t. } \vdash_{\mathtt{m}} M : a\}.$$

The choice of a notion of solvability induces an observational equivalence on terms (denoting by $C(\_)$ a context, i.e. a term with possible some holes $(\_)$, and by $C(M)$ the term obtained by replacing the holes with M, allowing for the capture of free variables in M):

$$M \sim N \text{ iff } \forall C(\_), C(M) \text{ may-solvable} \Leftrightarrow C(N) \text{ may-solvable.} \qquad (2.3)$$

An easy consequence of Theorem 14 is that the equality of denotations ($[\![M]\!] = [\![N]\!]$) implies the observational equality ($M \sim N$). If also the converse implication holds ($M \sim N$ implies $[\![M]\!] = [\![N]\!]$) the model is said to be <u>fully abstract</u>. In fact, Breuvart (Ph.D. student co-directed by Bucciarelli, assistant professor at Paris 7, and myself) showed that this is not

[4] I.e. not having a positive occurrence of ω.

the case: there are two observationally equivalent resource terms having different interpretation in $\mathcal{M}_\infty$ (see [Bre13]). Indeed, a variant of Breuvart's contra-example shows that $\mathcal{M}_\infty$ fails to be fully abstract also for the purely non-deterministic calculus $\Lambda^{\oplus}$.

Open problem 4 ($\star\star\star$)**.** *Is it possible to get a fully-abstract model of the observational equivalence over Λ^r induced by may-solvability? Also, there is no known fully-abstract model, at least in* MREL*, of the purely non-deterministic extension $\Lambda^{\oplus}$ of the λ-calculus.*

$$\begin{array}{lll} M \equiv_\eta \lambda x.M[x^!], & \text{for } x \notin FV(M) & (\eta\text{-equivalence}) \\ [(\mathfrak{M}+\mathfrak{N})^!] \equiv_\tau [\mathfrak{M}^!, \mathfrak{N}^!], & [\mathfrak{M}^!] \equiv_\tau 1 + [\mathfrak{M}, \mathfrak{M}^!] & (\text{Taylor-equivalence}) \end{array}$$

Figure 2.5: the equations inducing the maximal non-trivial congruence on normalizable terms extending the reduction of Λ^r.

2.4 Böhm theorem in resource calculus (2010)

In 2010, I studied Böhm theorem for the resource calculus, in collaboration with Manzonetto (at that time post-doc at Radbound University). The results have been published in [MP11].

Böhm theorem is a fundamental result in untyped λ-calculus achieved in [Böh68] and stating that, given two closed distinct $\beta\eta$-normal forms M and N, there exists a sequence of λ-terms $\vec{L}$, such that $M\vec{L}$ β-reduces to the first projection $\lambda xy.x$ and $N\vec{L}$ β-reduces to the second projection $\lambda xy.y$. As an important consequence we have that the $\beta\eta$-equivalence is the maximal non-trivial congruence on normalizable λ-terms extending the β-equivalence.

In resource calculus the η-equivalence does not suffice to capture the maximal extension of the equivalence $\equiv_r$ induced by the Λ^r reduction. One should consider also the equivalence $\equiv_\tau$ defined in Figure 2.5, which is reminiscent of the equality $e^{x+y} = e^x e^y$ and a sort of "scalar-forgetting" Taylor approximation of the exponential $e^x \sim 1 + xe^x$. As corollary to our separation theorem (Theorem 16), we get that $\equiv_{r\eta\tau}$ is the maximal non-trivial congruence on normalizable terms extending $\equiv_r$ [MP11, Corr. 1].

We consider the calculus with an idempotent sum, i.e. terms are supposed to be equated by the congruence generated by

$$M = M + M. \tag{2.4}$$

Basically, this amounts to say that we only check whether a term appears in a result, not how many times it appears.

Theorem 16 ([MP11])**.** *Let M, N be closed resource r-normalizable terms. If $M \not\equiv_{r\eta\tau} N$, then there is a sequence $\vec{P}$ of bags such that either $M\vec{P} \xrightarrow{*} \lambda x.x$ and $N\vec{P} \xrightarrow{*} 0$, or* vice versa.

This property is called semi-separation because of the asymmetry between the two values: the identity $\lambda x.x$, on the one hand, and 0, on the other hand. In fact, the empty sum represents a normal form (from the rewriting point of view) which is undefined (from the semantic point of view), and one cannot hope to separate a term less defined than another one without reducing the first to 0.

A crucial ingredient in the proof of Theorem 16 is the notion of Taylor expansion of a term. This notion has been defined by Ehrhard and Regnier in [ER03], as a translation of resource terms to formal series of multi-linear terms (i.e. resource terms without reusable resources) and non-negative rational coefficients. The definition is by structural induction on the term to be expanded, the crucial step[5] being the translation of a bag of a reusable resource:

$$Taylor([M^!]) \triangleq \sum_{n=0}^{\infty} \frac{1}{n!} [\overbrace{Taylor(M), \ldots, Taylor(M)}^{n \text{ times}}]. \tag{2.5}$$

The coefficient $\frac{1}{n!}$ is there to take care of the number of permutations over the n linear occurrences of M in the multiset. The terminology is motivated by the quantitative models of λ-calculus, where the n-th term of the series is related to the n-th derivative at 0 of the expanded term.

The proof of Theorem 16 proceeds as follows. Suppose M, N are normal forms (the case of M, N normalizable will follow trivially) and $M \not\equiv_{\eta\tau} N$. One can prove that there is a multi-linear term T occurring with non-zero coefficient in the Taylor expansion of only one between M and N. Say T occurs in the support of $Taylor(M)$. By looking at the structure of T, the proof builds a sequence of bags $\vec{P}$ behaving as a kind of equality test to T, i.e. such that $T\vec{P} \xrightarrow{*} \lambda x.x$ and for all other multi-linear terms T' appearing in the Taylor expansions of M and N, $T'\vec{P} \xrightarrow{*} 0$. Since operational semantics commutes with Taylor expansion (as essentially showed in [ER08, ER06a]) we conclude that $M\vec{P} \xrightarrow{*} \lambda x.x$ and $N\vec{P} \xrightarrow{*} 0$.

Namely, the construction of $\vec{P}$ depends also on the depth (maximal number of nested applications) of M and N, which bounds the depth of any multi-linear term in the supports of $Taylor(M)$ and $Taylor(N)$. Hence, the proof does not generalize to non-normalizable terms.

Open problem 5 ($\star\star$). *The separation theorem of λ-calculus has been extended to non-normalizable terms by Hyland [Hyl76]. Indeed, the $\beta\eta$-equivalence is not maximal on such terms, and one must consider the congruence $\mathcal{H}^\star$ equating two λ-terms whenever they have the same Böhm tree up to possibly infinite η-expansions [Bar84, §16.2]. Then one proves that $\mathcal{H}^\star$ is the maximal non-trivial congruence on λ-terms extending β-equivalence and equating all unsolvable terms [Bar84, Thm. 16.2.6].*

What about resource calculus? May we extend the separation theorem to non-normalizable resource terms? Can we characterize the may operational equivalence on resource terms as an equivalence between tree-like structures?

[5]In all the other cases the Taylor expansion commutes by multi-linearity, e.g. $Taylor(MP) \triangleq \sum_T \sum_Q t_T q_Q TQ$, whenever $Taylor(M) = \sum_T t_T T$ and $Taylor(Q) = \sum_Q q_Q Q$, for $t_T, q_Q \in \mathbb{Q}^+$.

2.5 Characterizing the Taylor expansion (since 2009)

The proof of Theorem 16 should convince the reader of the crucial role played by Taylor expansion in comparing resource terms. Something similar will happen when dealing with the separation of probabilistic PCF terms in Section 3.3. Roughly speaking, Taylor expansion is a quantitative refinement of Böhm trees crucial in the study of languages with "quantitative" primitives (e.g. finitely available resources, random functions, quantum bits, etc...), exactly as Böhm trees are at the core of the study of qualitative properties of regular functional languages (e.g. termination, solvability in λ-calculus, PCF, etc...).

The papers [PT09a, BHP13] study characterizations of the images of regular λ-terms and linear logic proof nets via Taylor expansion. The goal is to understand whether the property of being a "deterministic program" can be recovered in the space of the formal series of multi-linear terms. Let us detail here [BHP13], being the result of a M2 stage of Fanny He (now Ph.D. student at Bath University) co-directed by Boudes (assistant professor at Paris 13) and myself.

Let me recall that I mean with multi-linear term a resource term without reusable resources (i.e. without the ! constructor).

The reduction of resource calculus is strongly normalizing on multi-linear terms (any ers decreases the number of occurrences of variables in any addendum of the contractum) and confluent, hence the normal form NF(T) of a multi-linear term T is well-defined and can be extended to formal series:

$$\mathrm{NF}\left(\sum_T p_T T\right) \triangleq \sum_T p_T \mathrm{NF}(T)$$

Basically, NF$(\mathit{Taylor}(M))$ is a canonical writing of the Taylor expansion of M, sound with respect to the quantitative models interpreting resource calculus. The question addressed in [BHP13] is to characterize the series of normal multi-linear terms which are Taylor expansion of λ-terms.

Fortunately, the problem is significantly simplified by a result of Ehrhard and Regnier. Namely, they prove [ER08, Corollary 35] that the normal form of the Taylor expansion of a λ-term M can be defined as:

$$\mathrm{NF}(\mathit{Taylor}(M)) = \sum_{T \in \mathrm{Supp}(\mathrm{NF}(\mathit{Taylor}(M)))} \frac{1}{\mathrm{m}(T)} T \qquad (2.6)$$

where: m(T) is a natural number (called multiplicity coefficient) univocally defined by T, and Supp (NF$(\mathit{Taylor}(M))$) is the support of the normal form of the Taylor expansion of M, i.e. the set of the simple terms appearing with

$$\lambda x_1 \dots x_m . y P_1 \dots P_n \preceq T \text{ iff}$$
$$\begin{cases} T = \lambda x_1 \dots x_m . y Q_1 \dots Q_n \text{ and} \\ \forall i < n, \text{if } P_i \neq 1 \text{ then } \exists T' \in Q_i, \forall T'' \in P_i, T'' \preceq T'. \end{cases}$$

Figure 2.6: definedness relation $\preceq$ over normal multi-linear terms, used to characterized the Taylor expansion of a λ-term.

a non-zero coefficient in $\mathrm{NF}(\mathit{Taylor}(M))$. The open issue is then to characterize the sets of multi-linear terms which are supports of $\mathrm{NF}(\mathit{Taylor}(M))$, for some λ-term M.

Theorem 17 ([BHP13])**.** *Let $\mathfrak{T}$ be a set (possibly infinite) of multi-linear normal terms. There is a λ-term M such that the support of* $\mathrm{NF}(\mathit{Taylor}(M))$ *is $\mathfrak{T}$ iff the following conditions hold:*

1. $\mathrm{FV}(\mathfrak{T})$ *is finite,*
2. $\mathfrak{T}$ *is recursively enumerable,*
3. $\mathfrak{T}$ *is downward closed and directed wrt $\preceq$ of Figure 2.6 (i.e. $\forall T \preceq T' \in \mathfrak{T}, T \in \mathfrak{T}$ and $\forall T, T' \in \mathfrak{T}, \exists T'' \in \mathfrak{T}$, s.t. $T, T' \preceq T''$).*

The characterization is rather easy to achieve (indeed, it has been the subject of a M2 stage) because based on two previous results: a theorem by Ehrhard and Regnier stating that Taylor expansion and resource reduction commute [ER06a, Thm. 8], and Barendregt's characterization of the Böhm-like trees which are Böhm trees of λ-terms ([Bar84, Thm. 10.1.23]). Conditions (1) and (2) are in fact an adaptation of Barendregt's characterization, but they are not sufficient to characterize Taylor expansions since this latter one gives a more atomic decomposition of λ-terms than that obtained from Böhm trees. Condition (3) looks like the usual one characterizing the set of finite approximants of a tree as an ideal with respect to the subtree order relation. However, the fine grained notion of approximant given by the multi-linear terms makes the $\preceq$ relation a bit subtler than a subtree relation, in fact it is not even a preorder relation[6].

In [PT09a], written in collaboration with Tasson (Ph.D. student at Paris 7 at that time), I studied the characterization of the support of Taylor expansion in the setting of propositional linear logic proof nets. The approach was quite different than that of [BHP13]: we defined a rewriting algorithm taking as input a finite set of cut-free differential nets (corresponding in

[6]The relation $\preceq$ is transitive, but it is not reflexive because $y[x,z] \not\preceq y[x,z]$, nor anti-reflexive ($y[x] \preceq y[x]$). Also, it is neither symmetric ($y1 \preceq y[x]$ but $y[x] \not\preceq y1$), nor anti-symmetric ($y[x] \preceq y[x,x]$ and $y[x,x] \preceq y[x]$).

resource calculus to the normal multi-linear terms) and either returning a cut-free proof net or falling in a deadlock.

Open problem 6 ($\star\star\star$)**.** *These characterizations rely on Ehrhard and Regnier's Equation* (2.6)*, stating a strict correspondence between supports and coefficients of the Taylor expansion of a* λ*-term. Unfortunately, this correspondence fails as soon as one allows to superpose programs (e.g. by adding a choice constructor, or a random operator). A challenging issue is then to capture the convergence of formal combinations of multi-linear terms in a non-deterministic or probabilistic setting.*

Roughly speaking, this amounts to distinguish between a series expressing a kind of "finitary non-determinism" (something that can be generated by a non-deterministic program) from a completely arbitrary series of multi-linear terms. A starting point can be [Ehr10], where a notion of finitary support of Taylor series is defined, by importing constructions from finiteness spaces ([Ehr05] and Section 1.2). However, finiteness spaces mix the question of finitary non-determinism with that of termination, hence they are unsuitable for studying Turing complete languages like $\Lambda^{\oplus}$ *or probabilistic* PCF*. In fact, [Ehr10] focuses on a non-deterministic extension of System F [Gir72].*

Is it possible to have a notion of convergence of a series to a non-deterministic program, independently whether its evaluation terminates or not? Is it possible to characterize the image of $\Lambda^{\oplus}$ *or of probabilistic* PCF *in the space of the formal series of multi-linear terms?*

Chapter 3

Quantitative semantics

- T. Ehrhard, M. Pagani, C. Tasson. *Probabilistic Coherence Spaces are Fully Abstract for Probabilistic PCF.* 41st ACM SIGACT – SIGPLAN Symposium on Principles of Programming Languages (POPL 2014), Sewell ed., to appear in 2014.

- M. Pagani, P. Selinger, B. Valiron. *Applying Quantitative Semantics to Higher-Order Quantum Computing.* 41st ACM SIGACT – SIGPLAN Symposium on Principles of Programming Languages (POPL 2014), Sewell ed., to appear in 2014.

- J. Laird, G. Manzonetto, G. McCusker, M. Pagani. *Weighted Relational Models of Typed Lambda-Calculi.* 28th Annual IEEE Symposium on Logic in Computer Science (LICS 2013), Kupferman ed., IEEE, 2013.

- T. Ehrhard, M. Pagani, C. Tasson. *The Computational Meaning of Probabilistic Coherence Spaces.* 26th Annual IEEE Symposium on Logic in Computer Science (LICS 2011), Grohe ed., IEEE, 2011.

3.1 Preliminaries

Take a program M of type $\mathtt{1} \to \mathtt{1}$, where $\mathtt{1}$ is the unit type whose only value is `skip`. The basic idea of quantitative semantics is to interpret M as a power series (centered at 0):

$$[\![M]\!] = \sum_{n=0}^{\infty} p_n x^n .$$

The unknown x corresponds to the input of M and the exponent n refers to the number of times M will call x in order to give an output. The whole series gathers all the possible number of calls for x. The coefficient p_n is the part specific to the program M, giving a weight to the possibility that M actually uses the input n times. Indeed, if M is a deterministic program (e.g. a regular λ-term), then p_n will be zero everywhere except for at most one monomial. The degree of this monomial tells us how many times M needs to use the input value `skip` in order to give the output `skip`. However, this simple situation changes as we move to more complex data types or languages.

The series has only one unknown because the input type has dimension one. The dimension of a ground type is the number of its values. Take then the boolean type `Bool` with two values `tt` and `ff`, and consider now M of type $\mathtt{Bool} \to \mathtt{1}$. Its denotation is a power series with two unknowns:

$$[\![M]\!] = \sum_{n=0}^{\infty} \sum_{h=0}^{\infty} p_{n,h} x_{\mathtt{tt}}^{n} x_{\mathtt{ff}}^{h} ,$$

where a single monomial $x_{\mathtt{tt}}^{n} x_{\mathtt{ff}}^{h}$ expresses the computations calling the input $n+h$ times, this input behaving n times as `tt` and h times as `ff`. Such semantics have a built-in form of non-determinism, allowing for a non-uniform behavior of the various calls for the input.

Finite multisets give a convenient notation for multivariable series:

$$[\![M]\!] = \sum_{m \in \mathcal{M}_{\mathrm{f}}(\{\mathtt{tt},\mathtt{ff}\})} p_m x_{\mathtt{tt}}^{m(\mathtt{tt})} x_{\mathtt{ff}}^{m(\mathtt{ff})} = \sum_{m \in \mathcal{M}_{\mathrm{f}}(\{\mathtt{tt},\mathtt{ff}\})} p_m x^m ,$$

where $\mathcal{M}_{\mathrm{f}}(\{\mathtt{tt},\mathtt{ff}\})$ is the set of the finite multisets over $\{\mathtt{tt},\mathtt{ff}\}$, and $m(\mathtt{tt})$, $m(\mathtt{ff})$ denote the number of occurrences in m of, respectively, `tt` and `ff`. Also, the x in the last series is a variable of dimension two, representing the two original unknowns $x_{\mathtt{tt}}$ and $x_{\mathtt{ff}}$.

If we consider output types with more than one possible value, then the denotation describes a family of power series and not a sole one. If M has type $\mathtt{Bool} \to \mathtt{Bool}$, we have two multivariable power series:

$$[\![M]\!]_{\mathtt{tt}} = \sum_{m \in \mathcal{M}_{\mathrm{f}}(\{\mathtt{tt},\mathtt{ff}\})} p_{m,\mathtt{tt}} x^m, \quad \text{and} \quad [\![M]\!]_{\mathtt{ff}} = \sum_{m \in \mathcal{M}_{\mathrm{f}}(\{\mathtt{tt},\mathtt{ff}\})} p_{m,\mathtt{ff}} x^m,$$

with two different families of coefficients. Indeed, once fixed input and output types, the only information needed to express such power series are the monomial coefficients. Hence, the denotation can be presented as a matrix, whose lines are indexed by finite multisets (describing monomials), whose columns are indexed by the possible output values (describing a specific series) and whose entries are the coefficients:

$$[\![M]\!] = \begin{array}{r c} & \begin{array}{cc} \overset{\texttt{tt}}{\downarrow} & \overset{\texttt{ff}}{\downarrow} \end{array} \\ \begin{array}{r} [\,]\rightarrow \\ [\texttt{tt}]\rightarrow \\ [\texttt{ff}]\rightarrow \\ [\texttt{tt},\texttt{ff}]\rightarrow \\ \vdots \end{array} & \begin{pmatrix} p & q \\ p' & q' \\ p'' & q'' \\ p''' & q''' \\ \vdots & \vdots \end{pmatrix} \end{array} \tag{3.1}$$

For example, the coefficient $[\![M]\!]_{[\texttt{tt},\texttt{tt},\texttt{ff}],\texttt{ff}}$ is the coefficient of the monomial $x^2_{\texttt{tt}}x_{\texttt{ff}}$ in the series describing the computations of M returning $\texttt{ff}$.

These intuitions are at the core of various semantics of linear logic and λ-calculus, that I refer to with the generic name of *quantitative semantics*. The first quantitative model is Girard's normal functors semantics [Gir88]. More recently, quantitative semantics provided solid, denotational models for various algebraic extensions of λ-calculus, such as probabilistic, differential and quantum λ-calculi (e.g. [DE11, EPT14], [Ehr05], [PSV14]).

Linear logic proofs are represented by linear functions (i.e. families of power series of degree one). This is the ideal setting to enlighten one among the most astonishing features of linear logic: the correspondence between the cut-elimination behavior of logical rules and the standard constructions of linear algebra, alluded by the linear logic jargon (tensor $\otimes$, direct sum $\oplus$, dual space, etc...). For example, the splitting of the classical sequent rules into multiplicatives and additives has been motivated in Section 1.1 with respect to two different behaviors under cut-elimination: branching and inter-communication. In the quantitative models, this splitting corresponds to two different ways of aggregating linear functions: direct products and tensors, both defined by universal properties.

There are in the literature a multitude of quantitative semantics. I list here some choices which must be taken in order to implement a concrete model. The list is not at all exhaustive, but I hope that it will give an idea of the different features of these models.

Scalars. In Girard's normal functors model [Gir88], scalars are possibly infinite sets. Ehrhard's Köthe sequences spaces [Ehr02] and finiteness spaces [Ehr05] recast Girard's intuitions on actual vector spaces, taking scalars from standard fields: Köthe sequences spaces consider the field of real numbers $\mathbb{R}$

as well as that of complex numbers $\mathbb{C}$, whilst the finiteness spaces construction work over any field[1]. In the weighted relational models [LMMP13] that I will detail in Section 3.2, scalars can be taken from any continuous commutative semi-ring. In the model developed by Selinger, Valiron and myself ([PSV14] and Section 3.4) the coefficients are completely positive maps of finite dimension.

Convergence. To choose scalars we must also consider a notion of convergence of a series of scalars. This is a crucial issue of the model, necessary to compose power series (hence to have a category) and to see them as well-defined functions from inputs to outputs. A shortcut is by postulating that every series converges everywhere. This can be done by endowing scalars with a complete order, having the bottom element 0 (the neutral element of the sum) and the top element ∞ (which is absorbing with respect to the sum), and then postulating that the "morally" diverging series will value ∞. This choice is taken in the models in [LMMP13] and [PSV14]. The price (not so high) to pay is that scalars must be always non-negative and so data types are modules rather than vector spaces.

Ehrhard's models [Ehr02] and [Ehr05] adopt less obvious topologies (in particular, Köthe sequences spaces use the standard topology of real and complex numbers), allowing for divergent series, but restricting the hom-sets to *continuous* power series, which always compose and converge absolutely in the vector space associated with the input type. The other side of the coin is that this restriction makes the hom-sets not cpo-enriched, hence failing to model fix-point combinators and the untyped λ-calculus.

Powers. Girard's translation of the functional type of programs $A \to B$ into the linear logic formula $!A \multimap B$ is the bridge between linear functions and power series. In particular, the promotion $!A$ is the operation lifting a vector x of type A into the space $!A$ of "powers" of type A. The question is: what is a "power" of x? Here is another point where quantitative models may differ considerably.

In [LMMP13] and [PSV14], the exponential $!A$ is defined as the infinite bi-product of the symmetric n-fold tensor powers of A, following the formula:

$$!A = \bigoplus_{n=0}^{\infty} A^n, \tag{3.2}$$

the intuition being that the n-th layer A^n of such a bi-product contains the monomials of degree n. The fact that we are considering symmetric tensors amounts to say that the order in which unknowns appear is irrelevant.

[1]We have presented the category FIN of finiteness spaces in Section 1.2.1 as a refinement of the relational model of LL. The point is that FIN induces a category of linearly topological vector spaces on an arbitrary field $\mathcal{K}$, see [Ehr05].

Equation (3.2) does not work in Ehrhard's models, namely infinite products and infinite co-products are different in finiteness and Köthe sequence spaces. However, Melliès et al. showed that the exponentials of finiteness spaces can be obtained via a slightly different formula [MTT09]. In all models mentioned so far, the exponential comonoid is the free commutative comonoid (see Appendix A), however the freeness is not necessary to model linear logic exponentials. For sure, we are ignoring many other notions of exponentials in these semantics, which might have an interest for modeling operational properties. Blute et al. proposed to consider, for example, the exponentials generated by the antisymmetric tensor [BPS94].

3.2 Weighted relational semantics (since 2011)

Relational semantics is a simple example of quantitative semantics. Indeed, the category REL of sets and relations (Figure A.1) can be seen as a category of modules and linear functions over the boolean semi-ring $\mathcal{B} = (\{\mathbf{1}, \mathbf{0}\}, \mathbf{0}, \vee, \mathbf{1}, \wedge)$: any subset of a set A can be seen as a vector in the module $\mathcal{B}^A$, and any relation $f \subseteq A \times B$ can be seen as a matrix in $\mathcal{B}^{A\times B}$, describing a linear function from $\mathcal{B}^A$ to $\mathcal{B}^B$, defined by using a possibly infinite $\bigvee$:

$$f_{a,b} \triangleq \begin{cases} \mathbf{1} & \text{if } (a,b) \in f, \\ \mathbf{0} & \text{otherwise.} \end{cases} \qquad\qquad f(x)_b \triangleq \bigvee_{a\in A} f_{a,b} \wedge x_a$$

Relational composition (Equation (1.3)) is the usual matrix multiplication:

$$(f\,;\,g)_{a,c} \triangleq \bigvee_{b\in B} f_{a,b} \wedge g_{b,c}\ . \tag{3.3}$$

This analogy extends to all constructions of propositional linear logic (tensors, products, exponentials,...). One is then tempted to generalize it to scalar structures richer than $\mathcal{B}$.

The goal of the extended abstract [LMMP13], written in collaboration with Laird and McCusker (resp. lecturer and professor at Bath University), and Manzonetto (assistant professor at Paris 13), is double. First, it introduces a large class of models (weighted relational semantics) obtained by replacing the boolean ring with any *continuous commutative semi-ring* $\mathcal{R}$ (Definition 18). Second, it gives examples of how, by varying the choice of $\mathcal{R}$, one can use these models to capture various quantitative properties of program executions, e.g. may-convergence; probability of convergence; and minimum and maximum number of reduction steps needed to converge (Table 3.1).

Definition 18. A continuous commutative semi-ring $\mathcal{R}$ is a commutative semi-ring $(|\mathcal{R}|, +, \cdot, \mathbf{0}, \mathbf{1})$ equipped with a partial order $\preceq$ such that: (i) $(|\mathcal{R}|, \preceq)$ is a cpo having $\mathbf{0}$ as bottom element; (ii) the operators $+$ and $\cdot$ are continuous, that is for every directed set $D \subseteq \mathcal{R}$ and $r \in \mathcal{R}$ we have $\sup(r + D) = r + \sup D$, $\sup(r \cdot D) = r \cdot \sup D$.

Since addition is monotonic, one can define infinite sums by directed suprema:

$$\sum_{p\in I} p \triangleq \sup_{F\subseteq_{\text{fin}} I} \left(\sum_{p\in F} p\right).$$

Henceforth, $\mathcal{R}$ will refer to a continuous commutative semi-ring. For any choice of $\mathcal{R}$ we define a category $\mathcal{R}^{\oplus}$.

Definition 19 (The category $\mathcal{R}^{\oplus}$, [LMMP13])**.** The objects are sets A, B, $C\dots$. A morphism from A to B is a matrix in $\mathcal{R}^{A\times B}$, i.e. a linear function from $\mathcal{R}^A$ to $\mathcal{R}^B$. The composition is matrix multiplication:

$$(f\,;\,g)_{a,c} \triangleq \sum_{b\in B} f_{a,b}\cdot g_{b,c} \tag{3.4}$$

The identity is the diagonal matrix.

Notice that Equation (3.4) generalizes the relational composition as expressed in Equation (3.3). The sum might be infinite, but it is always converging because $(\mathcal{R},\preceq)$ is a cpo.

The notation $\mathcal{R}^{\oplus}$ refers to the fact that the category is the free infinite bi-product completion of $\mathcal{R}$, the construction being a variant of the free finite bi-product completion of an *Ab*-category given in [Mac71, §VIII.2 Exercise 6][2].

Theorem 20 ([LMMP13])**.** *The category $\mathcal{R}^{\oplus}$ is a cpo-enriched, $\star$-autonomous (in fact compact closed), Lafont category (see Appendix A and Figure 3.1). In particular, $\mathcal{R}^{\oplus}$ yields a model of propositional linear logic.*

Once we have a model of linear logic, we can use it to construct semantics for various languages and evaluation strategies. The main goal of [LMMP13] is to show some examples of how the algebraic structure induced by $\mathcal{R}$ permits to describe quantitative properties of program evaluations. We focused our attention on the language PCF$^{\texttt{Coin}}$, the extension of Plotkin's PCF [Plo77] with a nondeterministic choice operator. Similar results can be obtained for other languages, like untyped λ-calculus, $\lambda\mu$-calculus, Gödel's System T, etc.

Open problem 7 ($\star\star$)**.** *It is more challenging to extend the quantitative approach to polymorphic types, e.g. Girard's system F. Indeed, as far as I know, there is no published paper on a quantitative semantics of second order typing. Girard mentioned its possibility in various places (e.g. [Gir88] and [Gir86]), and Ehrhard developed a second order version of finiteness spaces in a manuscript. May we lift $\mathcal{R}^{\oplus}$ to model system F? May we achieve in a polymorphic setting the same kind of quantitative observations as in [LMMP13] (Table 3.1)?*

The definition of PCF$^{\texttt{Coin}}$ is sketched in Figure 3.2. The operational semantics implements a call-by-name evaluation strategy. In the case of a `Coin` redex we have two possible contracta: `tt` or `ff`.[3]

[2]Indeed, $\mathcal{R}$ is an *Ab*-category with a single object, scalars as morphisms and multiplication as composition.

[3]Notice the difference with respect to the rewriting systems presented in the previous two chapters: we do not gather the two contracta in a unique sum as in differential nets and resource calculus. This is just a minor variant in the presentation of the calculus, due to the fact that here we are interested in understanding the cost of computing a *single* result rather than in proving the rewriting properties of an operational semantics.

monoidal unit, dualizing object	$1 \triangleq \bot \triangleq \{*\}$
tensor on objects, hom-object	$A \otimes B \triangleq A \multimap B \triangleq A \times B$
tensor on maps $A \otimes B \mapsto C \otimes D$	$(f \otimes g)_{(a,b),(c,d)} \triangleq f_{a,c} \cdot g_{b,d}$
evaluation $(A \multimap B) \otimes A \mapsto B$	$\mathrm{eval}_{((a,b),a'),b'} \triangleq \delta_{(a,b),(a',b')}$

(a) monoidal structure

indexed biproduct	$\bigoplus_{i \in I} A_i \triangleq \bigcup_{i \in I} \{i\} \times A_i$
j-th projection $\bigoplus_{i \in I} A_i \mapsto A_j$	$\mathrm{proj}^j_{(i,a),a'} \triangleq \delta_{i,j} \cdot \delta_{a,a'}$
j-th injection $A_j \mapsto \bigoplus_{i \in I} A_i$	$\mathrm{inj}^j_{a',(i,a)} \triangleq \delta_{i,j} \cdot \delta_{a,a'}$

(b) cartesian structure

object	$!A \triangleq \mathcal{M}_{\mathrm{f}}(A)$
comultiplication $!A \mapsto !A \otimes !A$	$\mathrm{contr}_{m,(m_1,m_2)} \triangleq \delta_{m,m_1+m_2}$
counit $!A \mapsto 1$	$\mathrm{weak}_{m,*} \triangleq \delta_{m,[\,]}$
dereliction $!A \mapsto A$	$\mathrm{der}_{m,a} \triangleq \delta_{m,[a]}$

(c) free commutative comonoid

Figure 3.1: the $\star$-autonomous Lafont structure of $\mathcal{R}^{\oplus}$. The writing $\delta_{a,a'}$ refers to the Kronecker symbol, which takes value $\mathbf{1} \in \mathcal{R}$ if $a = a'$, and $\mathbf{0} \in \mathcal{R}$ if $a \neq a'$.

$$\begin{array}{rlr}
M, N, L ::= & x \mid \lambda x^A.M \mid MN & \text{simply typed } \lambda, \\
& \mid \mathtt{Y}M & \text{with recursion,} \\
& \mid \mathtt{tt} \mid \mathtt{ff} \mid \mathtt{if}(M,N,L) & \text{booleans,} \\
& \mid \underline{n} \mid \mathtt{pred}\, M \mid \mathtt{succ}\, M \mid \mathtt{zero?}\, M & \text{arithmetics,} \\
& \mid \mathtt{Coin} & \text{non-determinism.}
\end{array}$$

(a) Grammar of terms of PCF$^{\mathtt{Coin}}$.

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

$$\frac{}{\Gamma \vdash \mathtt{tt} : \mathtt{Bool}} \qquad \frac{}{\Gamma \vdash \mathtt{ff} : \mathtt{Bool}} \qquad \frac{\Gamma \vdash M : \mathtt{Bool} \quad \Gamma \vdash P : A \quad \Gamma \vdash L : A}{\Gamma \vdash \mathtt{if}(M,P,L) : A}$$

$$\frac{}{\Gamma \vdash \underline{n} : \mathtt{Int}} \qquad \frac{\Gamma \vdash M : \mathtt{Int}}{\Gamma \vdash \mathtt{pred}\ M : \mathtt{Int}} \qquad \frac{\Gamma \vdash M : \mathtt{Int}}{\Gamma \vdash \mathtt{succ}\ M : \mathtt{Int}} \qquad \frac{\Gamma \vdash M : \mathtt{Int}}{\Gamma \vdash \mathtt{zero?}\ M : \mathtt{Bool}}$$

$$\frac{\Gamma \vdash M : A \to A}{\Gamma \vdash \mathtt{Y}M : A} \qquad \frac{}{\Gamma \vdash \mathtt{Coin} : \mathtt{Bool}}$$

(b) The simple type assignment system.

$$(\lambda x.M)N \to M\{N/x\} \qquad \mathtt{Y}M \to M(\mathtt{Y}M) \qquad \mathtt{Coin} \to \mathtt{tt} \qquad \mathtt{Coin} \to \mathtt{ff}$$

$$\mathtt{if}(\mathtt{tt}, M, N) \to M \qquad \mathtt{if}(\mathtt{ff}, M, N) \to N$$

$$\mathtt{zero?}(\underline{0}) \to \mathtt{tt} \quad \mathtt{zero?}(\underline{n{+}1}) \to \mathtt{ff} \quad \mathtt{pred}(\underline{n{+}1}) \to \underline{n} \quad \mathtt{succ}(\underline{n}) \to \underline{n{+}1}$$

(c) Elementary reduction steps (ers).

$$MN \to M'N \qquad \mathtt{if}(M,N,L) \to \mathtt{if}(M',N,L)$$

$$\mathtt{pred}\ M \to \mathtt{pred}\ M' \qquad \mathtt{succ}\ M \to \mathtt{succ}\ M' \qquad \mathtt{zero?}\ M \to \mathtt{zero?}\ M'$$

(d) Contextual rules, supposing $M \to M'$.

Figure 3.2: syntax and operational semantics of PCF$^{\mathtt{Coin}}$.

continuous commutative semi-ring $\mathcal{R}$	interpretation of constructors	$[\![M]\!]^{\mathcal{R}}_V$
booleans $(\{\mathbf{1},\mathbf{0}\},\vee,\wedge,\mathbf{0}\leq\mathbf{1})$	standard	may-convergence to V
naturals $(\mathbb{N}\cup\{\infty\},+,\times,\leq)$	standard	number of evaluations to V
non-negative reals $(\mathbb{R}^+\cup\{\infty\},+,\times,\leq)$	$[\![\texttt{Coin}]\!]=(\frac{1}{2},\frac{1}{2})$	probability of reducing to V
tropical $(\mathbb{N}\cup\{\infty\},\min,+,\geq)$	$[\![\lambda x.M]\!]=\mathit{Curry}(1+[\![M]\!])$	minimum number of β to V
arctic $(\mathbb{N}\cup\{+\infty,-\infty\},\max,+,\leq)$	$[\![\lambda x.M]\!]=\mathit{Curry}(1+[\![M]\!])$	maximum number of β to V

Table 3.1: examples of $\mathcal{R}$ and of operational properties described by $\mathcal{R}^{\oplus}_{!}$.

This language has a standard interpretation in the category $\mathcal{R}^{\oplus}_{!}$, which is the coKleisli category associated with the exponential comonad $(!, \mathit{der}, \mathit{digg})$ of $\mathcal{R}^{\oplus}$, a.k.a. the category induced by Girard's translation $A \to B = {!A} \multimap B$.

Definition 21 (CoKleisli category $\mathcal{R}^{\oplus}_{!}$, [LMMP13])**.** The objects are sets A, B, $C\ldots$, the morphisms from A to B are matrices in $\mathcal{R}^{\mathcal{M}_{\mathrm{f}}(A)\times B}$, i.e. formal power series with coefficients in $\mathcal{R}$ and unknowns in A (recall the discussion in Section 3.1). The composition is the usual composition of power series, i.e. for $f : A \to B$, $g : B \to C$ and $m \in \mathcal{M}_{\mathrm{f}}(A)$, $c \in C$:

$$(f\,;\,g)_{m,c} \triangleq \sum_{[b_1,\ldots,b_n]\in\mathcal{M}_{\mathrm{f}}(B)} \ \sum_{\substack{(m_1,\ldots,m_n)\text{ s.t.}\\ m=m_1\uplus\cdots\uplus m_n}} g_{[b_1,\ldots,b_n],c}\cdot\prod_{i=1}^{n} f_{m_i,b_i}. \tag{3.5}$$

Ground types (like $\texttt{1}$, $\texttt{Bool}$, $\texttt{Int}$) are denoted by the set of their values. The functional type $A \to B$ is denoted by $\mathcal{M}_{\mathrm{f}}(A) \times B$, i.e. the index-set of the corresponding matrices. Terms are interpreted in the standard way, following the cpo-enriched cartesian closed structure of $\mathcal{R}^{\oplus}_{!}$, with the coin operator naturally denoted by addition of the $\texttt{tt}$ and $\texttt{ff}$ vectors:

$$[\![\texttt{Coin}]\!] \triangleq (\mathbf{1},\mathbf{0})+(\mathbf{0},\mathbf{1})=(\mathbf{1},\mathbf{1}).$$

A closed term of type A is interpreted by a vector in $\mathcal{R}^{[\![A]\!]}$. So, a closed term of ground type is a $\mathcal{R}$-weighted distribution of ground values. Weights represent an "abstract cost" of reducing a term M to a value V, the game consisting in finding suitable instances of $\mathcal{R}$ and suitable interpretations of the constructors giving a concrete operational meaning to this cost. In the table of Figure 3.1 I sum up the main examples described in [LMMP13, §7].

The first column of the table gives examples of continuous commutative semi-rings: the booleans (giving rise to relational semantics); the standard semi-ring of natural numbers completed with the top ∞; the positive cone of real numbers completed; the tropical semi-ring, where addition is min and multiplication is the usual sum and where the order is reversed (so that ∞ is the bottom and 0 is the top); and the arctic semi-ring which is someway dual to the tropical semi-ring.

The second column sketches the variant to the standard interpretations of some constructors (if any) in order to characterize the operational property given in the last column. For example, the last two lines require to interpret an abstraction $\lambda x.M$ by currying not $[\![M]\!]$, as it would be standard, but the matrix $1 + [\![M]\!]$. In this way, the scalar 1 is used as a kind of *slot*, counting the number of times a β-redex is reduced.

The last column specifies the computational meaning associated with a ground value V by the denotation $[\![M]\!]$. In the first line we have a boolean value corresponding to whether M may-converges to V or not. By taking the standard semi-ring on $\mathbb{N} \cup \{\infty\}$, we already get a much finer observation on programs: $[\![M]\!]_V$ counts the number of possible evaluations from M to V. In the third line, we have the probability that M reduces to V, supposing that the probability of choosing `tt` or `ff` when firing a `Coin` redex is uniformly distributed. Finally, the last two lines are an example of resource analysis, giving the min/max number of β-step in an evaluation of M to V.

The proof of Table 3.1 is interesting because parametric in the choice of the semi-ring. The crucial tool is a metalanguage PCF$^{\mathcal{R}}$ for quantitative modeling of the execution of programs in PCF$^{\texttt{Coin}}$. This language extends PCF$^{\texttt{Coin}}$ by allowing a scalar annotation:

$$pM \text{ for any } p \in \mathcal{R}.$$

The new constructor is naturally interpreted in $\mathcal{R}^{\oplus}_{!}$ by scalar multiplication $[\![pM]\!] \stackrel{\Delta}{=} p\,[\![M]\!]$. The operational semantics is extended by adding the ers:

$$pM \xrightarrow{\mathtt{p}} M$$

In fact, each ers is now weighted by $\mathbf{1}$ (the multiplicative unit of $\mathcal{R}$) except for the scalar ers, where the weight is the reduced scalar. So, we define the weight of a single reduction sequence as the product of the weights of its ers, and the weight of reducing a term M to a value V as the (possibly infinite) sum of the weights of all finite reduction sequences from M to V:

$$M \Downarrow^{p} V \quad \text{iff} \quad p = \sum_{M \xrightarrow{\mathtt{p_1}} \cdots \xrightarrow{\mathtt{p_n}} V} \prod_{i=1}^{n} p_i\,. \tag{3.6}$$

By adapting Plotkin's proof of PCF adequacy for Scott's domains [Plo77][4], we link the denotation of a term with its execution in $\text{PCF}^{\mathcal{R}}$.

Theorem 22 ([LMMP13]). *Given a closed term $M \in \text{PCF}^{\mathcal{R}}$ of ground type, for any closed value V of that type, we have:*

$$[\![M]\!]_V = p \textit{ iff } M \Downarrow^p V.$$

Table 3.1 is then achieved by easy corollaries describing the operational meaning of $M \Downarrow^p V$ in the specific instances of $\mathcal{R}$ taken into account.

The question of full-abstraction

Notice that the definition of $M \Downarrow^p V$ induces a $\mathcal{R}$-parametric observational equivalence on programs of $\text{PCF}^{\mathcal{R}}$:

$$M \sim_{\mathcal{R}} N \text{ iff } \forall C[\_], \forall p, C[M] \Downarrow^p \texttt{tt} \Leftrightarrow C[N] \Downarrow^p \texttt{tt} \tag{3.7}$$

Taking $\mathcal{R}$ to be the boolean semi-ring, this equivalence corresponds to the one defined in Equation (2.3) in an untyped setting.

The standard interpretation of $\text{PCF}^{\mathcal{R}}$ into $\mathcal{R}^{\oplus}_{!}$ is not fully abstract wrt this equivalence, for any choice of $\mathcal{R}$. This is due to the possibility of constructing in the language absorbing elements for the sum. In fact, let ∞ be the top element of $\mathcal{R}$ (which always exists as the sum of all scalars), and consider the term $\infty\texttt{tt}$.[5] Of course, $\infty\texttt{tt} \Downarrow^{\infty} \texttt{tt}$. We can then define (writing by $\mathbf{\Omega}$ the usual looping term, in this language $\mathbf{Y}(\lambda x.x)$):

$$M_1 \triangleq \lambda y.\infty\texttt{tt}, \qquad M_2 \triangleq \lambda y.\texttt{if}(\texttt{Coin}, \infty\texttt{tt}, \texttt{if}(y, \texttt{tt}, \mathbf{\Omega})). \tag{3.9}$$

For every argument N, we have $M_1 N \Downarrow^{\infty} \texttt{tt}$ and also, since $\infty = p + \infty$, $M_2 N \Downarrow^{\infty} \texttt{tt}$. By a context lemma (i.e. two terms are observationally equal iff they give the same ground values with the same weight on any argument), one then concludes that M_1 and M_2 are operationally indistinguishable.

However, the semantics induced by any non-trivial semi-ring (i.e. $\mathbf{1} \neq \mathbf{0}$), separates the two terms:

$$[\![M_1]\!]_{m,b} = \begin{cases} \infty & \text{if } m = [\,],\ b = \texttt{tt}, \\ \mathbf{0} & \text{otherwise.} \end{cases}, \quad [\![M_2]\!]_{m,b} = \begin{cases} \infty & \text{if } m = [\,],\ b = \texttt{tt}, \\ \mathbf{1} & \text{if } m = [\texttt{tt}],\ b = \texttt{tt}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

[4]Danos and Ehrhard have already given a quantitative version of Plotkin's adequacy in the setting of probabilistic coherence spaces ([DE11] and next Section 3.3). Our proof is a simple variant of theirs.

[5]Notice that, whenever $\infty = \sum_n \underbrace{\mathbf{1} + \cdots + \mathbf{1}}_{n \text{ times}}$ the term $\infty\texttt{tt}$ can be simulated without using the scalar multiplication:

$$\infty\texttt{tt} \triangleq \mathbf{Y}(\lambda x.\texttt{if}(\texttt{Coin}, \texttt{tt}, x)) \xrightarrow{\mathbf{1}}\xrightarrow{\mathbf{1}} \texttt{tt} \text{ or } \infty\texttt{tt}. \tag{3.8}$$

This phenomenon can be equivalently stated in more categorical terms, saying that the cartesian closed category $\mathcal{R}_!^{\oplus}$ is not well-pointed, i.e. there are morphisms (namely, matrices like $[\![M_1]\!]$ and $[\![M_2]\!]$) which are different but describe the same function over the points of $[\![\texttt{Bool}]\!]$ (i.e. over the vectors in $\mathcal{R}^{\{\texttt{tt},\texttt{ff}\}}$).

Open problem 8 ($\star\star\star$)**.** *From a non well-pointed model, one can build a well-pointed one by an extensional collapse of the morphisms in the starting category. Is it possible to give a concrete description of this collapse? Does it describe a scalar-based version of Scott domains, where derivations/linear resources are forbidden? Is it able to make the same observations of $\mathcal{R}_!^{\oplus}$ described in Table 3.1? Of course, the problem of full-abstraction for* $\textsc{pcf}^{\mathcal{R}}$ *is related to the open problem 4 of finding a fully abstract model for $\Lambda^{\oplus}$.*

3.3 Semantics of probabilistic primitives (since 2009)

The question of full-abstraction changes as one moves from a purely non-deterministic superposition of two data to a barycentric sum:

$$\frac{1}{2}\texttt{Coin} \xrightarrow{\frac{1}{2}} \texttt{tt}, \qquad \frac{1}{2}\texttt{Coin} \xrightarrow{\frac{1}{2}} \texttt{ff}.$$

Henceforth, $\mathfrak{R}$ will denote the continuous semi-ring $(\mathbb{R}^+ \cup \{\infty\}, +, \times, \leq)$, i.e. the order completion of the positive cone of real numbers.

The language of probabilistic PCF, PPCF in short, can be defined as the restriction of $\text{PCF}^{\mathfrak{R}}$ to the grammar of terms generated by the usual constructs of PCF plus $\frac{1}{2}\texttt{Coin}$. The crucial property of this language is that the weight p in $M \Downarrow^p V$ (Equation (3.6)), for a value V, is always within the interval $[0, 1]$, defining the probability that the term M evaluates to V.

In the extended abstract [EPT14], written in collaboration with Ehrhard and Tasson (both at Paris 7), we prove that probabilistic coherence spaces yield a fully abstract model of PPCF (Theorem 25). This model has been introduced in [DE11] and can be seen as a refinement of the $\mathfrak{R}$-weighted relational semantics, obtained by shrinking the hom-sets in a way somehow related to the syntactical restriction giving PPCF out of $\text{PCF}^{\mathfrak{R}}$. In fact, the morphisms between probabilistic coherence spaces describe true power series over real numbers, not just formal series as in $\mathfrak{R}^{\Pi}_{!}$. Our proof of full abstraction relies on this feature, crucially using standard tools of Calculus for handling probabilities. Let me detail a bit more the result.

The inner product of two vectors $u, v \in (\mathbb{R}^+ \cup \{\infty\})^A$ is defined as $\langle u, v\rangle \triangleq \sum_{a\in A} u_a v_a$. The polar $S^\perp$ of a subset $S \subseteq (\mathbb{R}^+ \cup \{\infty\})^A$ is

$$S^\perp \triangleq \{u \in (\mathbb{R}^+ \cup \{\infty\})^A \text{ s.t. } \forall v \in S, \langle u, v\rangle \leq 1\ \}. \tag{3.10}$$

Definition 23 ([DE11]). A probabilistic coherence space is a pair $\mathcal{A} = (|\mathcal{A}|, \mathrm{P}(\mathcal{A}))$ of a set $|\mathcal{A}|$, called web, and a subset $\mathrm{P}(\mathcal{A}) \subseteq (\mathbb{R}^+)^{|\mathcal{A}|}$ s.t.:

(i) $\mathrm{P}(\mathcal{A})^{\perp\perp} = \mathrm{P}(\mathcal{A})$,

(ii) $\forall a \in |\mathcal{A}|, \exists v \in \mathrm{P}(\mathcal{A}), v_a \neq 0$,

(iii) $\forall a \in |\mathcal{A}|, \exists p \in \mathbb{R}^+, \forall v \in \mathrm{P}(\mathcal{A}), v_a \leq p$.

Condition (i) is the important one, giving a probabilistic effect to the model. In particular, one can easily check that the set of sub-probabilistic distributions of the elements in $|\mathcal{A}|$ (i.e. $\{v\ ;\ \sum_{a\in|\mathcal{A}|} v_a \leq 1\}$) enjoys the above three conditions. However, there are probabilistic coherence spaces which are not associated with sub-probabilistic distributions: they are needed to interpret higher-order types. In general, we can say that $\mathrm{P}\left(\mathcal{A}\right)$ is a convex set, cpo with respect to the component-wise order on $(\mathbb{R}^+)^{|\mathcal{A}|}$.

Probabilistic coherence spaces yield both a $\star$-autonomous Lafont category, modeling linear logic, and a cartesian closed category, modeling λ-calculus, PPCF, etc. However, let me go directly to the point and sketch only the model of PPCF.

Definition 24 (CoKleisli category $\mathrm{PCOH}_!$, [DE11])**.** The objects are probabilistic coherence spaces $\mathcal{A}, \mathcal{B}, \mathcal{C} \dots$. A morphism from $\mathcal{A}$ to $\mathcal{B}$ is a matrix f in $(\mathbb{R}^+)^{\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)\times|\mathcal{B}|}$ such that: for every $v \in \mathrm{P}\left(\mathcal{A}\right)$, $f(v) \in \mathrm{P}\left(\mathcal{B}\right)$, where $f(v)$ is the vector of $(\mathbb{R}^+)^{|\mathcal{B}|}$ defined by

$$f(v)_b \triangleq \sum_{m\in\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)} f_{m,b} \prod_{a\in\mathrm{Supp}(m)} v_a^{m(a)}, \tag{3.11}$$

where we recall that $\mathrm{Supp}\left(m\right)$ is the support of the multiset m and $m(a)$ is the number of occurrences of a in m.

Given a type A, its denotation $[\![A]\!]$ is a probabilistic coherence space defined as follows.

- if A is a ground type, then:
  - $|[\![A]\!]|$ is the set of closed values of type A e.g. $|[\![\mathtt{1}]\!]| = \{\mathtt{skip}\}$, $|[\![\mathtt{Bool}]\!]| = \{\mathtt{tt}, \mathtt{ff}\}$, $|[\![\mathtt{Int}]\!]| = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \dots\}$;
  - $\mathrm{P}\left([\![A]\!]\right)$ is the set of sub-probability distributions of closed values e.g. $\mathrm{P}\left([\![\mathtt{1}]\!]\right) = [0, 1]$, $\mathrm{P}\left([\![\mathtt{Bool}]\!]\right) = \{(p, q) \text{ s.t. } p + q \leq 1\}$, $\mathrm{P}\left([\![\mathtt{Int}]\!]\right) = \{(v_n) \text{ s.t. } \sum_n v_n \leq 1\}$;
- if A is a functional type $B \to C$, then:
  - $|[\![A]\!]|$ is the index set $\mathcal{M}_{\mathrm{f}}\left(|[\![B]\!]|\right) \times |[\![C]\!]|$;
  - $\mathrm{P}\left([\![A]\!]\right)$ is the set of matrices f describing morphisms in $\mathrm{PCOH}_!$, i.e. such that for all $x \in \mathrm{P}\left([\![B]\!]\right)$, $f(x) \in \mathrm{P}\left([\![C]\!]\right)$.

It must be stressed that at higher-order types, vectors in $\mathrm{P}\left([\![A]\!]\right)$ might not be sub-probability distributions. For example, one can show that there are matrices with arbitrary large scalars in $\mathrm{P}\left([\![\mathtt{Bool} \to \mathtt{1}]\!]\right)$.

There is a forgetful functor from $\mathrm{PCOH}_!$ to $\mathfrak{R}_!^{\oplus}$, mapping a probabilistic coherence space $\mathcal{A}$ to its web $|\mathcal{A}|$ and being the identity on morphisms. The interpretation of PPCF commutes with this functor. In particular, the interpretation of terms is exactly the same as in $\mathfrak{R}_!^{\oplus}$.

Theorem 25 ([EPT14])**.** *The category* $\mathrm{PCOH}_!$ *(and so* $\mathfrak{R}_!^{\oplus}$*) is fully abstract for probabilistic* PCF*, i.e.* $[\![M]\!] = [\![N]\!]$ *iff for every context* $C(\_)$ *of type* `Bool`*,* $C(M)$ *and* $C(N)$ *reduces to* `tt` *with equal probability.*

To our knowledge, no model of probabilistic PCF has been proved to be fully abstract, yet. Games semantics provide fully abstract models of standard PCF via an extensional collapse [HO00, AJM00]. This technique has been adapted to probabilistic game semantics, providing a fully abstract model of probabilistic idealized ALGOL [DH02] (an extension of probabilistic PCF with references). Probabilistic coherence spaces characterize the operational indistinguishability without the need of an extensional collapse and deals with a functional probabilistic language with no references.

The proof of Theorem 25 has an interest in its own. The implication denotational equality $\Rightarrow$ observational equality is a straight corollary of adequacy (i.e. Theorem 22 for PCOH$_!$, proved in [DE11]). As for the converse, one supposes $[\![M]\!] \neq [\![N]\!]$ and then tries to find a context $C(\_)$ where M and N behave differently. The first step is to make more tractable $[\![M]\!]$ and $[\![N]\!]$. Indeed the two matrices describe power series (recall the discussion in Section 3.1) on a number of unknowns equal to the cardinality of $|[\![A]\!]|$ (with A the type of M, N), which can be infinite in general. Power series of infinite dimension are quite difficult to deal with, so we prefer to move to finite dimension. $[\![M]\!] \neq [\![N]\!]$ means that there exists at least a web element $a \in |[\![A]\!]|$ such that $[\![M]\!]_a \neq [\![N]\!]_a$. By looking at a we construct a term F^a of type $A \to \texttt{Bool}^k \to \texttt{Bool}$ (with k depending on the size of a) such that $[\![F^a M]\!] \neq [\![F^a N]\!]$. Now the power series have type $\texttt{Bool}^k \to \texttt{Bool}$, which are much more tractable, because have only a finite number of unknowns (namely, 2^k). From $[\![F^a M]\!] \neq [\![F^a N]\!]$ we get that $[\![F^a M]\!] - [\![F^a N]\!]$ is a non-zero matrix, hence it defines a power series $g : \mathbb{R}^{2^k} \mapsto \mathbb{R}$ with a non-zero coefficient. By iterated derivation one can prove that g does not vanish on any dense subset of an open set of the domain of g. This means that there exist rational sub-probability distributions of booleans $(p_1, q_1), \dots, (p_k, q_k)$ such that $g(p_1, q_1) \dots (p_k, q_k)$ is non-null. This means: $[\![F^a M]\!] (p_1, q_1) \dots (p_k, q_k) \neq [\![F^a N]\!] (p_1, q_1) \dots (p_k, q_k)$. By using fix-points and $\frac{1}{2}\texttt{Coin}$ constructors, we can represent each rational sub-probability distribution (p_i, q_i) of type $\texttt{Bool}$ with a term of the syntax, let me denote it by $p_i\texttt{tt} \oplus q_i\texttt{ff}$. The adequacy theorem then gives that $(F^a M)(p_1\texttt{tt} \oplus q_1\texttt{ff}) \dots (p_n\texttt{tt} \oplus q_n\texttt{ff})$ reduces to $\texttt{tt}$ with probability $[\![F^a M]\!] (p_1, q_1) \dots (p_k, q_k)$ which is different from $[\![F^a N]\!] (p_1, q_1) \dots (p_k, q_k)$, i.e. the probability of $(F^a N)(p_1\texttt{tt} \oplus q_1\texttt{ff}) \dots (p_n\texttt{tt} \oplus q_n\texttt{ff})$. We conclude by setting the context equal to $(F^a(\_))(p_1\texttt{tt} \oplus q_1\texttt{ff}) \dots (p_n\texttt{tt} \oplus q_n\texttt{ff})$.

The unusual (for denotational semantics) ingredient of the above reasoning is in the use of iterated derivations on g for proving the existence of the scalars (p_i, q_i). This is a genuine example of the interest of the quantitative semantics approach, interpreting programs as power series.

Open problem 9 ($\star\star$)**.** *The first denotational approach to probabilistic functional languages was based on Jones and Plotkin's probabilistic powerdomains [SD80, JP89]. Programs are interpreted as Scott's continuous func-*

tions from the input domain to the power-domain of the output domain. Intuitively, a program takes an input and returns a sub-probability distribution on outputs. This line of work has been continued by the continuous random variable construction [GLV11]. What is the relation between these semantics and probabilistic coherence spaces? Are these semantics (or at least their extensional collapse) fully abstract? Indeed, the power-domains approach follows Moggi's computational λ-calculus [Mog89], which is call-by-value. It might be possible then that one should compare probabilistic power-domains with a category of probabilistic coherence spaces implementing a call-by-value strategy.

Open problem 10 ($\star\star$)**.** *In the extended abstract [EPT11], we study a reflexive object of* $\mathrm{PCOH}_!$ *modeling untyped probabilistic λ-calculus. We achieve an adequacy theorem, but it is an open issue whether the model enjoys full abstraction also. Notice that one cannot use directly the above proof of Theorem 25, because based on the simple types discipline of* PPCF*.*

3.4 Semantics of quantum primitives (since 2010)

An important problem in the semantics of quantum languages is how to combine quantum computing with higher-order functions, or in other words, how to design a functional quantum programming language. A syntactic answer to this question was arguably given with the design of the quantum λ-calculus [Val08, SV06]. This language has a well-defined syntax and operational semantics. However, the question of how to give a denotational semantics to the language turned out to be difficult, and has remained open for many years [Sel04b, SV09]. One reason that designing such a semantics is difficult is that quantum computation is inherently defined on *finite dimensional* Hilbert spaces, whereas the semantics of higher-order functional programming languages, including such features as infinite data types and recursion, is inherently infinitary.

In a joint project with Selinger (professor at Dalhousie University) and Valiron (ATER at Paris 13 at the time the project started, now post-doc at Paris 7), we give a quantitative semantics of a higher-order quantum language with full recursion and an infinite data type. The extended abstract [PSV14] has been recently accepted.

The starting point is Selinger's category CPM of *completely positive maps* [Sel04a]. This category is suitable for implementing first-order quantum languages, but it has no higher-order duplication and, consequently, cannot model higher-order languages. Basically, we extend CPM in order to make meaningful Equation (3.2), allowing for linear logic exponentials. The result is the model $\overline{\mathrm{CPM}}_s^{\oplus}$ (Definition 26) interpreting a higher-order quantum program as an infinite dimensional block matrix whose blocks are finite dimensional completely positive maps.

Our main result is the adequacy of the model with respect to the operational semantics (Theorem 28).

The grammar of the quantum language λ_{Q} presented in [PSV14] is given in Figure 3.3, the simple type assignment system is in Figure 3.4 and the operational semantics in Figure 3.5. From my point of view, this language is a significant improvement with respect to the previous quantum λ-calculi: basically, λ_{Q} is a true extension of linear logic with quantum data types. Previous languages (e.g. [Val08, SV06]) had a more involved type system and term grammar. Of course, it is not an accident that the quest for a denotational semantics helps in the design of a clean language.

The language λ_{Q} is based on a minimal fragment of propositional linear logic and not of intuitionistic logic. This is typical of quantum languages, in order to write types enforcing a linear use of quantum data, coherently with the no-cloning property of quantum physics. Besides, the operational semantics is call-by-value, allowing for an easy encoding of the usual quan-

$$
\begin{array}{l}
\textit{Terms} \quad M,N,P \quad ::= \\
\quad x \mid \lambda x^A.M \mid MN \mid \texttt{skip} \mid M;N \mid \\
\quad M\otimes N \mid \texttt{let}\ x^A\otimes y^B \ = \ M\ \texttt{in}\ N \mid \\
\quad \texttt{in}_\ell\ M \mid \texttt{in}_r\ M \mid \texttt{match}\ P\ \texttt{with}\ (x^A : M | y^B : N) \mid \\
\quad \texttt{split}^A \mid \texttt{letrec}\ f^{A\multimap B}x = M\,\texttt{in}\,N \mid \texttt{meas} \mid \texttt{new} \mid U \\
\\
\textit{Values} \quad V,W \quad ::= \\
\quad x \mid c \mid \lambda x^A.M \mid V\otimes W \mid \texttt{in}_\ell\ V \mid \texttt{in}_r\ W \\
\\
\textit{Types} \quad A,B,C \quad ::= \\
\quad \textbf{qubit} \mid A\multimap B \mid !(A\multimap B) \mid 1 \mid A\otimes B \mid A\oplus B \mid A^\ell.
\end{array}
$$

Figure 3.3: Grammars of terms, values and types. The constant U ranges over a set of unitary transformations on quantum bits. In the grammar of values, the variable c stands for any constant `skip`, `split`, `meas`, `new`. Finally, we define **bit** $\triangleq 1\oplus 1$, so that `tt` $\triangleq$ $\texttt{in}_r$ `skip` and `ff` $\triangleq$ $\texttt{in}_\ell$ `skip`.

tum algorithms. For example, in [PSV14, Example 6] we describe the term corresponding to the quantum teleportation protocol, and give its denotation in the model [PSV14, Table 6].

The language has full-recursion (the `letrec` constructor) and a list data type A^ℓ, for any A. Quantum data is handled by the ground type **qubit** and three kinds of constants: `meas`, transforming a **qubit** into a **bit** $\triangleq 1\oplus 1$; `new`, creating a new **qubit**; and a set of elementary unitary transformations U on **qubit**'s. Variables of type **qubit** are pointers to an external quantum array, so that the operational semantics (Figure 3.5) is defined on a quantum closure, i.e. a triplet $[q,\ell,M]$ of a term M with a number $n\geq 0$ of free variables of type **qubit**, a normalized vector $q\in\mathbb{C}^{2^n}$ ($\mathbb{C}$ denoting the field of complex numbers), describing the state of a quantum array of n qubits, and a linking function ℓ fixing a one-to-one correspondence between the free variables of M and the qubits in q.

The operational semantics is probabilistic, any ers happening with probability 1, except for qubit measurements (right column of Figure 3.5(b)), where the redex `meas` x rewrites into `tt` or `ff` with a probability depending on the amplitudes of the component of the quantum array pointed by x. The ers changes also the whole state of the quantum array: this side effect is due to the entanglement of the qubits in the quantum array, and it makes the operational semantics particularly subtle to model.

Consider the action of the operational semantics on the derivations of the typing system of Figure 3.4. Notice that the reduction steps are implement-

$$\frac{A \text{ not !-formula}}{!\Delta, x : A \vdash x : A}\,ax \qquad \frac{A \text{ not a !-formula}}{!\Delta, x : !A \vdash x : A}\,axd \qquad \frac{!\Delta \vdash V : A \multimap B \quad V \text{ value}}{!\Delta \vdash V : !(A \multimap B)}\,p$$

$$\frac{}{!\Delta \vdash \texttt{skip} : 1}\,1_I \qquad \frac{!\Delta, \Gamma \vdash M : 1 \quad !\Delta, \Sigma \vdash N : A}{!\Delta, \Gamma, \Sigma \vdash M;N : A}\,1_E$$

$$\frac{\Delta, x : A \vdash M : B}{\Delta \vdash \lambda x^A.M : A \multimap B}\,\multimap_I \qquad \frac{!\Delta, \Gamma \vdash M : A \multimap B \quad !\Delta, \Sigma \vdash N : A}{!\Delta, \Gamma, \Sigma \vdash MN : B}\,\multimap_E$$

$$\frac{!\Delta, \Gamma \vdash M : A \quad !\Delta, \Sigma \vdash N : B}{!\Delta, \Gamma, \Sigma \vdash M \otimes N : A \otimes B}\,\otimes_I \qquad \frac{!\Delta, \Gamma \vdash M : A \otimes B \quad !\Delta, \Sigma, x : A, y : B \vdash N : C}{!\Delta, \Gamma, \Sigma \vdash \texttt{let}\ x^A \otimes y^B \ = \ M\ \texttt{in}\ N : C}\,\otimes_E$$

$$\frac{!\Delta, \Gamma \vdash M : A}{!\Delta, \Gamma \vdash \texttt{in}_\ell\ M : A \oplus B}\,\oplus_I^\ell \qquad \frac{!\Delta, \Gamma \vdash M : B}{!\Delta, \Gamma \vdash \texttt{in}_r\ M : A \oplus B}\,\oplus_I^r$$

$$\frac{\begin{array}{c} !\Delta, \Sigma, y : B \vdash N : C \\ !\Delta, \Gamma \vdash P : A \oplus B \quad !\Delta, \Sigma, x : A \vdash M : C \end{array}}{!\Delta, \Gamma, \Sigma \vdash \texttt{match}\ P\ \texttt{with}\ (x^A : M | y^B : N) : C}\,\oplus_E$$

$$\frac{!\Delta, \Gamma \vdash M : 1 \oplus (A \otimes A^\ell)}{!\Delta, \Gamma \vdash M : A^\ell}\,-_I^\ell \qquad \frac{}{!\Delta \vdash \texttt{split}^A : A^\ell \multimap 1 \oplus (A \otimes A^\ell)}\,\texttt{split}$$

$$\frac{!\Delta, f : !(A \multimap B), x : A \vdash M : B \quad !\Delta, \Gamma, f : !(A \multimap B) \vdash N : C}{!\Delta, \Gamma \vdash \texttt{letrec}\ f^{A \multimap B}\ x = M\ \texttt{in}\ N : C}\,\texttt{rec}$$

$$\frac{}{!\Delta \vdash \texttt{meas} : \mathbf{qubit} \multimap \mathbf{bit}} \quad \frac{}{!\Delta \vdash \texttt{new} : \mathbf{bit} \multimap \mathbf{qubit}} \quad \frac{U \text{ of arity } n}{!\Delta \vdash U : \mathbf{qubit}^{\otimes n} \multimap \mathbf{qubit}^{\otimes n}}$$

Figure 3.4: typing rules. The contexts Γ and Σ are assumed to be without !-formulas.

$$[q, \ell, (\lambda x^A.M)\, V] \xrightarrow{1} [q, \ell, M\{V/x\}]$$
$$[q, \ell, \texttt{skip};N] \xrightarrow{1} [q, \ell, N]$$
$$[q, \ell, \texttt{let}\ x^A \otimes y^B \ = \ V \otimes W\ \texttt{in}\ N] \xrightarrow{1} [q, \ell, N\{V/x, W/y\}]$$
$$[q, \ell, \texttt{match}\ (\texttt{in}_\ell\ V)\ \texttt{with}\ (x^A : M | y^B : N)] \xrightarrow{1} [q, \ell, M\{V/x\}]$$
$$[q, \ell, \texttt{match}\ (\texttt{in}_r\ V)\ \texttt{with}\ (x^A : M | y^B : N)] \xrightarrow{1} [q, \ell, N\{V/y\}]$$
$$[q, \ell, \texttt{split} V] \xrightarrow{1} [q, \ell, V]$$
$$[q, \ell, \texttt{letrec}\ f\, x = M\ \texttt{in}\ N] \xrightarrow{1} [q, \ell, N\{(\lambda x^A.\texttt{letrec}\ f\, x = M\ \texttt{in}\ M)/f\}]$$

(a) Classical control.

$$[q, \ell, U(x_1 \otimes \cdots \otimes x_k)] \xrightarrow{1} [q', \ell, x_1 \otimes \cdots \otimes x_k]$$

$$[q, \emptyset, \texttt{new ff}] \xrightarrow{1} [q \otimes |0\rangle, \{y \mapsto n+1\}, y] \quad [\alpha q_0 + \beta q_1, \{x \mapsto i\}, \texttt{meas}\ x] \xrightarrow{|\beta^2|} [q_1', \emptyset, \texttt{tt}]$$
$$[q, \emptyset, \texttt{new tt}] \xrightarrow{1} [q \otimes |1\rangle, \{y \mapsto n+1\}, y] \quad [\alpha q_0 + \beta q_1, \{x \mapsto i\}, \texttt{meas}\ x] \xrightarrow{|\alpha^2|} [q_0', \emptyset, \texttt{ff}]$$

(b) Quantum data. The variable y is fresh. The quantum state q' in the first rule is obtained by applying the k-ary unitary gate U to the qubits $\ell(x_1), \ldots, \ell(x_k)$. In the rules about measurements, if q_0 and q_1 are of the form $\sum_j \alpha_j^0 |\phi_j\rangle \otimes |0\rangle \otimes |\psi_j\rangle$, $\sum_j \alpha_j^1 |\phi_j\rangle \otimes |1\rangle \otimes |\psi_j\rangle$, then q_0' and q_1' are resp. $\sum_j \alpha_j^0 |\phi_j\rangle \otimes |\psi_j\rangle$, $\sum_j \alpha_j^1 |\phi_j\rangle \otimes |\psi_j\rangle$, where ϕ_j has dimension $\ell(x) - 1$ (so that the measured qubit is $\ell(x)$).

$$[q, \ell, MN] \xrightarrow{p} [q', \ell', M'N] \quad [q, \ell, M \otimes N] \xrightarrow{p} [q', \ell', M' \otimes N] \quad [q, \ell, \texttt{in}_\ell M] \xrightarrow{p} [q', \ell', \texttt{in}_\ell M']$$
$$[q, \ell, VM] \xrightarrow{p} [q', \ell', VM'] \quad [q, \ell, V \otimes M] \xrightarrow{p} [q', \ell', V \otimes M'] \quad [q, \ell, \texttt{in}_r M] \xrightarrow{p} [q', \ell', \texttt{in}_r M']$$
$$[q, \ell, M;N] \xrightarrow{p} [q', \ell', M';N] \quad [q, \ell, \texttt{let}\ x^A \otimes y^B = M\,\texttt{in}\,N] \xrightarrow{p} [q', \ell', \texttt{let}\ x^A \otimes y^B = M'\,\texttt{in}\,N]$$
$$[q, \ell, \texttt{match}\ M\ \texttt{with}\ (x^A : P | y^B : N)] \xrightarrow{p} [q', \ell', \texttt{match}\ M'\ \texttt{with}\ (x^A : P | y^B : N)]$$

(c) Congruence rules, under the hypothesis that for some ℓ_0 we have $\ell = \ell_0 \uplus \ell|_M$, $\ell' = \ell_0 \uplus \ell'|_{M'}$ and $[q, \ell|_M, M] \xrightarrow{p} [q', \ell'|_{M'}, M']$.

Figure 3.5: elementary reduction steps (ers) on closures.

ing a strategy on the standard cut-elimination of linear logic, corresponding to reduce cuts below any promotion rule (in the proof nets formalism, cuts outside all exponential boxes). This is the crucial property making quantitative semantics to work.

Let us move to denotational semantics. As previously written, the starting point is Selinger's category CPM of completely positive maps [Sel04a]. Selinger's category is the free *finite* biproduct completion of the category having as objects natural numbers and as morphisms from n to m the completely positive maps from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$. Indeed, the space of square matrices $\mathbb{C}^{2\times 2}$ is suitable for interpreting the basic data type of quantum computing, **qubit**.

Mathematically, a quantum bit is a vector in $\mathbb{C}^2$, which can be expressed as a linear combination $p|0\rangle + q|1\rangle$ of the standard basis vectors $|0\rangle \triangleq \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $|1\rangle \triangleq \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$. However, quantum languages deal with probabilistic distributions of quantum bits and not just with "pure" quantum bits. The difference between the two is known in quantum mechanics as the difference between mixed states and pure states. Von Neumann gave a convenient representation of the mixed states as density matrices, which are particular square matrices over complex numbers. In particular, the density matrix representing a probabilistic distribution of a quantum bit is in $\mathbb{C}^{2\times 2}$.

Square matrices can be partially ordered by the so-called Löwner order, commonly used in linear programming: $f \sqsubseteq g$ iff $g-f$ is a positive hermitian. I will not enter into the details, just let me underline that one astonishing outcome of [Sel04a], at least for me, is that Selinger succeeded in showing that the Löwner order can be used as an informative order on quantum data types, as it is standard in denotational semantics. In particular, the zero matrix $\mathbf{0}$ represents the absence of information, quantum data are denoted by positive matrices (i.e. greater or equal to $\mathbf{0}$) and quantum programs are completely positive maps, i.e. linear maps mapping positive matrices to positive matrices (also when tensorized with the identity), and in particular they are monotone and preserving directed suprema (when they exist).

The problem is that Selinger's CPM is not a model of linear logic, since it misses exponentials. No exponentials, no higher-order duplication; no higher-order duplication, no functional programming. The main goal of [PSV14] is to extend CPM in order to be able to apply Equation (3.2), i.e. to define exponentials as infinite biproducts of symmetric tensor powers. However, CPM misses two essential requirements for doing that:

(i) CPM has only finite biproducts,

(ii) CPM has no symmetric tensor.

The problem with item (i) is that composing infinite biproducts of completely positive maps (i.e. matrices of infinite dimension) leads to convergency issues (recall the sum in Equation (3.4)) since Löwner order is not complete on positive matrices. Our solution to this problem is rather brutal, consisting in choosing a suitable cpo-completion of the positive cone of the Löwner order.

Open problem 11 ($\star\star$)**.** *Transforming the Löwner positive cone into a cpo adds some new "infinite" elements which are actually useless for interpreting programs. In fact, we proved that such elements do not appear in the denotations of* λQ *programs. This means that it exists a subcategory modeling* λQ *and avoiding the cpo-completion. Is it possible to catch this subcategory by introducing a kind of double polar closure, in the spirit of probabilistic coherence spaces (Definition 23)? Indeed, Girard proposed the notion of* quantum coherence space *by setting two hermitians* v, u polar *whenever the trace of the matrix product* vu *is in* $[0, 1]$ *(the trace of an hermitian is always a real number) [Gir04]. Quantum coherence spaces might give a model of linear logic (the exponentials have not been detailed in [Gir04]), however they are unsuitable for* λQ *because the tensor is too small, preventing entanglements of its components. See [Sel04b] for more details.*

The problem with item (ii) is that the symmetric tensor of a space $\mathbb{C}^{n\times n}$ is a subspace of $\mathbb{C}^{n\times n} \otimes \mathbb{C}^{n\times n} \simeq \mathbb{C}^{n^2\times n^2}$ which has not a square dimension. For example, the symmetric two-fold tensor power of $\mathbb{C}^{2\times 2}$ is the set of matrices of a certain symmetry, namely of the form:

$$\begin{pmatrix} a & b & b & c \\ d & e & f & g \\ d & f & e & g \\ h & i & i & j \end{pmatrix}, \text{ for } a, b, c, d, e, f, g, h, i, j \in \mathbb{C}.$$

This space has dimension 10, hence it cannot be conveniently associated with an object of Selinger's CPM – the object 3 being associated with $\mathbb{C}^{3\times 3}$, of dimension 9, and the object 4 with $\mathbb{C}^{4\times 4}$, of dimension 16. In fact, the symmetric two-fold tensor power of $\mathbb{C}^{2\times 2}$ can be described as the subspace of $\mathbb{C}^{4\times 4}$ invariant under the action of the subgroup $\{(1, 2, 3, 4), (1)(2, 3)(4)\}$ of permutations over $\{1, \dots, 4\}$, acting on rows and columns simultaneously.

Actually, this is true in general: Selinger's CPM can be lifted to a category CPM$_s$ allowing a convenient representation of symmetric tensors. The objects are families of pairs (n, G) of a natural number n and a group G of permutations over n. The space of matrices associated with (n, G) is the subspace of $\mathbb{C}^{n\times n}$ made of those matrices which are invariant under the action of G.

To sum up, our model of λQ is defined in the following category.

Definition 26 (The category $\overline{\mathrm{CPM}}_s^{\oplus}$, [PSV14])**.** The objects are (possibly infinite) indexed families $\{(n_a, G_a)\}_{a\in|\mathcal{A}|}$, where the index set $|\mathcal{A}|$ is called the web of $\mathcal{A}$ and, for every $a \in |\mathcal{A}|$, n_a is a natural number and G_a a group of permutations over n_a. The morphisms from $\{(n_a, G_a)\}_{a\in|\mathcal{A}|}$ to

$\{(m_b, H_b)\}_{b\in|\mathcal{B}|}$ are matrices f indexed by $|\mathcal{A}| \times |\mathcal{B}|$ and such that $f_{a,b}$ is either a completely positive map from $\mathbb{C}^{n_a\times n_a}$ to $\mathbb{C}^{m_b\times m_b}$ invariant under the actions of G_a and H_b (i.e. such that $G_a\,;\, f_{a,b}\,;\, H_b = f_{a,b}$), or the top ∞.

Theorem 27 ([PSV14])**.** *The category $\overline{\mathrm{CPM}}_s^{\oplus}$ is a $\star$-autonomous (in fact compact closed), cpo-enriched, Lafont category. In particular, $\overline{\mathrm{CPM}}_s^{\oplus}$ yields a model of propositional linear logic.*

The interpretation of λ_{Q} into $\overline{\mathrm{CPM}}_s^{\oplus}$ extends the standard interpretation of linear logic into a Lafont category to also include the quantum features of the calculus. So, for example:

$[\![\mathbf{qubit}]\!]$ is the singleton web object $\{(2, \{\mathrm{id}\})_*\}$, associated with $\mathbb{C}^{2\times 2}$;

$[\![1]\!]$ is the tensor unit $\mathbf{1} = \{(1, \{\mathrm{id}\})_*\}$ of $\overline{\mathrm{CPM}}_s^{\oplus}$, associated with $\mathbb{C}$;

$[\![\mathbf{bit}]\!] \triangleq [\![1 \oplus 1]\!]$ is the two-element family $\{(1, \{\mathrm{id}\})_{\mathtt{tt}}, (1, \{\mathrm{id}\})_{\mathtt{ff}}\}$, describing the biproduct $\mathbb{C} \oplus \mathbb{C}$.

$[\![1 \multimap \mathbf{qubit}]\!]$ is equal to $[\![1 \otimes \mathbf{qubit}]\!]$ (the category is indeed compact closed) and hence isomorphic to $[\![\mathbf{qubit}]\!]$. The same for $[\![1 \multimap \mathbf{bit}]\!]$.

$[\![!(1 \multimap \mathbf{bit})]\!]$ is the biproduct of the symmetric tensor powers of $[\![\mathbf{bit}]\!]$, concretely, the web is the set of finite multisets over the set $\{\mathtt{tt}, \mathtt{ff}\}$, and for any multiset $[\mathtt{tt}^k, \mathtt{ff}^q]$ the associated pair is simply $(1, \{\mathrm{id}\})$.

$[\![!(1 \multimap \mathbf{qubit})]\!]$ is the biproduct of the symmetric tensors of $[\![\mathbf{qubit}]\!]$, which is much more complex than $[\![!(1 \multimap \mathbf{bit})]\!]$. The web is the set of finite multisets of the singleton $\{*\}$. For any multiset $[*^k]$ the associated pair can be expressed with $(2^k, S_k)$, where S_k is the group of all permutations over $\{1, \ldots, k\}$ acting on 2^k by looking at it as the set of k-length sequences of 0 and 1: $(i_1, \ldots, i_k)$.

A term M of type A in an environment Γ is denoted as a morphism $[\![M]\!]^{\Gamma\vdash A}$ from $[\![\Gamma]\!]$ to $[\![A]\!]$. In particular, if Γ is empty and A is the unit 1, then $[\![M]\!]^{\vdash 1}$ is a non-negative real number (i.e. a positive hermitian of $\mathbb{C}$). The main property of the model proved in [PSV14] is the adequacy: the scalar $[\![M]\!]^{\vdash 1}$ is equal to the probability that the evaluation of the quantum closure $[|\,\rangle, |\,\rangle, M]$ terminates into $[|\,\rangle, |\,\rangle, \mathtt{skip}]$.

Theorem 28 ([PSV14])**.** *Let M be a closed term of unit type. Then*

$$[\![M]\!]_*^{\vdash 1} = p \text{ iff } [|\,\rangle, |\,\rangle, M] \Downarrow^p [|\,\rangle, |\,\rangle, \mathtt{skip}].$$

Open problem 12 ($\star\star$)**.** *Our proof of Theorem 28 is syntactical, following a technique of [HH11], consisting in defining an auxiliary language with a* *bounded* `letrec` *operator. Indeed, we failed in giving a suitable logical relation for λ_{Q}, mainly because of the side effects generated by the qubits measurements (recall Figure 3.5(b)). However, I consider worthwhile developing a notion of logical relation for λ_{Q}, allowing for a general way of transforming properties on ground types into properties on higher-order types. It might be possible that this issue is related to the open problem 11: indeed the double polar closed sets can be seen as a kind of logical (unary) relations.*

In recent years, a number of different models of quantum lambda calculi have been proposed, with varying degrees of success. One approach [Mal10] was to construct a semantics of higher-order quantum computation by applying a presheaf construction to a first-order model. This indeed succeeds in yielding a semantics of the quantum λ-calculus, albeit without recursion. The main drawback of presheaf models is the absence of recursion, and the fact that such models are relatively difficult to reason about. Another remarkable approach [HH11] was based on the Geometry of Interaction. Starting from a traced monoidal category of basic quantum operations, Hasuo and Hoshino applied a sequence of categorical constructions, which eventually yielded a model of higher-order quantum computation. The problem with this approach is that the tensor product constructed from the geometry-of-interaction construction does not coincide with the tensor product of the underlying physical data types. Therefore, the semantics drops the possibility of entangled states, and thereby fails to model one of the defining features of quantum computation.

Our model permits general entanglement, can represent *infinite dimensional* structures, and is expressive enough to describe recursive types, such as lists of qubits, and to model recursion. One thing that our model explains and illustrates clearly is the distinction between the quantum and classical parts of λ_{Q}. The quantum part is described by completely positive maps (finite dimension), whereas the classical control is given by the Lafont category (i.e. linear logic). The model demonstrates that the two "universes" work well together, but also – surprisingly – that they do not mix too much, even at higher order types (we always have an *infinite* dimensional block matrix whose blocks are *finite* dimensional completely positive maps). The control flow is completely handled by the biproduct completion, and not by the CPM structure.

Bibliography

[Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543–574, June 1994.

[AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, December 2000.

[Bar71] Hendrik Barendregt. *Some extensional term models for combinatory logics and lambda-calculi.* PhD thesis, University of Utrecht, 1971.

[Bar84] Henk Barendregt. *The Lambda-Calculus, its Syntax and Semantics.* Stud. Log. F. Math., vol. 103. North-Holland, 1984.

[BBE06] Alexandra Bac Bruasse and Thomas Ehrhard. The relational (not quite a) model of second-order linear logic. Preprint, 2006.

[BCL99] Gérard Boudol, Pierre-Louis Curien, and Carolina Lavatelli. A semantics for lambda calculi with resources. *Math. Struct. Comp. Sci.*, 9(4):437–482, 1999.

[Béc98] Denis Béchet. Minimality of the correctness criterion for multiplicative proof nets. *Mathematical Structures in Computer Science*, 8(6):543–558, 1998.

[BHP13] Pierre Boudes, Fanny He, and Michele Pagani. A characterization of the Taylor expansion of lambda-terms. In *Proceedings of the 22nd EACSL Annual Conference Computer Science Logic CSL 2013, September 2-5, 2013, Torino, Italy*, 2013.

[BL96] Gérard Boudol and Cosimo Laneve. The discriminating power of multiplicities in the lambda-calculus. *Inf. Comput.*, 126(1):83–102, 1996.

[Böh68] Corrado Böhm. Alcune proprietà delle forme beta-eta-normali del lambda-kappa-calcolo. Pubblicazioni dell'I.A.C. 696, Istituto per le Applicazioni del Calcolo, Roma, 1968.

[Bou93] Gérard Boudol. The Lambda-Calculus with Multiplicities. *INRIA Report 2025*, 1993.

[BPS94] R. F. Blute, Prakash Panangaden, and R. A. G. Seely. Fock space: A model of linear exponential types, 1994.

[Bre13] Flavien Breuvart. The resource lambda calculus is short-sighted in its relational model. In Masahito Hasegawa, editor, *Typed Lambda Calculi and Applications, 11th International Conference, TLCA 2013, Eindhoven, The Netherlands, June 26-28, 2013. Proceedings*, volume 7941 of *Lecture Notes in Computer Science*, pages 93–108. Springer, 2013.

[Dan90] Vincent Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul).* Thèse de doctorat, Université Paris VII, 1990.

[dC07] Daniel de Carvalho. *Sémantiques de la logique linéaire et temps de calcul.* PhD thesis, Université Aix-Marseille II, 2007. Thèse de Doctorat.

[dC09] Daniel de Carvalho. Execution Time of λ-Terms via Denotational Semantics and Intersection Types. Preprint, 2009.

[DCMP13] Alejandro Díaz-Caro, Giulio Manzonetto, and Michele Pagani. Call-by-value non-determinism in a linear logic type discipline. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 164–178. Springer, 2013.

[dCPTdF11] Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A semantic measure of the execution time in linear logic. *Theorical Computer Science, Special issue Girard's Festschrift*, 412(20):1884–1902, 2011.

[DE11] Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inf. Comput.*, 209(6):966–991, 2011.

[dF10] Marc de Falco. An explicit framework for interaction nets. *Logical Methods in Computer Science*, 6(4), 2010.

[DH02] Vincent Danos and Russel Harmer. Probabilistic game semantics. *ACM Transactions on Computational Logic*, 3(3):359–382, July 2002.

[DR89] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[Ehr95] Thomas Ehrhard. Hypercoherence: A strongly stable model of linear logic. In *Advances in Linear Logic*, pages 83–108. Cambridge University Press, 1995.

[Ehr02] Thomas Ehrhard. On Köthe sequence spaces and linear logic. *Math. Struct. Comput. Sci.*, 12:579–623, 2002.

[Ehr05] Thomas Ehrhard. Finiteness spaces. *Math. Struct. Comput. Sci.*, 15(4):615–646, 2005.

[Ehr10] Thomas Ehrhard. A finiteness structure on resource terms. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 402–410. IEEE Computer Society, 2010.

[Ehr11] Thomas Ehrhard. A model-oriented introduction to differential linear logic. Submitted for publication, 2011.

[Ehr12a] Thomas Ehrhard. Collapsing non-idempotent intersection types. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France*, volume 16 of *LIPIcs*, pages 259–273, 2012.

[Ehr12b] Thomas Ehrhard. The Scott model of linear logic is the extensional collapse of its relational model. *Theor. Comput. Sci.*, 424:20–45, 2012.

[EL07] Thomas Ehrhard and Olivier Laurent. Interpreting a finitary pi-calculus in differential interaction nets. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 333–348. Springer, 2007.

[EPT11] Thomas Ehrhard, Michele Pagani, and Christine Tasson. The Computational Meaning of Probabilistic Coherence Spaces. In Martin Grohe, editor, *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS 2011)*, IEEE Computer Society Press, pages 87–96, 2011.

[EPT14] Thomas Ehrhard, Michele Pagani, and Christine Tasson. Probabilistic Coherence Spaces are Fully Abstract for Probabilistic PCF. In P. Sewell, editor, *The 41th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL14, San Diego, USA*. ACM, 2014.

[ER03] Thomas Ehrhard and Laurent Regnier. The Differential Lambda-Calculus. *Theor. Comput. Sci.*, 309(1):1–41, 2003.

[ER06a] Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms. In *CiE*, volume 3988 of *LNCS*, pages 186–197, 2006.

[ER06b] Thomas Ehrhard and Laurent Regnier. Differential Interaction Nets. *Theor. Comput. Sci.*, 364(2):166–195, 2006.

[ER08] Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor Expansion of Ordinary Lambda-Terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.

[Gan80] R. O. Gandy. Proofs of strong normalization. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 457–477. Academic Press Limited, 1980.

[Gim11] Stéphane Gimenez. Realizability proof for normalization of full differential linear logic. In *Typed Lambda Calculi and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings*, volume 6690 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2011.

[Gir72] Jean-Yves Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur.* Thèse de doctorat, Université Paris 7, 1972.

[Gir86] Jean-Yves Girard. The system F of variable types, fifteen years later. *Theor. Comput. Sci.*, 45:159–192, 1986.

[Gir87a] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[Gir87b] Jean-Yves Girard. *Proof Theory and Logical Complexity.* Studies in Proof-theory. Bibliopolis, Napoli, 1987.

[Gir88] Jean-Yves Girard. Normal functors, power series and lambda-calculus. *Ann. Pure Appl. Logic*, 37(2):129–177, 1988.

[Gir01] Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001.

[Gir04] Jean-Yves Girard. Between logic and quantic: a tract. In Thomas Ehrhard, Jean-Yves Girard, Paul Ruet, and Philip Scott, editors, *Linear Logic in Computer Science*, volume 316 of *London Math. Soc. Lect. Notes Ser.* CUP, 2004.

[GLV11] Jean Goubault-Larrecq and Daniele Varacca. Continuous random variables. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 97–106. IEEE Computer Society, 2011.

[HH11] Ichiro Hasuo and Naohiko Hoshino. Semantics of higher-order quantum computation via geometry of interaction. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 237–246. IEEE Computer Society, 2011.

[HO00] Martin Hyland and Luke Ong. On full abstraction for PCF. *Information and Computation*, 163(2):285–408, December 2000.

[HvG03] Dominic Hughes and Rob van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science, LICS 2003*, pages 1–10. IEEE Computer Society Press, 2003.

[Hyl76] J. Martin E. Hyland. A Syntactic Characterization of the Equality in Some Models of the Lambda Calculus. *J. London Math. Soc.*, 2(12):361–370, 1976.

[JP89] Claire Jones and Gordon Plotkin. A probabilistic powerdomains of evaluation. In *Proceedings of the 4th Annual IEEE Symposium on Logic in Computer Science, LICS 1989*. IEEE Computer Society, 1989.

[Laf88] Yves Lafont. *Logiques, catégories et machines.* PhD thesis, Université Paris 7, 1988.

[Laf90] Yves Lafont. Interaction nets. In *POPL '90: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 95–108, New York, NY, USA, 1990. ACM.

[LMMP13] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2013), 25-28 June 2013, New Orleans, USA, Proceedings*, pages 301–310, 2013.

[Mac71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, Berlin, 1971.

[Mal10] O. Malherbe. *Categorical models of computation: partially traced categories and presheaf models of quantum computation.* PhD thesis, University of Ottawa, 2010.

[Man12] Giulio Manzonetto. What is a categorical model of the differential and the resource λ-calculi? *Mathematical Structures in Computer Science*, 22(3):451–520, 2012.

[Mel09] Paul-André Melliès. Categorical semantics of linear logic. *Panoramas et Synthèses*, 27, 2009.

[Mog89] Eugenio Moggi. Computational lambda-calculus and monads. In *LICS*, pages 14–23. IEEE Computer Society, 1989.

[Mon03] Raphaël Montelatici. Polarized proof nets with cycles and fixpoints semantics. In Martin Hofmann, editor, *Typed Lambda Calculi and Applications*, volume 2701 of *Lecture Notes in Computer Science*, pages 256–270. Springer Berlin Heidelberg, 2003.

[MP11] Giulio Manzonetto and Michele Pagani. Bohm's Theorem for Resource Lambda-Calculus through Taylor Expansion. In Luke Ong, editor, *Typed Lambda Calculi and Applications - 10th International Conference (TLCA 2011)*, volume 6690 of *Lecture Notes in Comput. Sci.* Springer, 2011.

[MTT09] Paul-André Melliès, Nicolas Tabareau, and Christine Tasson. An explicit formula for the free exponential modality of linear logic. In *Automata, Languages and Programming*, volume 5556 of *LNCS*, pages 247–260. Springer, 2009.

[Ned73] Robert Pieter Nederpelt. *Strong Normalization for a Typed Lambda Calculus with Lambda Structured Types.* Ph.D. thesis, Technische Hogeschool Eindhoven, 1973.

[Pag06] Michele Pagani. Acyclicity and coherence in multiplicative and exponential linear logic. In *Computer Science Logic*, volume 4207 of *Lecture Notes in Comput. Sci.*, pages 531–545, 2006.

[Pag09] Michele Pagani. The Cut-Elimination Thereom for Differential Nets with Boxes. In Pierre-Louis Curien, editor, *Proceedings of the Ninth International Conference on Typed Lambda Calculi and Applications (TLCA 2009)*, Lecture Notes in Computer Science, pages 219–233. Springer, 2009.

[Pag12] Michele Pagani. Visible acyclic differential nets, part I: Semantics. *Ann. Pure Appl. Logic*, 163(3):238–265, 2012.

[Plo77] Gordon D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):225–255, 1977.

[PRDR10a] Michele Pagani and Simona Ronchi Della Rocca. Linearity, non-determinism and solvability. *Fundamenta Informaticae*, 103(1-4):173–202, 2010.

[PRDR10b] Michele Pagani and Simona Ronchi Della Rocca. Solvability in Resource Lambda-Calculus. In Luke Ong, editor, *FOSSACS*, volume 6014 of *Lecture Notes in Comp. Sci.*, pages 358–373, 2010.

[PSV14] Michele Pagani, Peter Selinger, and Benoit Valiron. Applying Quantitative Semantics to Higher-Order Quantum Computing. In P. Sewell, editor, *The 41th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL14, San Diego, USA*. ACM, 2014.

[PT09a] Michele Pagani and Christine Tasson. The Taylor Expansion Inverse Problem in Linear Logic. In Andrew Pitts, editor, *Proceedings of the Twenty-Fourth Annual IEEE Symposium on Logic in Computer Science (LICS 2009)*, pages 222–231. IEEE Computer Society Press, 2009.

[PT09b] Michele Pagani and Paolo Tranquilli. Parallel Reduction in Resource Lambda-Calculus. In Zhenjiang Hu, editor, *Programming Languages and Systems, 7th Asian Symposium (APLAS 2009)*, volume 5904 of *Lecture Notes in Comput. Sci.*, pages 226–242, 2009.

[PT11] Michele Pagani and Paolo Tranquilli. The conservation theorem for differential nets. preprint, accepted to *Mathematical Structures in Computer Science*, 2011.

[PTdF10] Michele Pagani and Lorenzo Tortora de Falco. Strong Normalization Property for Second Order Linear Logic. *Theoretical Computer Science*, 411(2):410–444, 2010.

[Reg92] Laurent Regnier. *Lambda-Calcul et Réseaux.* Thèse de doctorat, Université Paris VII, 1992.

[Ret97] Christian Retoré. A semantic characterisation of the correctness of a proof net. *Math. Struct. Comput. Sci.*, 7(5):445–452, October 1997.

[SD80] N. Saheb-Djahromi. Cpo's of measures for nondeterminism. *Theor. Comput. Sci.*, 12:19–37, 1980.

[Sel04a] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.

[Sel04b] Peter Selinger. Towards a semantics for higher-order quantum computation. In *QPL'04*, TUCS General Publication No 33, pages 127–143, 2004.

[SV06] Peter Selinger and Benoit Valiron. A lambda calculus for quantum computation with classical control. *Math. Struct. Comput. Sci.*, 16(3):527–552, 2006.

[SV09] Peter Selinger and Benoît Valiron. Quantum lambda calculus. In Simon Gay and Ian Mackie, editors, *Semantic Techniques in Quantum Computation*, chapter 9, pages 135–172. Cambridge University Press, 2009.

[Tak95] Masako Takahashi. Parallel reductions in lambda-calculus. *Information and Computation*, 118(1):120–127, April 1995.

[Tas06] Christine Tasson. Foncteurs analytiques et logique linéaire. Mémoire MPRI, Université Paris VII, 2006.

[Ter03] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2003.

[Tra08] Paolo Tranquilli. A characterization of hypercoherent semantic correctness in multiplicative additive linear logic. In Michael Kaminski and Simone Martini, editors, *CSL*, volume 5213 of *Lecture Notes in Computer Science*, pages 246–261. Springer, 2008.

[Tra09] Paolo Tranquilli. Confluence of pure differential nets with promotion. In Erich GrÃďdel and Reinhard Kahle, editors, *Computer Science Logic*, volume 5771 of *Lecture Notes in Computer Science*, pages 500–514. Springer Berlin Heidelberg, 2009.

[Tra11] Paolo Tranquilli. Intuitionistic differential nets and lambda-calculus. *Theor. Comput. Sci.*, 412(20):1979–1997, April 2011.

[Val08] Benoît Valiron. *Semantics for a higher-order functional programming language for quantum computation.* PhD thesis, University of Ottawa, 2008.

[Wad71] Christopher P. Wadsworth. *Semantics and pragmatics of the lambda-calculus.* Ph.D. thesis, Oxford University, 1971.

[Wad76] Christopher P. Wadsworth. The relation between computational and denotational properties for Scott's D_∞-models of the lambda-calculus. *SIAM J. Comput.*, 5(3):488–521, 1976.

Appendix A

Lafont category

I recall in a nutshell the categorical semantics of propositional linear logic as formulated in Lafont's thesis [Laf88]. This is not the most general definition of a LL model, but it has the advantage of being simple and general enough to encompass the class of models that have been considered in this memoire. My main reference for categorical models of LL is [Mel09], as well as `http://ncatlab.org/`.

The definition of a Lafont category is in the folklore, but it might be not immediate to grasp it from the literature, while all the other definitions are completely standard. The notation that I use however is not completely standard outside the linear logic community, so it is worthwhile to recall it here.

Given a category C and objects A, B we denote by $\mathrm{C}(A, B)$ the corresponding hom-set and by $f, g, h, \ldots$ its elements. We write the identity morphism on A as id^A. Composition is written using infix ; in diagram order.

In a *symmetric monoidal category* (*smc*) C, we denote by $\otimes$ the tensor product and by 1 its unit. When C is monoidal closed (*smcc*), the monoidal exponential object is denoted as $A \multimap B$. We use $\mathrm{eval}^{A,B} \in \mathrm{C}((A \multimap B) \otimes A, B)$ for the monoidal evaluation morphism.When C is moreover $\star$-autonomous with respect to a dualizing object $\bot$, we indicate by $A^\perp$ the dual object $A \multimap \bot$.

In a *cartesian closed category* (*ccc*) C, we denote by $\&$ the product, and we write $\top$ for the terminal object. We use $\mathrm{proj}^1, \mathrm{proj}^2$ for the corresponding projections. In presence of *bi-products*, we denote the bi-product by $\oplus$ and by $\mathrm{inj}^1, \mathrm{inj}^2$ the corresponding injections. The exponential object is denoted by $A \to B$, the evaluation map by $\mathrm{Eval}^{A,B} \in \mathrm{C}((A \to B) \mathbin{\&} A, B)$ and the currying of $f \in \mathrm{C}(A \mathbin{\&} B, C)$ by $\mathrm{Curry}(f) \in \mathrm{C}(A, B \to C)$.

An object A of a smcc C is a *(commutative) comonoid* if it is equipped with a multiplication $\mathrm{contr} \in \mathrm{C}(A, A \otimes A)$ and a unit $\mathrm{weak} \in \mathrm{C}(A, 1)$ satisfying the usual associativity (commutativity) and unit equations. A *comonoid morphism* f from $(A_1, \mathrm{contr}_1, \mathrm{weak}_1)$ to $(A_2, \mathrm{contr}_2, \mathrm{weak}_2)$ is defined as a morphism $f \in \mathrm{C}(A_1, A_2)$ such that $f\,;\,\mathrm{contr}_2 = \mathrm{contr}_1\,;\,(f \otimes f)$ and $f\,;\,\mathrm{weak}_2 = \mathrm{weak}_1$.

Definition 29. A smcc C is a Lafont category whenever:

1. it is cartesian;
2. for every object A, there exists an object $!A$ being the free commutative comonoid generated by A, i.e. $!A$ is endowed with a commutative comonoid structure:

$$\mathrm{contr}^A \in \mathrm{C}(!A, !A \otimes !A), \qquad \mathrm{weak}^A \in \mathrm{C}(!A, 1),$$

and a morphism $\mathrm{der}^A \in \mathrm{C}(!A, A)$ satisfying the following universality property: for every commutative comonoid B and for every morphism $f \in \mathrm{C}(B, A)$ there exists a unique comonoid morphism $f^\dagger \in \mathrm{C}(B, !A)$ satisfying $f^\dagger\,;\,\mathrm{der}^A = f$.

Theorem 30. *A $\star$-autonomous Lafont category yields a model of linear logic.*

A model of linear logic can be transformed into a model of simply typed λ-calculus in the coKleisli category associated with the exponential comonad. Indeed, if C is a Lafont category, the exponential comonad is induced by the freeness of the comonoid on $!A$, as follows:

- the endofunctor $!$ sends every object A into the free commutative comonoid $!A$ and every morphism $f \in \mathrm{C}(A, B)$ into $(\mathrm{der}^A\,;\,f)^\dagger \in \mathrm{C}(!A, !B)$,
- the multiplication is called *digging* and defined as $\mathrm{digg}^A \triangleq (\mathrm{id}^{!A})^\dagger \in \mathrm{C}(!A, !!A)$,
- the unit is the morphism $\mathrm{der}^A \in \mathrm{C}(!A, A)$ given above.

The coKleisli category $\mathrm{C}_!$ over the comonad $(!, \mathrm{digg}, \mathrm{der})$ is defined to have the same objects as C and $\mathrm{C}_!(A, B) \triangleq \mathrm{C}(!A, B)$. Composition in $\mathrm{C}_!$ is defined as $f\,;_!\,g \triangleq \mathrm{digg}\,;\,!f\,;\,g$ and identities $A \triangleq \mathrm{der}^A$.

Theorem 31. *The coKleisli category* $\mathrm{C}_!$ *of a Lafont category* $\mathrm{C}_!$ *is cartesian closed.*

Proof. (Sketch) The functor $!$ is equipped with a monoidal structure turning it into a strong symmetric monoidal functor from the smc $(\mathrm{C}, \otimes)$ to the smc $(\mathrm{C}, \&)$: the corresponding two isomorphisms are given by $\mathrm{m}^\top \triangleq (\top^1)^\dagger \in \mathrm{C}(1, !\top)$ and $\mathrm{m}^{A,B} \triangleq \langle(\mathrm{der}^A \otimes \mathrm{weak}^B), (\mathrm{weak}^A \otimes \mathrm{der}^B)\rangle^\dagger \in \mathrm{C}(!A \otimes !B, !(A \,\&\, B))$. Then, the structure of cartesian smcc of C is lifted to a cartesian closed structure in $\mathrm{C}_!$ by the isomorphisms m. The exponential object $A \to B$

is defined as $!A \multimap B$ and the morphism $\mathrm{Eval}^{A,B} \in \mathrm{C}_!((A \to B) \,\&\, A, B)$ is given by $(\mathrm{m}^{!A \multimap B, A})^{-1} \,;\, (\mathrm{der}^{!A \multimap B} \otimes !A) \,;\, \mathrm{eval}^{!A,B}$. This defines an exponentiation since for every $f \in \mathrm{C}_!(C \,\&\, A, B)$ there is a unique morphism $\mathrm{Curry}(f) \triangleq (\mathrm{m}^{C,A} \,;\, f)^\dagger \in \mathrm{C}_!(C, A \to B)$ satisfying $\mathrm{Curry}(f) \,\&\, A \,;_!\, \mathrm{Eval} = f$. $\square$

A very simple example of $\star$-autonomous (indeed compact closed) Lafont category is the category REL of sets and relations, whose structure is sketched in Figure A.1. Figure A.2 outlines the coKleisli category MREL associated with the multiset comonad.

objects	sets
hom-set	$\mathrm{REL}(A,B) \triangleq \mathcal{P}(A \times B)$
composition	$f\,;\,g \triangleq \{(a,c)\ ;\ \exists b \in B, (a,b) \in f, (b,c) \in g\}$
identity	$\mathrm{id}^A \triangleq \{(a,a)\ ;\ a \in A\}$

(a) objects and morphisms ($f \in \mathrm{REL}(A,B)$, $g \in \mathrm{REL}(B,C)$)

unit, dualizing object	$1 \triangleq \perp \triangleq \{*\}$
$\otimes$ on objects, hom-object	$A \otimes B \triangleq A \multimap B \triangleq A \times B$
$\otimes$ on morphisms	$f \otimes g \triangleq \{((a,b),(c,d))\ ;\ (a,c) \in f, (b,d) \in g\}$
evaluation	$\mathrm{eval}^{A,B} \triangleq \{(((a,b),a),b)\ ;\ a \in A, b \in B\}$

(b) monoidal structure

indexed biproduct	$\bigoplus_{i\in I} A_i \triangleq \bigcup_{i \in I} \{i\} \times A_i$
j-th projection $\bigoplus_{i\in I} A_i \mapsto A_j$	$\mathrm{proj}^j \triangleq \{((j,a),a)\ ;\ a \in A_j\}$
j-th injection $A_j \mapsto \bigoplus_{i\in I} A_i$	$\mathrm{inj}^j \triangleq \{(a,(j,a))\ ;\ a \in A_j\}$

(c) cartesian structure

object	$!A \triangleq \mathcal{M}_{\mathrm{f}}(A)$
comult. $!A \mapsto\ !A \otimes !A$	$\mathrm{contr} \triangleq \{(m_1 + m_2, (m_1, m_2))\ ;\ m_1, m_2 \in\ !A\}$
counit $!A \mapsto 1$	$\mathrm{weak} \triangleq \{([\,], *)\}$
dereliction $!A \mapsto A$	$\mathrm{der} \triangleq \{(m, [a])\ ;\ a \in A\}$

(d) free commutative comonoid

Figure A.1: the $\star$-autonomous Lafont structure of the category REL.

objects	sets
hom-set	$\mathrm{MREL}(A,B) \triangleq \mathcal{P}(\mathcal{M}_{\mathrm{f}}(A) \times B)$
composition	$f\ ;\ _!g \triangleq \left\{ (m,c)\ ;\ \begin{array}{l} \exists [b_1, ..., b_n] \in \mathcal{M}_{\mathrm{f}}(B), \\ ([b_1, ..., b_n], c) \in g, \\ \exists m_1, \ldots, m_n \in \mathcal{M}_{\mathrm{f}}(A), \\ \biguplus_i m_i = m \text{ and } (m_i, b_i) \in f \end{array} \right\}$
identity	$\mathrm{id}^A \triangleq \{([a], a)\ ;\ a \in A\}$

(a) objects and morphisms ($f \in \mathrm{MREL}(A,B), g \in \mathrm{MREL}(B,C)$)

terminal object	$\top \triangleq \emptyset$
cartesian product	$A \,\&\, B \triangleq A \uplus B$
hom-object	$A \to B \triangleq \mathcal{M}_{\mathrm{f}}(A) \times B$
evaluation	$\mathrm{Eval}^{A,B} \triangleq \{((([(m,b)], m), b)\ ;\ m \in \mathcal{M}_{\mathrm{f}}(A), b \in B\}$

(b) cartesian closed structure (in the definition of $\mathrm{Eval}^{A,B}$, we use the linear logic isomorphism $!((A \to B) \,\&\, A) \simeq\ !(A \to B) \otimes !A$)

Figure A.2: the Cartesian closed structure of the category MREL.

Résumé

La recherche sur l'extension différentielle de la logique linéaire et ses sémantiques quantitatives a été particulièrement dynamique ces dernières années. Cette approche est très riche et peut potentiellement être appliquée dans différents domaines de l'informatique théorique. Le but principal de mon travail est de concrétiser cette démarche, en utilisant les sémantiques quantitatives pour étudier de nouvelles primitives de programmation, ou pour comparer les algorithmes non seulement par rapport à ce qu'ils calculent, mais aussi relativement à la manière dont ils calculent: avec combien de ressources, avec quelle probabilité (dans un cadre probabiliste), de combien de façons différentes (dans un cadre non-determiste).

Nous présentons ici les résultats que nous avons récemment obtenus dans trois lignes de recherche:

- étude de l'élimination des coupures des réseaux de preuves et réseaux différentiels;
- étude des lambda-calculs avec ressources bornées;
- sémantiques des langages non-deterministes, probabilistes et quantiques d'ordre supérieur.