Quantum Game Semantics

Marc de Visme

▶ **To cite this version:**

Marc de Visme. Quantum Game Semantics. Logic in Computer Science [cs.LO]. Université de Lyon, 2020. English. NNT : 2020LYSEN056 . tel-03045844

HAL Id: tel-03045844
https://theses.hal.science/tel-03045844

Submitted on 8 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro National de Thèse : 2020LYSEN056

Thèse de Doctorat de l'Université de Lyon
opérée par
l'École Normale Supérieure de Lyon

École Doctorale N° 512
École doctorale en Informatique et Mathématiques de Lyon

Spécialité de doctorat : Informatique
Discipline : Informatique fondamentale

Soutenue publiquement le 28/09/2020, par :

Marc de Visme

Sémantique des Jeux Quantique

Devant le jury composé de :

Selinger, Peter	Professeur	Dalhousie University	Rapporteur
Tasson, Christine	Maître de conférences	Université de Paris	Rapporteure
Danos, Vincent	Directeur de recherche	ENS Paris, CNRS	Examinateur
Faggian, Claudia	Chargée de recherche	Université de Paris, CNRS	Examinatrice
Laurent, Olivier	Directeur de recherche	ENS de Lyon, CNRS	Directeur de thèse
Winskel, Glynn	Professeur	University of Cambridge	Co-directeur international
Clairambault, Pierre	Chargé de recherche	ENS de Lyon, CNRS	Co-encadrant de thèse

Contents

Table of Contents **3**

Résumé – Summary in French **9**
Contexte . . . 9
Contexte . . . 9

Remerciements – Thanks **11**

Introduction **13**
Context . . . 13
State of the Art . . . 14
Challenges . . . 15
Our Approach . . . 16
Contributions . . . 16
Outline . . . 17

I Background and Preliminaries **19**

Overview **21**

1 CBV Semantics for Λ **23**
1.1 Notations . . . 23
1.2 Category Theory . . . 24
1.2.1 Monoidal Product in Categories . . . 26
1.2.2 Various Closed Categories . . . 30
1.2.3 Coproducts . . . 33
1.2.4 Linear Exponential Comonads . . . 35
1.3 The Linear Lambda Calculus (Λ) . . . 37
1.3.1 Overview of the Language . . . 39
1.3.2 Linearity of the Language . . . 42

1.3.3 Typing Derivations . . . 42
1.3.4 Operational Semantics . . . 43
1.4 Freyd Categories as a Categorical Model for Λ . . . 44
1.4.1 Denotational Semantics . . . 44
1.4.2 Freyd Categories as a Model . . . 45
1.4.3 Proof of Soundness and Adequacy . . . 49

2 Introduction to Quantum Computation 53
2.1 Hilbert spaces . . . 53
2.1.1 Complex Numbers . . . 53
2.1.2 Hilbert Spaces . . . 54
2.1.3 Tensor Product . . . 56
2.1.4 Positive Operators . . . 57
2.1.5 The Categories of Hilbert Spaces . . . 59
2.2 Preliminaries on Partial Orders . . . 65
2.2.1 Partial Orders . . . 65
2.2.2 Directed Complete Partial Orders . . . 65
2.2.3 Positive Monoids . . . 66
2.2.4 Positive Cones . . . 68
2.3 Quantum Computation . . . 69
2.3.1 A First Language for Quantum Computation . . . 70
2.3.2 Pure States . . . 72
2.3.3 Mixed States . . . 76

3 Relational Model for LQΛ 79
3.1 The Linear Quantum λ-calculus . . . 79
3.1.1 Quantum Primitives . . . 80
3.1.2 Lists . . . 80
3.1.3 Operational Semantics . . . 81
3.2 The Linear Quantum Relational Model . . . 84
3.2.1 Definition of the Model . . . 84
3.2.2 Examples . . . 88
3.2.3 Soundness and Adequacy for LQΛ . . . 91
3.2.4 Full Abstraction for LQΛ . . . 93

II Quantum Game Semantics 99

Overview 101

4 Preliminaries on Concurrent Game Semantics **103**
4.1 Introduction to Game Semantics . . . 103
4.1.1 Player and Opponent Moves . . . 103
4.1.2 Games . . . 103
4.1.3 Strategies . . . 105
4.2 The Category of Event Structures . . . 106
4.2.1 Event Structures . . . 106
4.2.2 Parallel Composition and Coproduct . . . 110
4.2.3 Interactive Composition . . . 111
4.3 Matching Pairs of Configurations . . . 118
4.3.1 Examples . . . 118
4.3.2 Definition of Matching Pairs . . . 120
4.3.3 Matching Pairs in the Interaction . . . 123
4.3.4 Interactive Composition . . . 126
4.4 Games and Strategies . . . 127
4.4.1 Event Structures with Polarities . . . 128
4.4.2 The Category of Concurrent Games and Strategies . . . 128

5 Quantum Concurrent Games **133**
5.1 Guiding Example . . . 133
5.2 Probabilistic Concurrent Strategies . . . 137
5.2.1 The Guiding Example . . . 137
5.2.2 Defining Probabilistic Strategies . . . 137
5.3 Quantum Concurrent Games . . . 140
5.3.1 Definition of Quantum Games . . . 140
5.3.2 Back to the Guiding Example . . . 142
5.3.3 Definition of Quantum Strategies . . . 144
5.3.4 Properties of the Drop Function . . . 148
5.3.5 The Drop Condition . . . 150
5.3.6 The Category of Quantum Games and Strategies . . . 152
5.3.7 The Polarised Quantum Valuation . . . 153
5.4 Payoff Games and Winning Strategies . . . 155
5.5 The Freyd Category of Quantum Arenas and Strategies . . . 161
5.5.1 The Category **QA** . . . 161
5.5.2 The Premonoidal Tensor . . . 163
5.5.3 The Freyd Closure . . . 168
5.5.4 The Distributive CFC . . . 173

6 Game Semantics for QΛ **175**
6.1 Games Model for LQΛ . . . 175
6.1.1 Semantics for Terms . . . 175
6.1.2 Semantics for Quantum Closures . . . 178
6.1.3 Soundness and Adequacy . . . 179
6.1.4 Examples . . . 182
6.2 An ≡-Adequate Model . . . 186
6.2.1 Visibility and Deadlock-Free Composition . . . 189
6.2.2 The Exhaustive Equivalence ≡ . . . 192
6.2.3 ≡-Adequacy . . . 195
6.3 ≡-Full Abstraction for LQΛ . . . 196
6.3.1 Web of a Type . . . 196
6.3.2 Test Terms and Generator Terms . . . 197
6.3.3 Relational Collapse of LQΛ . . . 200
6.4 Affine Quantum Semantics . . . 201

III Fully Abstract Models for the Full Quantum λ-calculus **203**

Overview **205**

7 The Quantum λ-calculus **207**
7.1 The Quantum λ-calculus . . . 207
7.1.1 Motivation . . . 207
7.1.2 Syntax and Typing System . . . 207
7.1.3 Operational Semantics . . . 212
7.1.4 Convergence and Observational Equivalence . . . 215
7.2 Categorical Pre-Model for the Quantum λ-Calculus . . . 215
7.2.1 The Categorical Pre-Model . . . 216
7.2.2 Term Invariance for the Categorical Model . . . 217

8 Relational Semantics for QΛ! **223**
8.1 Modelling Replication . . . 223
8.1.1 Quantum Relations on Arenas instead of Webs . . . 223
8.1.2 Arenas and the Notion of Symmetry . . . 224
8.1.3 Quantum Valuations and Symmetry . . . 227
8.2 Event Structures with Symmetry . . . 227
8.2.1 The Category of Event Structures with Symmetry . . . 227
8.2.2 Games with Symmetry . . . 229
8.2.3 The Linear Exponential Comonad ! . . . 231
8.3 Quantum Games and Arenas with Symmetry . . . 233

8.3.1 Quantum Payoff Games with Symmetry . . . 233
8.3.2 Quantum Arenas with Symmetry . . . 235
8.4 Quantum Relations on Arenas with Symmetry . . . 236
8.4.1 Example . . . 237
8.4.2 The Category $\sim$-**QARel** . . . 239
8.4.3 A Sound and Adequate Semantics for LQ$\Lambda_!$ . . . 243
8.4.4 Affine Case . . . 245

9 Game Semantics for Q$\Lambda_!$ 247
9.1 Polarisation of the Symmetry . . . 247
9.1.1 Uniformity of Strategies . . . 247
9.1.2 The Map(-,-) Example . . . 247
9.1.3 Another Example . . . 249
9.1.4 Polarisation of Symmetry . . . 249
9.1.5 Miscellaneous Lemmas on Games with Symmetry . . . 253
9.2 Strategies with Symmetry . . . 254
9.2.1 Interactive Composition with Symmetry . . . 254
9.2.2 Strategies with Symmetry . . . 257
9.3 Quantum Strategies with Symmetry . . . 260
9.3.1 Example . . . 260
9.3.2 Quantum Strategies with Symmetry . . . 260
9.3.3 Categorical Model for LQ$\Lambda_!$ . . . 265
9.3.4 Pre-Model for AQ$\Lambda_!$ . . . 269
9.4 The Exhaustive Equivalence . . . 269
9.4.1 Motivation . . . 269
9.4.2 Configurations with Symmetry . . . 272
9.4.3 Witnesses up to Symmetry . . . 273
9.4.4 Quantum Valuations and Symmetry . . . 275
9.4.5 Exhaustive Equivalence of Strategies with Symmetry . . . 280
9.4.6 Collapse of Strategies into Relations on Arenas . . . 281
9.5 A Sound and $\equiv$-Adequate Game Model . . . 281

10 Full Abstraction for Q$\Lambda_!$ 285
10.1 Adding Formal Parameters . . . 285
10.1.1 Motivation . . . 285
10.1.2 Q$\Lambda_!$ with Formal Parameters . . . 286
10.1.3 Formal Power Series . . . 286
10.2 Parametrised Game Semantics . . . 288
10.2.1 Parametrised Quantum Strategy with Symmetry . . . 288
10.2.2 The Exhaustive Equivalence . . . 291
10.2.3 The Skeleton of a Strategy . . . 293

10.3 Full Abstraction . . . 293
10.3.1 Extended Configurations of an Arena . . . 293
10.3.2 Parametrised Test and Generator Terms . . . 294
10.3.3 Example . . . 294
10.3.4 Properties of Test and Generator Terms . . . 298
10.3.5 Full Abstraction for $\mathrm{LQ}\Lambda_!$ . . . 301
10.3.6 Affine Case . . . 302
10.3.7 Full Abstraction of the Relational Model . . . 303

Conclusion and Perspectives **305**

IV Appendix **307**

A Miscellaneous Lemmas **309**
A.1 Interaction and Interactive Composition . . . 309
A.2 $\oplus$-Covered Configurations . . . 312
A.3 Test Strategies . . . 315

B Postponed Proofs **317**
B.1 Parallel Tensor and Semi-Bifunctoriality . . . 317
B.1.1 Definitions and Objective . . . 317
B.1.2 Initialised Interactive Composition . . . 318
B.1.3 Proof of Semi-Bifunctoriality . . . 319
B.2 Visibility and Deadlock-Free Composition . . . 323
B.2.1 Definitions . . . 323
B.2.2 Category of Visible Strategies . . . 324
B.2.3 Deadlock-Free Composition . . . 326

Bibliography **331**

Remerciements – Thanks

Je suis infiniment reconnaissant à tous ceux et celles qui m'ont accompagné durant ces trois magnifiques années de thèse. J'ai pu m'épanouir dans un milieu bienveillant et intellectuellement stimulant, dans lequel je me sentais considéré comme un chercheur à part entière.

Merci tout particulièrement à Pierre pour son encadrement, ses conseils, et son investissement dans mon travail de recherche et dans toutes les étapes de ma thèse. C'est en grande partie grâce à son expérience, ses critiques détaillées, et les longues discussions que nous avons eues, que je peux vous présenter un manuscrit de thèse dont je suis fier. Special thanks to Glynn, which was with me at the very beginning of this long story, six years ago, and has continued to guide me since then. I've always felt listened as a (young) researcher rather than as a regular student. Merci aussi à Olivier, qui était là pour me conseiller et répondre à mes questions, qu'elles soient scientifiques ou non.

Merci à Benoît et Ugo, tout particulièrement pour les discussions scientifiques que nous avons eues lors des mois qui précédaient ma thèse. Elles m'ont aidé à la commencer sur de bonnes bases. Many thanks to the members of my jury: Vincent Danos, Claudia Faggian, Peter Selinger, and Christine Tasson for being available despite the very unique circumstances we are currently living through, and for their very interesting questions. Special thanks to Peter and Christine, who read this long manuscript in details and came up with many insightful questions, advices and corrections.

Merci à Aurore, Hugo, Lison et Simon, qui ont su me faire part de leur expérience et leur soutient lorsque j'en avais besoin. Plus généralement, merci à l'équipe Plume pour toute sa bienveillance et toutes les discussions que j'ai pu apprécier. Merci notamment à Denis, pour les nombreuses discussions scientifiques, ludiques, et combinaisons des deux que nous avons eus. And thanks to Christian for the multiple boardgaming sessions we had together. Merci à Adrien, Florent, Laureline, Pierre et Valentin, avec qui j'ai passé de nombreuses heures à préparer des TDs. Merci à Catherine et Marie, sans qui j'aurais eu beaucoup plus de problèmes administratifs.

En dehors de Plume, merci à Laure, Ludovic, Matthieu et Nicolas, avec qui j'ai apprécié toutes les discussions et grâce à qui j'ai pu préparer et encadrer les séances de TPs les plus enrichissantes pour moi pédagogiquement. Et merci à Nicolas pour nos discussions sur les graphes parfaits. Et merci plus généralement à tous les autres personnes avec qui j'ai pu

interagir au sein du LIP.

Merci au club des Improfesseurs pour les séances d'improvisation qui m'ont permis de m'aérer l'esprit de manière hebdomadaire, et pour toutes ces improvisations inoubliables que nous avons faite ensemble.

Merci à mes amis de l'ENS, avec qui j'ai étudié, vécu, et joué pendant quatre ans. Merci tout particulièrement à François, Thomas, Vincent, Thomas, Sarah, Luc, et Thomas, pour ces jeux réguliers durant ces trois dernières années.

Merci à ma famille, à mes parents qui ont toujours été là pour moi et m'ont aidé et soutenu à toutes les étapes, à mes frères à qui je souhaite le meilleur, et à mes grands-parents, dont mes souvenirs les plus anciens sont des souvenirs d'amour et de bienveillance. Merci à Jean-Philippe, Séverine et leurs enfants, chez qui m'ont toujours accueilli chez eux comme si c'était un deuxième "chez moi".

Merci à toutes les personnes avec qui j'ai passé mes vacances, que ce soit famille ou ami·e·s. Ces moments inoubliables m'ont chargé de l'énergie nécessaire pour le reste de l'année.

Résumé – Summary in French

Contexte

Cette thèse s'inscrit dans le domaine de la *sémantique dénotationnelle*, qui est l'étude et la conception de représentations mathématiques des langages de programmation afin d'en extraire leur sens et leurs propriétés. Un des sujets d'étude principaux de la sémantique dénotationnelle est la notion d'*équivalence observationnelle*, qui est la garantie que deux programmes ont le même comportement quel que soit le contexte, et peuvent donc être utilisés indifféremment l'un de l'autre.

Le thème de cette thèse est la *sémantique des langages de programmation quantique* et en particulier la conception de modèles interactifs (jeux) pour ces langages. De plus en plus, les chercheurs en sémantique dénotationnelle cherchent à enrichir les modèles avec des aspects quantitatifs afin de représenter par exemple des effets probabilistes ou quantiques. Les programmes du premier ordre contenant ces effets quantitatifs sont en général bien compris et disposent de modèles à base d'outils mathématiques traditionnels. Par exemple, pour le quantique, l'un des modèles standard utilise les espaces de Hilbert et les matrices de densité. Néanmoins, les programmes qui disposent de flot de contrôle plus complexe, par exemple en présence d'ordre supérieur, sont significativement plus difficiles à modéliser, car mixer les modèles de sémantique dénotationnelle traditionnelle (domaines, fonctions continues, etc) et les effets quantitatifs s'avère délicat. Cette thèse explore l'utilisation des modèles de sémantique dénotationnelle interactive, plus précisément la *sémantique des jeux concurrents*, qui permettent une étude plus précise du flot de contrôle, facilitant l'ajout d'effets quantitatifs tels que le quantique.

Contenu de la Thèse

Le manuscrit de thèse se décompose en trois parties. Au cœur de la première partie sont les *préliminaires quantiques*, qui introduisent les notions de calcul quantique utilisées, deux manières de représenter ces calculs quantiques (états purs et états mixtes), et une reformulation d'un modèle dénotationnel déjà existant (de Selinger et Valiron) pour le

fragment linéaire du λ-calcul quantique. L'une des propriétés clé des effets quantiques et de leurs modèles est la notion de linéarité : les états quantiques sont détruits après utilisation, et ne peuvent pas être dupliqués. Afin de pouvoir mieux modéliser cette notion de linéarité, cette première partie commence par des préliminaires catégoriques généralisant la notion de catégorie de Freyd fermée dans un contexte plus adapté à la modélisation des langages linéaires.

La deuxième partie présente la première contribution de cette thèse, un *modèle de jeux quantiques* linéaires. Cette partie commence par des préliminaires posant les bases techniques des jeux concurrents (de Rideau, Winskel), puis y ajoute la notion de donnée quantique. Enfin, ces outils sont utilisés pour former un modèle de jeux pour le fragment linéaire du λ-calcul quantique, pour lequel on montre la *pleine adéquation*, l'une des propriétés les plus fortes que peut respecter un modèle dénotationnel, qui caractérise la notion d'équivalence observationnelle mentionnée précédemment.

Lors de ces deux premières parties, nous nous étions restreints au fragment linéaire du λ-calcul quantique, dans lequel les fonctions et variables ne peuvent être utilisées qu'une seule et unique fois. Dans le λ-calcul quantique complet, on autorise les fonctions qui ne consomment pas de ressources quantiques à être dupliquées et réutilisées. Cela introduit des complexités techniques significatives, mais pour la plupart déjà étudiées dans des cas plus simples que la sémantique des langages de programmation quantique. Dans la troisième partie, nous étendons toutes les notions des deux premières parties afin de pouvoir modéliser le λ-calcul quantique dans toute sa généralité. Nous démontrons que le modèle de jeux ainsi défini est pleinement adéquat. C'est une des contributions principales de cette thèse : aucun modèle n'était connu pleinement adéquat pour le λ-calcul quantique complet. Finalement, nous construisons un pont fonctoriel avec le modèle antérieur de Pagani, Selinger et Valiron, et déduisons que celui-ci était déjà pleinement adéquat.

Introduction

Context

As quantum programming has been leaving the realm of fiction and preparing its entrance in the real world, it promises some significant impact on computing and its theory. Traditional views on algorithms and complexity are challenged, some of the most well known examples being polynomial time integer factorisation [Sho97], quick data-base search [Gro96], or quantum cryptography [GRTZ02]. Understanding the computational power that can be extracted from the unique properties of quantum information is still an active research domain, *e.g.*, the reserach on the fundamental mechanisms behind locality and contextuality [AB11, ABK+15].

As potential uses of quantum mechanics in computing continue to be discovered, many programming languages including quantum features have been developed, QCL [Öm05] is one of the first quantum programming languages to be implemented, and is a C-like imperative programming language allowing its user to define variables of quantum data type and apply any of the standard quantum operations to it. On the other side of the spectrum, Quipper [GLR+13], is one of the most recent functional quantum programming language, and is an Haskell-like language allowing its user to create, manipulate and execute quantum circuits. As quantum hardware has yet to become mainstream, those languages come with built-in systems that allow the programmers to simulate the execution of the quantum program, making quantum programming a field pre-dating quantum hardware.

In order to study quantum programming, similarly to how we study other kinds of programming, we rely on paradigmatic languages: a paradigmatic language is a programming language which has not been designed to be actually used by programmers; it is a minimalistic language that has been designed to isolate specific computational features and study the interactions between them. A well known example of a paradigmatic language is the λ-calculus: it is a programming language which only features functions, function calls, and variables. Variants of the λ-calculus are used to study the interaction between higher-order computation and various other features, like memory [Sum09], probabilistic branching [ETP14], quantum computation [SV06], and many more.

One of the first steps in studying a language is to define its *operational semantics*. Operational semantics are direct descriptions of the behaviour of a program. They are

often presented as rewriting systems, *e.g* "replace $12 + 30$ by 42" or "replace the function call $f(7)$ by the body of the function $f(x)$ with every instance of the input x substituted by 7". Their down-to-earth approach make operational semantics usually easy to define but hard to analyse.

The natural counterpart of operational semantics is *denotational semantics*. A denotational semantics of a language is a mathematical abstraction which tries to extract the underlying meaning of programs, *e.g* "the function that doubles its input" or "the function that returns 1 half of the time, 2 a quarter of the time, 3 an eight of the time, ...". Since a denotational semantics is an abstraction, a central question is to determine how faithful the abstraction is to the language. The degree of faithfulness describes what kind of properties proven on the semantics will be able to be lifted to the language itself. The golden standard is *full abstraction* [Mil77], which ensures an exact correspondence between the denotational semantics and the program behaviours within the language. More precisely, a denotational semantics is said to be fully abstract when for every two terms of the language, those two terms have the same observational behaviour (*i.e.*, in every context, those two terms are indistinguishable) if and only if they have exactly the same denotational semantics.

This thesis focuses on two particular kinds of denotational semantics: *relational semantics* [Ehr12] and *game semantics* [HO00, AJM00]. The former describes a program by a relation between possible inputs and possible outputs, *e.g.*, the program that negates its boolean input would be represented by $\{(\text{false}, \text{true}), (\text{true}, \text{false})\}$. Game semantics is part of the family of *dynamic* denotational models, as opposed to *static* denotational models like relational semantics. In game semantics, a program is represented by the opposition between a Player, standing for the program itself, and an Opponent, standing for the user or the environment of the program. The exchange of information between the program and the user is represented by moves of a game, Opponent moves corresponding to inputs to the program, and Player moves corresponding to outputs. In fact, game semantics can be understood as a relational semantics in which we remember the order and possible interleaving of inputs and outputs of the programs. This makes game semantics a much more intensional model: while it is not as down-to-earth as the operational semantics, it is still possible to observe and study the order of evaluation of a program within its game semantics.

State of the Art

The quantum λ-calculus was introduced in [SV06] as a paradigmatic language for quantum programming. Like Quipper, it follows the paradigm of quantum data over classical control flow. It is a λ-calculus together with some quantum primitives allowing us to generate, act on, and observe quantum data. In [SV06], Selinger and Valiron provide an operational semantics on tuples of a term of the language together with a quantum store representing the quantum information (like entanglement) accumulated along the computation. This

quantum information is represented using vectors in a Hilbert space.

Because of the different challenges of modelling quantum computation, works on denotational semantics first focussed on a fragment of the quantum λ-calculus, called the linear quantum λ-calculus. Following a line of work from Selinger [Sel04, Sel07], a fully abstract denotational model for the linear quantum λ-calculus was achieved in [SV08]. This model is a relational model weighted by *completely positive maps* representing quantum operations. This model was later [PSV14] extended into a model for the (full) quantum λ-calculus, but the proof of full abstraction in the linear case did not generalise to the general case, leaving the question of full abstraction for the quantum λ-calculus open.

This was not the only attempt at giving a denotational semantics for quantum programming. Delbecque [Del11] made a game semantics model for a fragment of the language with a significant restriction on quantum data, in particular making it impossible for quantum operations to act on quantum data coming from different parts of the program. Those restrictions ensure that the quantum information remains local, and no "long distance" entanglement occurs between different parts of the program. More recently, Hasuo and Hoshino proposed a model, based on Girard's geometry of interaction [Gir89], of a language with a similar restriction as in Delbecque's language [HH17]. But Pagani, Selinger and Valiron were not the only ones to manage to avoid this restriction on entanglement, as Malherbe [MSS13] also presented a model for the quantum λ-calculus (though without recursion, which was present in [PSV14]), using a model based on presheaves, and Dal Lago, Faggian, Valiron, and Yoshimizu recently presented a geometry of interaction model [LFVY17] for the full quantum λ-calculus (with recursion).

Challenges

- Because of entanglement, quantum data is fundamentally non-local, meaning that the state of a quantum bit might be correlated with the state of other quantum bits in the program. While this non-locality is very restricted[1], multiple previous models have encountered difficulties when trying to model it in its whole generality [Del11, HH17].

- Another central property of quantum mechanics is given by the no-cloning theorem: quantum data cannot be duplicated. We say that quantum data is *linear*. This linearity forces the presence in the quantum λ-calculus of two kinds of functions: linear functions, that consume some quantum data hence can be used only once, and non-linear functions, that do not consume any quantum data and can be used as many times as one wants. Any denotational model for the quantum λ-calculus should be able to represent both linear and non-linear behaviours.

[1] And in particular does not allow faster-than-light communication [GRW80].

- The model presented in [PSV14] contains a lot of "junk", *i.e.*, elements that are absurd, such that infinite probabilities, or completely positive maps that correspond to no physically realisable quantum operation. While we did not aim for a definability result, which would be "every element of the denotational model can be realised by a term of the quantum λ-calculus", we wanted to build a model that fits more tightly within physically realisable quantum computation.

Our Approach

Our model uses the recently developed framework of concurrent game semantics [AM99, RW11, CCRW17, CCW19]. One of the motivations behind this choice is the privileged relationship between concurrent game semantics and the relational model. Indeed, given that game semantics is essentially a denotational semantics in which we remember some temporal information, one might wonder whether we recover a static denotational semantics when taking a game semantics and "forgetting" the temporal information. This is unfortunately not always the case, as such forgetting operations are often not functorial due to issues related to deadlocks, as pointed out in [BDER97], or not semantics-preserving [CM10]. However, under some reasonable assumptions, "forgetting" the temporal information of a concurrent game semantics model will exactly give the relational model. This collapse has been proven in the case of probabilistic game semantics of probabilistic PCF in [CCPW18], and we prove it in the case of quantum game semantics.

Our model follows the same paradigm as the quantum λ-calculus: "quantum data, classical control flow". Indeed, our games and strategies will be classical games and classical strategies together with some annotation describing quantum effects. This approach allows us to expect other works made on concurrent game semantics to smoothly extend to the quantum case, *e.g.*, semantics of concurrent languages or with state.

Contributions

- The first contribution of this thesis is a minor one: we provide *a categorical model for the Call-by-Value linear lambda calculus*, adapting the notion of Freyd category [PT97, PT99a] to the monoidal case. Freyd categories rely on premonoidal categories, which distinguish the left-then-right evaluation order from the right-then-left evaluation order. This level of generality is necessary as game semantics does also observe the difference in the evaluation order, hence would not fit more restricted notions of categorical models.

- We extend concurrent games and strategies defined in [CCRW17] to the quantum case, by defining a notion of quantum strategy consisting of event structures [NPW79] annotated by completely positive maps. We then make this notion compatible with

the notion of symmetry on concurrent games [CCW19], used to represent non-linear behaviours. Moreover, concurrent game semantics à la [CCW19] had only been used in the context of call-by-name languages [CCPW18, CCW15a, CCW17, CCHW18], so we adapted some of the definitions to better fit a call-by-value context.

- We build a fully abstract game model for the linear quantum λ-calculus. While [SV08] already provide a fully abstract model for this language, we believe that the more intensional approach of game semantics would ease the extension to more effectful language features.
- Extending our game model from the linear case, we build the first proven fully abstract model for the (full) quantum λ-calculus. The proof of full abstraction relies on a proof method from [ETP14].
- Relying on the privileged relationship between concurrent game semantics and relational semantics, we build a semantics-preserving functor from our game semantics model to a variant of the relational model of [PSV14]. This functor allows us to deduce the full abstraction of the model of [PSV14] from the full abstraction of our game model. We believe that our proof of full abstraction could be directly applied to their model.
- Most previous models of the quantum λ-calculus took a "strictly" linear approach, meaning that functions could be either non-linear, *i.e.*, usable as many time as one want, or linear, *i.e.*, usable *exactly once*. The alternative is the *affine* approach, where functions can either be non-linear, or affine, *i.e.*, usable *at most once*. For all of our models and results, we provide an affine alternative.

Outline

The thesis is organised in three parts. As this thesis bridges multiple domains, from quantum computation to concurrent game semantics, passing by category theory and relational semantics, a lot of pages are reserved to explaining the basics of each of those domains. The first part goes over multiple preliminaries, and contains only minor contributions. The second part develops our model of quantum game semantics in a simpler context than the full quantum λ-calculus: the linear quantum λ-calculus, which has no !, *i.e.*, all variables and function are expected to be used exactly once. The third and last part enriches the model of the second part into a fully abstract model for the full quantum λ-calculus.

Part I The first part starts with a chapter of preliminaries on category theory. After some standard definitions, we present the first minor contribution of this thesis: an adaptation of the notion of Freyd category in a model for the linear call-by-value λ-calculus. In the second chapter, we present the basic mathematics behind Hilbert

spaces and completely positive maps, which will be used in the third chapter to represent quantum computation. In this third chapter, we present the linear quantum λ-calculus. We conclude by defining a variant of the fully abstract model of [SV08] for this language.

Part II In the second part, we start in chapter four by introducing the basics of concurrent game semantics: the partial-order based object called *event structure*, the notion of game and strategy, and the notions of composition of strategies, not yet equipped to deal with symmetry. Then, in chapter five, we present the first major contribution of this thesis: the category of quantum games and quantum strategies. At last, in chapter six, we use those quantum strategies to give a fully abstract game semantics model to the linear quantum λ-calculus, and relate this model to the quantum relational model defined in the first part. We conclude by explaining how to adapt this model to the affine quantum λ-calculus, where the condition "variables and functions must be used exactly once" is relaxed to "variables and functions must be used at most once".

Part III In the third part, we start in chapter seven by presenting the full quantum λ-calculus, and some of its properties. In chapter eight, we extend the quantum relational model of the first part into a model for the full quantum λ-calculus, similarly to the model of [PSV14]. In chapter nine, we do the same for the game model of the second part, and finally in chapter ten we prove that both models are fully abstract, and related by a collapse functor.

Part I

Background and Preliminaries

Overview of Part I

In Parts II and III, we build a game semantics model for a quantum λ-calculus $\mathrm{Q}\Lambda_!$. As such, we first need to introduce each of those notions separately before combining them together.

In the first chapter of this part, we go through some preliminaries on category theory. We then present a categorical model for Λ, a linear call-by-value λ-calculus. This model relies on linear Freyd categories, a generalisation of Freyd categories as defined in [PT99b], and will serve as a template for models of more complex λ-calculi we will study later in this thesis. We note that while the construction is not particularly original, to the best of our knowledge this model does not appear in the literature, and as such is the first contribution of this thesis.

In the second chapter, we present some preliminaries on the mathematics of quantum computation, including Hilbert spaces, hermitian matrices, and infinitary completion of partial orders. We then use them to represent quantum computation. This chapter does not contain any original contribution.

In the third chapter, we define the language $\mathrm{Q}\Lambda$, the linear fragment of $\mathrm{Q}\Lambda_!$. We will restrict ourselves to this fragment until Part III. We present a model for $\mathrm{Q}\Lambda$, which we call the linear quantum relational model, for its similarity with the relational model. This model is a reformulation of the model of Selinger and Valiron in [SV08], made to ease the relationship with game semantics in Section 6.3.3. We provide a proof of full abstraction of this model, which differs from the one of Selinger and Valiron as we aim to set up some definitions and methods useful for later proofs.

Chapter 1

Call-by-Value Semantics for the Linear λ-calculus

1.1 Notations

We start by recalling some basic notations about sets. We will have a use for three different kinds of union of sets:

- $A \cup B$ is the standard set-theoretic union.
- $A \sqcup B$ is $A \cup B$, where we additionally know that $A \cap B = \varnothing$.
- $A \uplus B := \{(0, a) \mid a \in A\} \sqcup \{(1, b) \mid b \in B\}$ is the tagged disjoint union.

We write $\mathbb{N}, \mathbb{Z}, \mathbb{R}_{\geq 0}, \mathbb{R}, \mathbb{C}$ for the sets of natural numbers (including zero), integers, non-negative real numbers, real numbers and complex numbers. We recall some standard operations on complex numbers in Section 2.1.1.

We also write $\overline{\mathbb{R}_{\geq 0}}$ for the set $\mathbb{R}_{\geq 0} \sqcup \{+\infty\}$. When needed, we take the conventions $0 \times \infty = 0 = \infty \times 0$, and carefully restrict ourselves to operations that are compatible with this convention.

Lastly, when considering tuples $a = (a_1, \ldots, a_n) \in A^n$ and $a' = (a_{n+1}, \ldots, a_{n+k}) \in A^k$, we will often make implicit the isomorphism between $A^n \times A^k$ and A^{n+k}, writing $(a, a') = (a_1, \ldots, a_{n+k}) \in A^{n+k}$. Similarly when using monoidal categories, we will often implicitly use associativity isomorphisms. Those omissions are purely for syntactic convenience. We keep the uses of unit isomorphisms $A \times \{\star\} \cong A$ (and its equivalent in monoidal categories) and commutativity isomorphisms $A \times B \cong B \times A$ (and its equivalent in monoidal categories) explicit, except when specified otherwise, *e.g.*, quantum annotations in Section 5.3.1.

1.2 Category Theory

In game semantics, and more generally in denotational semantics, category theory is central to the representation of types and programs: types are represented by objects, programs by morphisms and features of the studied programming language translate to structure of the category; *e.g.*, pairing is represented by a monoidal product, functions are represented through monoidal closures, booleans rely on coproducts, and replication relies on linear exponential comonads [Mel09]. This section will focus on the following notions :

- Symmetric monoidal categories (SMC), symmetric premonoidal categories (SPC), and symmetric linear Freyd categories (SFC)
- Symmetric monoidal closed category (SMCC), compact closed categories (CpCC), $\star$-automonous categories and closed linear Freyd categories (CFC)
- Coproducts and cocartesian categories
- Comonads and Linear Exponential Comonads

All but SFCs and CFCs are standard notions well studied in the literature, as in [PEO15, Mel09, ML98], and SFCs and CFCs are simple generalisations of existing notions to non-cartesian contexts.

We recall some notations: for $\mathcal{C}$ a category, for $A, B \in \mathcal{C}$ two objects, we write $\mathcal{C}(A, B)$ the set of morphisms (also called maps) of $\mathcal{C}$ from A to B. We also write $\mathbf{id}_A \in \mathcal{C}(A, A)$ for the identity on A. We say that $f \in \mathcal{C}(A, B)$ is an isomorphism if there exists $g \in \mathcal{C}(B, A)$ such that $f \circ g = \mathbf{id}_B$ and $g \circ f = \mathbf{id}_A$. When such an isomorphism exits, we say that A and B are isomorphic, and we write $A \cong B$. We refer to [ML98] for more background on category theory.

Example 1.1 Category of Relations **Rel**

As a recurring example, we will consider the category **Rel** of relations. Its objects are finite and countably infinite sets. A map $R \in \mathbf{Rel}(A, B)$ is a subset of $A \times B$, the cartesian product of A and B. The identity map on A is $\mathbf{id}_A^{\mathbf{Rel}} = \{(a, a) \mid a \in A\}$. The composition of $R \in \mathbf{Rel}(A, B)$ and $R' \in \mathbf{Rel}(B, C)$ is simply:

$$(a, c) \in R' \circ R \iff \exists b \in B, (b, c) \in R' \text{ and } (a, b) \in R$$

We distinguish one singleton object $\mathbf{1} = \{\star\}$. We define the booleans with $\mathbf{ff} = (0, \star)$, $\mathbf{tt} = (1, \star)$, and $\mathbf{bit} = \mathbf{1} \uplus \mathbf{1} = \{\mathbf{ff}, \mathbf{tt}\}$.
With those notations, it can be useful to think of elements of $\mathbf{Rel}(\mathbf{bit}, \mathbf{bit})$ as non-deterministic programs from booleans to booleans. So $\{(\mathbf{tt}, \mathbf{ff}), (\mathbf{tt}, \mathbf{tt})\}$ would be the program only terminating on the input $\mathbf{tt}$, which non-deterministically outputs $\mathbf{tt}$ or $\mathbf{ff}$ in that case. In fact, in Section 1.4.2, we use **Rel** as a model of the call-by-value linear λ-calculus Lλ.

Example 1.2 Category of Weighted Relations **WRel**

Another recurring example will be the category **WRel** of weighted relations. Similarly to **Rel**, its objects are finite and countably infinite sets. A map $w \in \mathbf{WRel}(A, B)$ is a function from $A \times B$ to $\overline{\mathbb{R}_{\geq 0}}$. The identity map on A is $\mathbf{id}_A^{\mathbf{WRel}} = (a, b) \mapsto 1$ if $a = b$ and 0 otherwise. The composition $w \in \mathbf{WRel}(A, B)$ and $w' \in \mathbf{WRel}(B, C)$ is:

$$(w' \circ w)(a, c) = \sum_{b \in B} w'(b, c) \cdot w(a, b)$$

We note that infinite sums of elements of $\overline{\mathbb{R}_{\geq 0}}$ are always well defined. Similarly to the case of **Rel**, $\mathbf{WRel}(\mathbf{bit}, \mathbf{bit})$ should be thought of as containing the representation of probabilistic programs from booleans to booleans. So $(a, b) \mapsto 0.5$ if $a = \mathbf{tt}$ and 0 otherwise would be the program only terminating on the input $\mathbf{tt}$, flipping a fair coin to determine the output. We note that **WRel** also contains the representation of programs using absurd probabilities like 2 or even ∞.

1.2.1 Monoidal Product in Categories

Symmetric Monoidal Categories

A symmetric monoidal category is a category with a well-behaving notion of "pairing" of objects and morphisms, but less restrictive than a category with finite products. More formally:

Definition 1.2.1. *A symmetric monoidal category, or SMC, $(\mathcal{C}, \otimes, \mathbf{1})$ is a category $\mathcal{C}$ together with a distinguished object $\mathbf{1}$, a bifunctor $\otimes$ on $\mathcal{C}$, and the following natural isomorphisms:*

- *The left-unitor on A written* $\mathrm{lu}_A \in \mathcal{C}(\mathbf{1} \otimes A, A)$
- *The right-unitor on A written* $\mathrm{ru}_A \in \mathcal{C}(A \otimes \mathbf{1}, A)$
- *The associator on A, B, C written* $\mathrm{as}_{A,B,C} \in \mathcal{C}((A \otimes B) \otimes C, A \otimes (B \otimes C))$
- *The braiding on A, B written* $\mathrm{br}_{A,B} \in \mathcal{C}(A \otimes B, B \otimes A)$

Those isomorphisms are expected to make the coherence diagrams of Fig. 1.1 commute.

In a such an SMC, for $A_1, \ldots, A_n$ some objects, we can define $\bigotimes_{1 \leq i \leq n} A_i := ((A_1 \otimes A_2) \otimes \ldots) \otimes A_n$. The associator, the braiding and the unitors, ensure that any other bracketing of the $\otimes$, any addition of removal of $\mathbf{1}$ objects, and any other ordering of the A_i, would give rise to an isomorphic object. For $k \leq n$ the number of non-$\mathbf{1}$ objects, this isomorphism induces a permutation over $\{1 \ldots k\}$. The coherence diagrams ensure that two isomorphisms obtained from the associator, the braiding, and the unitors, inducing the same permutation are necessarily equal. See [ML98] for a proof of this statement.

We will keep the uses of the unitors and the braiding explicit, however, for syntactical convenience, we will often keep the uses of the associator implicit. In particular, we will usually omit the isomorphism between $\bigotimes_{1 \leq i \leq n} A_i \otimes \bigotimes_{n+1 \leq i \leq n+m} A_i$ and $\bigotimes_{1 \leq i \leq n+m} A_i$.

Example 1.3 Symmetric Monoidal Categories

Both $(\mathbf{Rel}, \otimes, \mathbf{1})$ and $(\mathbf{WRel}, \otimes, \mathbf{1})$ are SMCs, with:

$$\begin{gathered} \mathbf{1} := \{\star\} \\ A \otimes B := A \times B \\ R \otimes R' := \{((a, a'), (b, b')) \mid (a, b) \in R \text{ and } (a', b') \in R'\} \\ w \otimes w' := ((a, a'), (b, b')) \mapsto w(a, b) \cdot w'(a', b') \end{gathered}$$

Morphisms of $\mathbf{Rel}(\mathbf{bit} \otimes \mathbf{bit}, \mathbf{bit})$ or $\mathbf{WRel}(\mathbf{bit} \otimes \mathbf{bit}, \mathbf{bit})$ can be thought of as containing the representation of non-deterministic or probabilistic programs taking two booleans as input, and having one boolean as output.

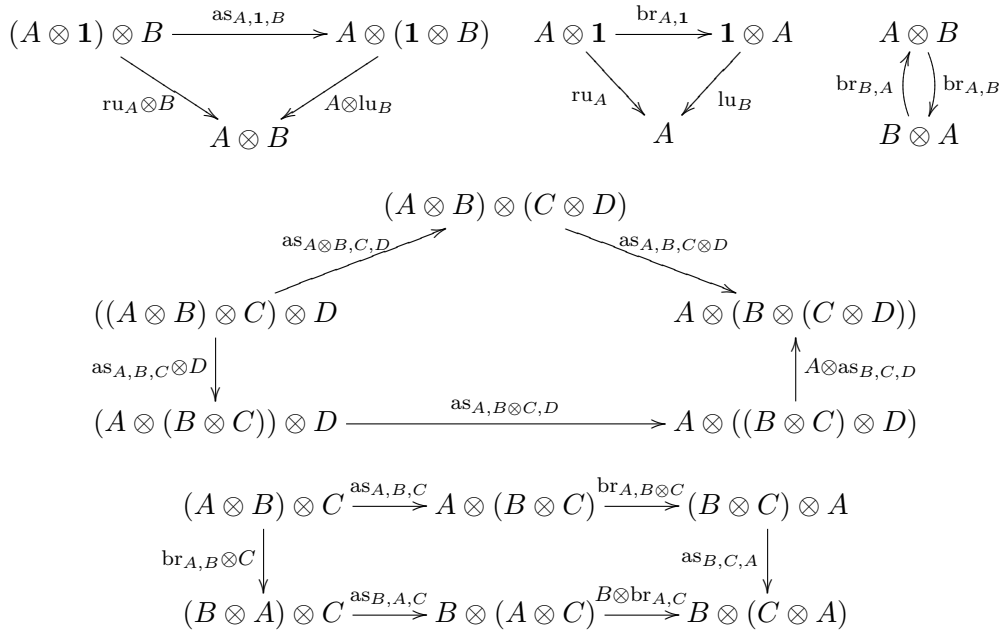


Figure 1.1: Coherence Diagrams for SMCs

When considering functors between SMCs, we will often require them to respect the structure of SMCs. For $(\mathcal{C}, \otimes, \mathbf{1})$ and $(\mathcal{D}, \bullet, \mathbf{e})$ two SMCs, we say that a functor F from $\mathcal{C}$ to $\mathcal{D}$ is a *lax* symmetric monoidal functor if for every objects $A, B \in \mathcal{C}$, there is a morphism $\mathrm{m}_{\mathbf{1}} \in \mathcal{D}(\mathbf{e}, F\mathbf{1})$ and a natural transformation $\mathrm{m}_{A,B} \in \mathcal{D}(FA \bullet FB, F(A \otimes B))$ such that the diagrams of Fig. 1.2 commute.

A *strong* symmetric monoidal functor is a lax symmetric monoidal functor such that both $\mathrm{m}_{\mathbf{1}}$ and $\mathrm{m}_{A,B}$ are isomorphisms.

Symmetric Premonoidal Categories

In some instances, our categories will have a structure that respects all the properties of an SMC but one: the monoidal product is not bifunctorial. We call them symmetric premonoidal categories, and give here a more formal definition. We refer to [PR97] for more background.

Definition 1.2.2. *A binoidal category $(\mathcal{C}, \otimes)$ is a category with an object $A \otimes B$ for each pair of objects A, B, and two functors $A \otimes _$ and $_ \otimes A$ for every object A, sending the object B to $A \otimes B$ and $B \otimes A$ respectively.*

In a binoidal category, a morphism $f \in \mathcal{C}(A, B)$ is called **central** if for every $f' \in \mathcal{C}(A', B')$, the following diagrams commute:

$$\begin{array}{ccc} FA \bullet \mathbf{e} & \xrightarrow{FA \bullet \mathrm{m}_{\mathbf{1}}} & FA \bullet F\mathbf{1} \\ \downarrow{\scriptstyle \mathrm{ru}_{FA}} & & \downarrow{\scriptstyle \mathrm{m}_{A,\mathbf{1}}} \\ FA & \xleftarrow[F(\mathrm{ru}_A)]{} & F(A \otimes \mathbf{1}) \end{array} \qquad \begin{array}{ccc} \mathbf{e} \bullet FA & \xrightarrow{\mathrm{m}_{\mathbf{1}} \bullet FA} & F\mathbf{1} \bullet FA \\ \downarrow{\scriptstyle \mathrm{lu}_{FA}} & & \downarrow{\scriptstyle \mathrm{m}_{\mathbf{1},A}} \\ FA & \xleftarrow[F(\mathrm{lu}_A)]{} & F(\mathbf{1} \otimes A) \end{array} \qquad \begin{array}{ccc} FA \bullet FB & \xrightarrow{\mathrm{m}_{A,B}} & F(A \otimes B) \\ \downarrow{\scriptstyle \mathrm{br}_{FA,FB}} & & \downarrow{\scriptstyle F(\mathrm{br}_{A,B})} \\ FB \bullet FA & \xrightarrow[\mathrm{m}_{B,A}]{} & F(B \otimes A) \end{array}$$

$$\begin{array}{ccccc} (FA \bullet FB) \bullet FC & \xrightarrow{\mathrm{m}_{A,B} \bullet FC} & F(A \otimes B) \bullet FC & \xrightarrow{\mathrm{m}_{A \otimes B,C}} & F((A \otimes B) \otimes C) \\ \downarrow{\scriptstyle \mathrm{as}_{FA,FB,FC}} & & & & \downarrow{\scriptstyle F(\mathrm{as}_{A,B,C})} \\ FA \bullet (FB \bullet FC) & \xrightarrow[FA \bullet \mathrm{m}_{B,C}]{} & FA \bullet F(B \otimes C) & \xrightarrow[\mathrm{m}_{A,B \otimes C}]{} & F(A \otimes (B \otimes C)) \end{array}$$

Figure 1.2: Commutative Diagrams for Symmetric Monoidal Functors

$$\begin{array}{ccc} A \otimes A' & \xrightarrow{f \otimes A'} & B \otimes A' \\ \downarrow{\scriptstyle A \otimes f'} & & \downarrow{\scriptstyle B \otimes f'} \\ A \otimes B' & \xrightarrow[f \otimes B']{} & B \otimes B' \end{array} \qquad \begin{array}{ccc} A' \otimes A & \xrightarrow{f' \otimes A} & B' \otimes A \\ \downarrow{\scriptstyle A' \otimes f} & & \downarrow{\scriptstyle B' \otimes f} \\ A' \otimes B & \xrightarrow[f' \otimes B]{} & B' \otimes B \end{array}$$

In that case, we write $f \otimes f'$ and $f' \otimes f$ for the composite morphisms.

In general, in any binoidal category, we can define $f \otimes_\ell f' := (B \otimes f') \circ (f \otimes A')$ and $f \otimes_r f' := (f \otimes B') \circ (A \otimes f')$, which are **not** bifunctors, but intuitively correspond to the left-then-right and right-then-left evaluation orders. Morphisms that are central can be thought of as having "no side-effects", hence can be evaluated first or second without changing the result.

Definition 1.2.3. *A symmetric premonoidal category, or SPC, $(\mathcal{C}, \otimes, \mathbf{1})$ is a binoidal category with a left-unitor, right-unitor, associator and braiding isomorphisms which:*

- *Respect the same naturality squares and coherence diagrams as in the case of SMC.*
- *Are central morphisms.*

The notion of symmetric monoidal functor extends to premonoidal categories, keeping the same definition.

Proposition 1.2.4. *For any SPC $(\mathcal{C}, \otimes, \mathbf{1})$, the objects of $\mathcal{C}$ and the central morphisms form a subcategory, called* centre *of $\mathcal{C}$. The centre is an SMC.*

Example 1.4 Symmetric Premonoidal Categories

Since they are SMCs, **Rel** and **WRel** also are SPCs, with all morphisms being central.

Symmetric (Linear) Freyd Categories

Freyd categories [PT99b] are usually defined within the context of cartesian categories, and are known to be appropriate for modelling call-by-value languages. However, the pairing in quantum computing is represented with a tensor product, not a cartesian one, making cartesian categories too restrictive for modelling quantum computation. In fact, the no-cloning theorem of quantum mechanics enforces linearity of quantum data; linearity of data translates in the categorical world to the "product" of the category not being a categorical cartesian product, but still being a symmetric monoidal product. We will define here the "linear" variant of Freyd categories in the context of monoidal categories. While the possibility of this generalisation is mentioned in few places, we do not know any paper where this notion is properly defined, making it the first contribution of this thesis.

Definition 1.2.5. *A symmetric (linear) Freyd category, or SFC,* $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$*, consists of the following elements:*

- *An SPC* $(\mathcal{C}, \otimes, \mathbf{1})$*, called the category of* computations*, or computation category.*
- *An SMC* $(\mathcal{V}, \otimes, \mathbf{1})$*, called the category of* values*, or value category.*
- *An identity-on-objects symmetric monoidal functor* $J : \mathcal{V} \to \mathcal{C}$*, such that for all* f *morphism of* $\mathcal{V}$ *the image of* $J(f)$ *is central in* $\mathcal{C}$*. We call this functor the* Freyd inclusion*.*

It follows that $J(\mathcal{V})$ is a subcategory of the centre of $\mathcal{C}$. An SPCs $(\mathcal{C}, \otimes, \mathbf{1})$, can be seen as an SFC $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$ with $\mathcal{V}$ being the centre of $\mathcal{C}$ and J being the identity functor.

Definition 1.2.6. *An SFC* $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$ *is* affine *if* $\mathbf{1}$ *is a final object in the category of values, meaning that for every object* A *there exists a unique morphism* $\mathrm{destr}_A \in \mathcal{V}(A, \mathbf{1})$*.*

Example 1.5 Symmetric Freyd Categories

Rel and **WRel** are SMCs, and can be seen as SFC as follows: the value category and the computation categories are equal, and the Freyd inclusion is the identity functor. However, none of them are affine, since there is at least one non-identity morphism from $\mathbf{1}$ to $\mathbf{1}$: the empty relation for **Rel**, and the constant equal to zero for **WRel**. We detail in Section 5.5 an affine SFC of games and strategies, and it is possible to build and affine variation of **Rel** and **WRel**, which we do in Definition 6.4.2.

1.2.2 Various Closed Categories

Symmetric Monoidal Closed Categories

Before introducing the notion of monoidal closed category, we first need to introduce the notion of adjunction. Adjunctions are a central tool in category theory, and the natural extension of Galois connections from partial order theory. One can see adjunctions as a weak notion of equivalence of categories: two categories linked by an adjunction are categories similar enough to have well-behaved "translation functors" from one to another.

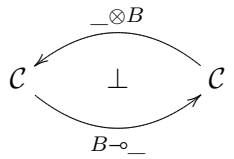
Definition 1.2.7. *For $\mathcal{C}$ and $\mathcal{D}$ two categories, with $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ two functors. We say that F and G are respectively the left and right adjoints of an adjunction, and we write $F \dashv G$ if for every objects $A \in \mathcal{C}$ and $B \in \mathcal{D}$, we have a bijection between $\mathcal{D}(FA, B)$ and $\mathcal{C}(A, GB)$ natural in A and B.*

In particular, we will be interested in one specific kind of adjunctions: currying adjunctions. We recall that the currying property is informally "a function with two arguments behaves the same as a function with one argument but returning a function". Expressed in a syntax inspired from the λ-calculus, it is the following:

$$\lambda(x, y).M \text{ "=" } \lambda x.(\lambda y.M)$$

Expressed formally, it is an adjunction of the form $\mathcal{C}(A \otimes B, C) \cong \mathcal{C}(A, B \multimap C)$, in other words $(_ \otimes B) \dashv (B \multimap _)$, for some $\otimes$ and $\multimap$ to be defined. Such a situation is usually described by monoidal closed categories, but we will later focus on two other cases: the more restricted case of compact closure and the more general case of Freyd closure.

Definition 1.2.8. *A symmetric monoidal closed category, or SMCC, $(\mathcal{C}, \otimes, \multimap, \mathbf{1})$ is an SMC $(\mathcal{C}, \otimes, \mathbf{1})$, such that $(_ \otimes B)$ has a right adjoint $(B \multimap _)$:*



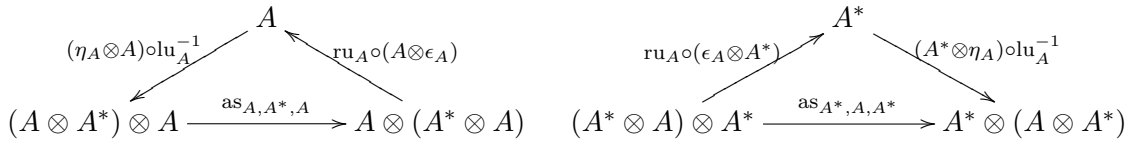
$$\mathcal{C}(A \otimes B, C) \cong \mathcal{C}(A, B \multimap C)$$

The object $A \multimap B$ shall be seen as mimicking the set of functions from A to B. Hence, SMCCs allow to consider higher order functions.

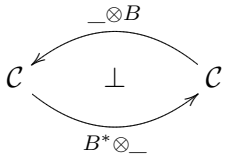
Compact Closed Categories

In compact closed categories, the object $A \multimap B$ can be expressed as $A^* \otimes B$, with $(_)^*$ an adequate notion of duality.

Definition 1.2.9. *A compact closed category, or CpCC,* $(\mathcal{C}, \otimes, \mathbf{1}, (_)^*)$ *is an SMC* $(\mathcal{C}, \otimes, \mathbf{1})$ *together with, for every object* A*, a dual object* A^**, a unit* $\eta_A \in \mathcal{C}(\mathbf{1}, A \otimes A^*)$ *and a counit* $\epsilon_A \in \mathcal{C}(A^* \otimes A, \mathbf{1})$ *such that the following diagrams commute:*

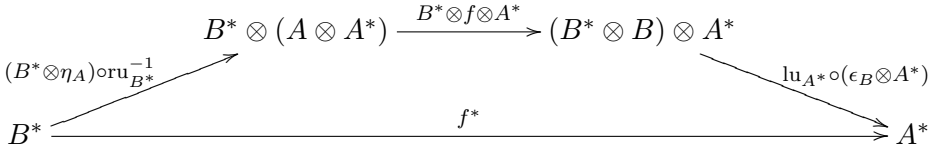


Proposition 1.2.10. *A CpCC* $(\mathcal{C}, \otimes, \mathbf{1}, (_)^*)$ *is an SMCC, i.e., it has a currying adjunction:*



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

Moreover, the dual extends to a contravariant functor by mapping $f \in \mathcal{C}(A, B)$ *to* $f^* \in \mathcal{C}(B^*, A^*)$ *defined as:*



This contravariant functor is strong symmetric monoidal, and involutive up to a natural isomorphism, called double dual isomorphism, $\mathrm{dd}_A \in \mathcal{C}(A, A^{**})$*.*

While compact closed categories can be defined in the more general context of monoidal categories which might not be symmetric, we will only consider symmetric ones.

⋆-Autonomous Categories

While we will not rely extensively on them, we remark in Section 5.4 that the categories of games and strategies we define are ⋆-autonomous categories, which can be seen as a relaxation of the constraints of a CpCC, and as a special case of SMCC.

Definition 1.2.11. *A* ⋆*-autonomous category* $(\mathcal{C}, \otimes, \multimap, \mathbf{1}, \perp)$ *is an SMCC* $(\mathcal{C}, \otimes, \multimap, \mathbf{1})$ *together with a global dualising object* $\perp$ *such that the canonical morphism* $\mathrm{dd}_A \in \mathcal{C}(A, (A \multimap \perp) \multimap \perp)$ *is an isomorphism.*

We recall that dd_A is obtained as follows from the monoidal closure:

$$\begin{array}{ccccc} \mathcal{C}(A \multimap \perp, A \multimap \perp) & \cong & \mathcal{C}((A \multimap \perp) \otimes A, \perp) & \cong & \mathcal{C}(A, (A \multimap \perp) \multimap \perp) \\ \cup & & \cup & & \cup \\ \mathbf{id}_{A \multimap \perp} & \longleftrightarrow & \mathrm{eval}_{\perp, A} & \longleftrightarrow & \mathrm{dd}_A \end{array}$$

Example 1.6 Compact Closed Categories

Rel and **WRel** are CpCC. The dual of an object is itself, *i.e.*, $A^* = A$. The unit and counit are simply defined as follows:

$$\begin{array}{rlll} \eta_A^{\mathbf{Rel}} = & \{(\star,(a,a)) \mid a \in A\} & & \\ \epsilon_A^{\mathbf{Rel}} = & \{(a,a),\star) \mid a \in A\} & & \\ \eta_A^{\mathbf{WRel}} = & (\star,(a,a)) & \mapsto & 1 \\ & \text{otherwise} & \mapsto & 0 \\ \epsilon_A^{\mathbf{WRel}} = & ((a,a),\star) & \mapsto & 1 \\ & \text{otherwise} & \mapsto & 0 \end{array}$$

The sets of morphisms $\mathbf{Rel}(\mathbf{bit}^* \otimes \mathbf{bit}, \mathbf{bit})$ and $\mathbf{WRel}(\mathbf{bit}^* \otimes \mathbf{bit}, \mathbf{bit})$ contain the representations of non-deterministic and probabilistic programs taking as an input a function from booleans to booleans, and returning a boolean.

We write $A^\perp$ for $A \multimap \perp$, and $A \parr B$ for $(A^\perp \otimes B^\perp)^\perp$. We note that $\perp \cong \mathbf{1}^\perp$.

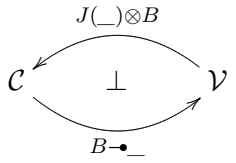
Proposition 1.2.12. *If $(\mathcal{C}, \otimes, \multimap, \mathbf{1}, \perp)$ is $\star$-autonomous then $(\mathcal{C}, \parr, \perp)$ is an SMC, $(_)^\perp$ is a full and faithful contravariant endofunctor, and $A \multimap B \cong A^\perp \parr B$.*

When we say that $(\mathcal{C}, \otimes, \parr, \mathbf{1}, (_)^\perp)$ is $\star$-autonomous, we instead mean that the category $(\mathcal{C}, \otimes, [(_)^\perp \parr _], \mathbf{1}, \mathbf{1}^\perp)$ is $\star$-autonomous. In the literature, the presentation with $\otimes$ and $\parr$ appears under the name of linearly (or weakly) distributive categories with negation, see [SCS91]. The two are known to be equivalent.

Closed (Linear) Freyd Categories

Closed (linear) Freyd categories will be the core of our semantic model, as they capture call-by-value computation. In them, the currying adjunction is between the category of values and the category of computations. This corresponds to the fact that functions (*i.e.*, λ-abstractions) are always values.

Definition 1.2.13. *A closed (linear) Freyd category, or CFC, is an SFC $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$ where for every object B the functor $J(_) \otimes B$ has a right adjoint $(B \multimap _)$:*



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

From this adjunction follows the evaluation and coevaluation natural transformations:

$$\mathrm{eval}_{A,B} \in \mathcal{C}((B \multimap A) \otimes B, A) \text{ and } \mathrm{fun}_{A,B} \in \mathcal{V}(A, B \multimap (A \otimes B))$$

SMCC and CpCC are special cases of CFC, where the category of computations and the one of values are equal ($\mathcal{C} = \mathcal{V}$), the Freyd inclusion J is the identity functor, and $A \multimap B = A \multimap B$ or $A^* \otimes B$ respectively.

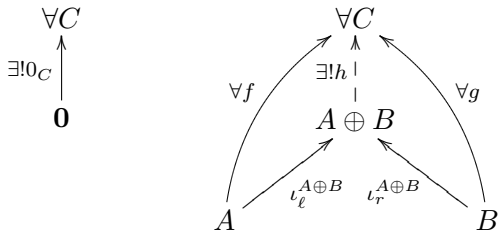
Example 1.7 Closed Freyd Categories

Since **Rel** and **WRel** are CpCCs, they are also CFCs.

1.2.3 Coproducts

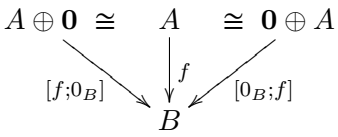
We now introduce the notion of coproducts, written $\oplus$. Coproducts are used to describe sum types, the most useful of them being booleans defined as $\mathbf{1} \oplus \mathbf{1}$.

Definition 1.2.14. *A cocartesian category $(\mathcal{C}, \oplus, \mathbf{0})$ is a category $\mathcal{C}$ with a distinguished object $\mathbf{0}$ called initial object, and for every objects A and B an object $A \oplus B$ called coproduct object, together with two morphisms $\iota_\ell^{A\oplus B} \in \mathcal{C}(A, A \oplus B)$ and $\iota_r^{A\oplus B} \in \mathcal{C}(B, A \oplus B)$ called injections, respecting the following universal properties:*



The morphism 0_C is called the initial morphism. The morphism h is called the copairing of f and g, and is written $[f; g]$.

This universal property induces a lot of other properties. Firstly, $A \oplus B$ and $\mathbf{0}$ are necessarily unique up to isomorphism. Secondly, $(\mathcal{C}, \oplus, \mathbf{0})$ is an SMC. Thirdly, the following equations are verified:



$$f_1 \oplus f_2 = [\iota_\ell^{A'\oplus B'} \circ f_1; \iota_r^{A'\oplus B'} \circ f_2]$$

Moreover, coproducts interact nicely with SPCs. Indeed, if $(\mathcal{C}, \oplus, \mathbf{0})$ is also an SPC $(\mathcal{C}, \otimes, \mathbf{1})$, then we have a natural transformation $\mathrm{dis}_{A_\ell, A_r, B} \in \mathcal{C}((A_\ell \otimes B) \oplus (A_r \otimes B), (A_\ell \oplus A_r) \otimes B)$ representing the distributivity of $\otimes$ over $\oplus$, defined as:

$$\mathrm{dis}_{A_\ell, A_r, B} = [\iota_\ell^{A_\ell \oplus A_r} \otimes B; \iota_r^{A_\ell \oplus A_r} \otimes B]$$

Definition 1.2.15. *A distributive SPC (or SMC)* $(\mathcal{C}, \otimes, \mathbf{1}, \oplus, \mathbf{0})$ *is an SPC (or SMC)* $(\mathcal{C}, \otimes, \mathbf{1})$ *and a cocartesian category* $(\mathcal{C}, \oplus, \mathbf{0})$ *such that* $0_{\mathbf{0}\otimes B}$ *is an isomorphism, and* $\mathrm{dis}_{A_\ell, A_r, B}$ *a natural isomorphism. A distributive SFC* $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1}, \oplus, \mathbf{0})$ *is an SFC* $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$ *where both* $\mathcal{C}$ *and* $\mathcal{V}$ *are distributive, and the Freyd inclusion* J *is such that* $J(f) \otimes _$ *preserves coproducts.*

In the cartesian case, a similar notion of distributive Freyd categories already appears in [Sta14]. In the case of CFCs, the distributivity of the category of computations comes for free (but not necessarily the distributivity of the category of values)

Proposition 1.2.16. *If* $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1})$ *is a CFC, and both* $(\mathcal{C}, \oplus, \mathbf{0})$ *and* $(\mathcal{V}, \oplus, \mathbf{0})$ *are cocartesian categories, then* $(\mathcal{C}, \otimes, \mathbf{1}, \oplus, \mathbf{0})$ *is a distributive SPC.*

Proof. The proof sketch is the following. We need to define an inverse for $0_{\mathbf{0}\otimes B} \in \mathcal{C}(\mathbf{0}, \mathbf{0} \otimes B)$. But we know that $\mathcal{C}(\mathbf{0} \otimes B, \mathbf{0}) \cong \mathcal{V}(\mathbf{0}, B \multimap \mathbf{0})$, so we take $0_{B\multimap\mathbf{0}} \in \mathcal{V}(\mathbf{0}, B \multimap \mathbf{0})$, and check it leads to an inverse for $0_{\mathbf{0}\otimes B}$. We then need to define an inverse for $\mathrm{dis}_{A_\ell, A_r, B} \in \mathcal{C}((A_\ell \otimes B) \oplus (A_r \otimes B), (A_\ell \oplus A_r) \otimes B)$. But we know that:

$$\mathcal{C}\left(\left(\bigoplus_{i\in\{\ell,r\}} A_i\right) \otimes B, \bigoplus_{j\in\{\ell,r\}} (A_j \otimes B)\right) \cong \mathcal{V}\left(\bigoplus_{i\in\{\ell,r\}} A_i, B \multimap \bigoplus_{j\in\{\ell,r\}} (A_j \otimes B)\right)$$

$$\cong \prod_{i\in\{\ell,r\}} \mathcal{V}\left(A_i, B \multimap \bigoplus_{j\in\{\ell,r\}} (A_j \otimes B)\right) \cong \prod_{i\in\{\ell,r\}} \mathcal{C}\left(A_i \otimes B, \bigoplus_{j\in\{\ell,r\}} (A_j \otimes B)\right)$$

So we take the two morphisms $\iota_\ell^{(A_\ell\otimes B)\oplus(A_r\otimes B)}$ and $\iota_r^{A_\ell\otimes B\oplus A_r\otimes B}$ and check they lead to an inverse for $\mathrm{dis}_{A_\ell, A_r, B}$. □

Corollary 1.2.17. *If* $(\mathcal{C}, \otimes, \mathbf{1}, (_)^*)$ *is a CpCC, and* $(\mathcal{C}, \oplus, \mathbf{0})$ *is a cocartesian category, then* $(\mathcal{C}, \otimes, \mathbf{1}, \oplus, \mathbf{0})$ *is a distributive SMC, and* $(_)^*$ *is strong symmetric monoidal with respect to* $(\mathcal{C}, \oplus, \mathbf{0})$*. We call such a category a distributive CpCC.*

Example 1.8 Cocartesian Categories

Both **Rel** and **WRel** have a cocartesian structure, given by the $\mathbf{0} = \varnothing$ and $\oplus = \uplus$. They also form distributive CpCCs, hence distributive CFCs. The copairing is:

$$\begin{array}{lcl} [R; R'] & = & \{((0,a),c) \mid (a,c) \in R\} \sqcup \{((1,b),c) \mid (b,c) \in R'\} \\ [w; w'] & : & ((0,a),c) \mapsto w(a,c) \\ & & ((1,b),c) \mapsto w'(b,c) \end{array}$$

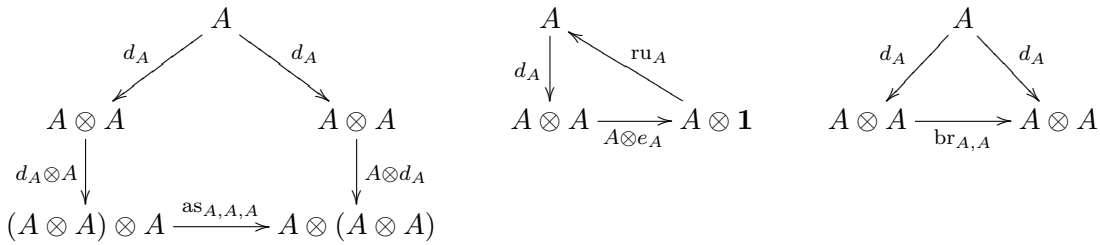


Figure 1.3: Coherence Diagrams for Commutative Comonoids

1.2.4 Linear Exponential Comonads

In Part I and Part II, we will only represent linear and affine languages, *i.e.*, languages where every variable and function can be used only once. In Part III, we will extend the results of the first two parts to a language with non-linear constructs. At a categorical level, we will use linear exponential comonads, as defined in [Mel09] among others.

Commutative Comonoid

We consider an SMC $(\mathcal{C}, \otimes, \mathbf{1})$ where objects have to be thought of as resources. A map from A to B means that we can transform the resources of A into the ones of B. The object $A \otimes B$ represents the pair of both resources, and the object $\mathbf{1}$ the absence of any resource. We want to express the notion of an object A being a non-linear resource, *i.e.*, a resource that can be duplicated and discarded at will. For that, we use commutative comonoids.

Definition 1.2.18. *A commutative comonoid on an SMC $(\mathcal{C}, \otimes, \mathbf{1})$ is an object A together with a pair of morphisms $e_A \in \mathcal{C}(A, \mathbf{1})$ and $d_A \in \mathcal{C}(A, A \otimes A)$ such that the coherence diagrams of Fig. 1.3 commute.*

Comonads

Often, non-linear resources will be of the form $!A$, which stands for "as many A as we want" and corresponds to the ! modality of linear logic. In the following definition, we axiomatise the informal notions of "if we have as many A as we want, then we can have one A" and "if we have as many A as we want, then we can have as many times as we want as many A as we want".

Definition 1.2.19. *A comonad on a category $\mathcal{C}$ is an endofunctor $!$ on $\mathcal{C}$, together with two natural transformations $\epsilon_A \in \mathcal{C}(!A, A)$ and $\delta_A \in \mathcal{C}(!A, !!A)$, called dereliction and digging, such that the coherence diagrams of Fig. 1.4 commute.*

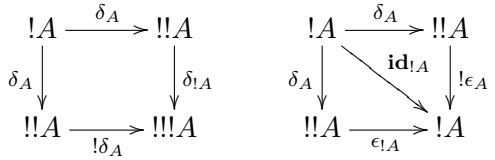


Figure 1.4: Coherence Diagrams for Comonads

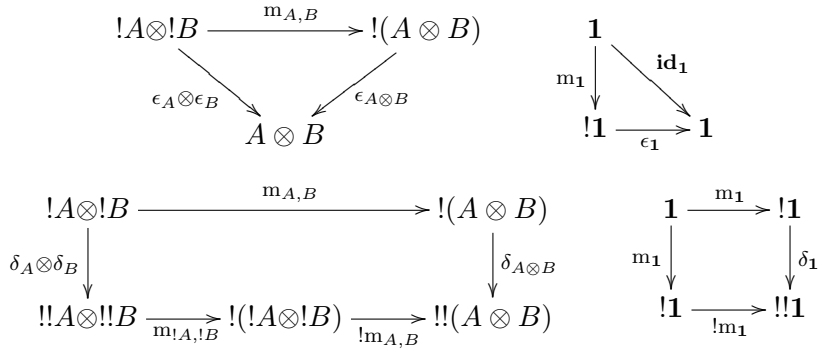


Figure 1.5: Coherence Diagrams for Symmetric Monoidal Comonads

Definition 1.2.20. *The tuple* $(!, \epsilon, \delta, \mathrm{m}, \mathrm{m}_{\mathbf{1}})$ *is a symmetric monoidal comonad on an SMC* $(\mathcal{C}, \otimes, \mathbf{1})$ *if*

- *The tuple* $(!, \epsilon, \delta)$ *is a comonad on* $\mathcal{C}$*.*
- *The functor* $!$ *is lax symmetric monoidal, with* $\mathrm{m}_{\mathbf{1}}$ *and* $\mathrm{m}_{A,B}$ *for monoidal morphisms.*
- *The natural transformations* ϵ *and* δ *are monoidal, i.e., the coherence diagrams of Fig. 1.5 commute.*

Linear Exponential Comonad

We now take both notions of commutative comonoids and symmetric monoidal comonads, and require them to be compatible with each other as follows.

Definition 1.2.21. *A linear exponential comonad on an SMC* $(\mathcal{C}, \otimes, \mathbf{1})$ *is tuple* $(!, \epsilon, \delta, \mathrm{w}, \mathrm{c},$

$\mathrm{m}, \mathrm{m}_{\mathbf{1}}$) *of an endofunctor* !*, five natural transformations and a morphism:*

$$\begin{array}{ccccc} ! & : & \mathcal{C} & \to & \mathcal{C} \\ \epsilon_A & : & !A & \to & A \\ \delta_A & : & !A & \to & !!A \\ \mathrm{w}_A & : & !A & \to & \mathbf{1} \\ \mathrm{c}_A & : & !A & \to & !A \otimes !A \\ \mathrm{m}_{A,B} & : & !A \otimes !B & \to & !(A \otimes B) \\ \mathrm{m}_{\mathbf{1}} & : & \mathbf{1} & \to & !\mathbf{1} \end{array}$$

satisfying the following properties:

- *The tuple* $(!, \epsilon, \delta, \mathrm{m}, \mathrm{m}_{\mathbf{1}})$ *is a symmetric monoidal comonad.*
- *The natural transformations* w_A *and* c_A *are lax symmetric monoidal, i.e., the diagrams on the first two lines of Fig. 1.6 commute.*
- *For every object* A*,* $(!A, \mathrm{w}_A, \mathrm{c}_A)$ *is a commutative comonoid.*
- *For every free !-coalgebra, i.e., pair* $(!A, \delta_A)$ *with* A *an object,* w_A *and* c_A *are coalgebra morphisms i.e., the third line of Fig. 1.6 commutes.*
- *The digging* δ *commutes with the weakening* w *and the contraction* c*, i.e., the fourth line of Fig. 1.6 commutes.*

The traditional definition of linear exponential comonad does not have the last item of the definition and instead has "every coalgebra morphism is a comonoid morphism", but we choose to use a simpler definition, as in [BGMZ14, Ead18], which is fully equivalent.

As shown in [Mel09], linear exponential comonads are a core components to models of MELL, hence are a good categorical tool for representing languages with both linear and non-linear behaviours.

1.3 The Linear Lambda Calculus (Λ)

The language we will study in Section 3.1 has a classical control flow together with quantum data. Accordingly, we first study how to represent the classical control flow in the absence of quantum data. After a quick overview of the language Λ, we detail the syntax, syntactic sugar and typing rules, and define its call-by-value operational semantics. We will consider two sets of typing rules for Λ, and will write LΛ for the language using the *strict* typing rules, AΛ for the language using the *affine* typing rules, and Λ for statements that apply indifferently to both. Section 1.3.2 describes the difference between the two.

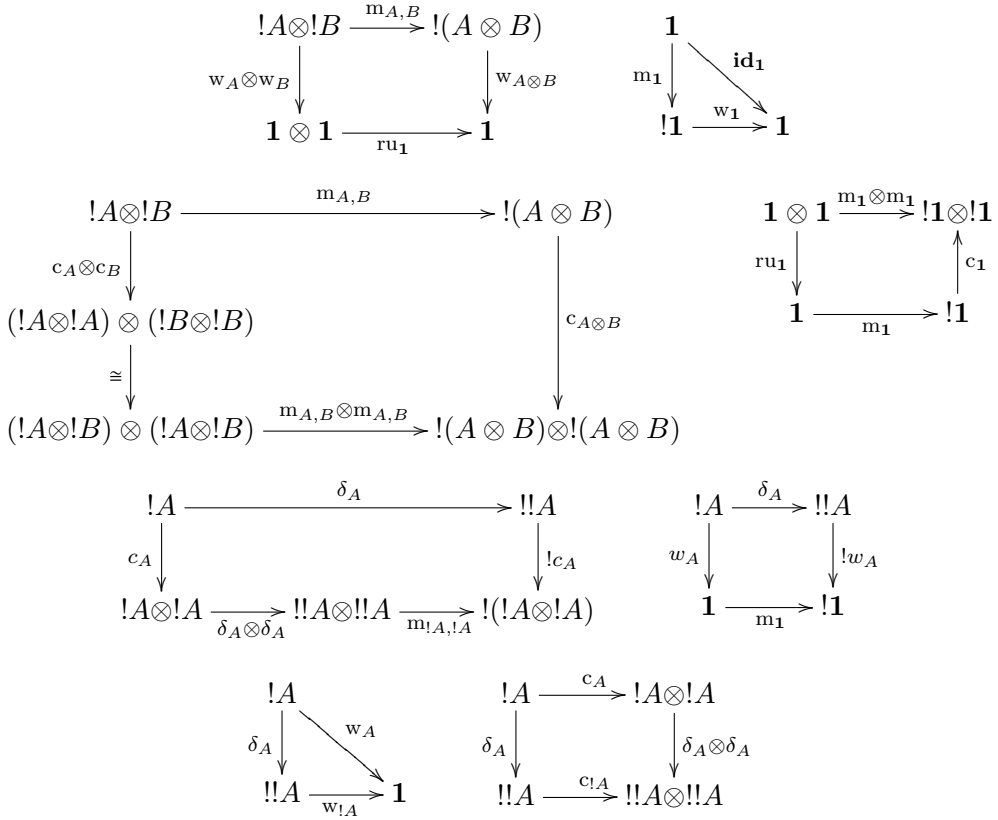


Figure 1.6: Coherence Diagrams for Linear Exponential Comonads

1.3.1 Overview of the Language

Λ is a call-by-value λ-calculus. For its types, we have the usual unit and function type, and arbitrary binary sum types and product types.

$$A, B ::= \mathbf{1} \mid A \multimap B \mid A \oplus B \mid A \otimes B$$

We write the types using notations from linear logic, as we will later enforce linear typing rules, meaning that every variable must be used exactly once. We will also consider an affine variant of Λ, where every variable must be used at most once. We write $t, s, \dots$ for terms and $x, y, \dots$ for variables. The terms of Λ are the following

- variable "x", abstraction "$\lambda x^A.t$" and application "$t\ s$" as in usual λ-calculus,
- skip "()", and sequence "$s\ ;\ t$",
- divergence "$\bot^A_{x_1,\dots,x_n}$" annotated by the set of variables it uses (see Section 1.3.2),
- injections "$\mathbf{inj}_\ell^{A\oplus B}$" and "$\mathbf{inj}_r^{A\oplus B}$", and discriminator "$\delta\ (t,\ x^A.s_1,\ y^B.s_2)$",
- pairing "$t \otimes s$" and destruction "$\mathbf{let}\ x^A \otimes y^B = t\ \mathbf{in}\ s$".

In the λ-abstraction, the discriminator, and the pair destruction, the constructs act as binders for the variables x,y,.... We work up to α-equivalence, hence allow to rename bound variables as long as it does not cause any capture. We write $\mathrm{FV}(t)$ for the set of free variables of t, with $\mathrm{FV}(\bot^A_{x_1,\dots,x_n}) = \{x_1, \dots, x_n\}$. A lot of the constructs of the language have type annotations to enforce uniqueness of typing, but we will often omit them for simplicity of notations.

In Table 1.1, we give typing rules for this language. We write typing judgements as $x_1 : A_1, \dots, x_n : A_n \vdash t : A$. The sequence $x_1 : A_1, \dots, x_n : A_n$ is called a typing context, and we expect all the x_i to be distinct variables. In particular, when concatenating typing contexts, we assume no variable appears in both initial typing contexts. While we consider a typing context as a sequence and not a set, we have the permutation typing rule to rearrange the variables when needed. The typing context shall bind all the free variables of t. The typing system we define will respect the following property:

Theorem 1.3.1 (Uniqueness of Typing)**.** *For Γ a typing context and t a term, with $FV(t) \subseteq \Gamma$, there exists at most one type A such that the typing judgement $\Gamma \vdash t : A$ is true, i.e., can be derived by the rules of Table 1.1.*

The proof of this theorem is direct, by induction over the syntax.

Structural rules

$$\frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, y : B, x : A, \Delta \vdash t : C}\ \text{permutation} \qquad \frac{\Gamma \vdash_{\mathbb{A}} t : B \qquad x \notin \mathrm{FV}(t)}{\Gamma, x : A \vdash_{\mathbb{A}} t : B}\ \begin{array}{l}(\mathrm{A}\Lambda\ \text{only})\\ \text{weakening}\end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash \bot^{A}_{x_1,\ldots,x_n} : A}\ \text{divergence}$$

λ-calculus

$$\frac{}{x : A \vdash x : A}\ \text{axiom} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^{A}.t : A \multimap B}\ \text{abstraction}$$

$$\frac{\Gamma \vdash t : A \multimap B \qquad \Delta \vdash s : A}{\Gamma, \Delta \vdash t\ s : B}\ \text{application}$$

Unit type

$$\frac{}{\vdash () : \mathbf{1}}\ \text{skip} \qquad \frac{\Gamma \vdash t : \mathbf{1} \qquad \Delta \vdash s : A}{\Gamma, \Delta \vdash t\ ;\ s : A}\ \text{sequence}$$

Tensor type

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash s : B}{\Gamma, \Delta \vdash t \otimes s : A \otimes B}\ \text{pairing} \qquad \frac{\Gamma \vdash t : A \otimes B \qquad x : A, y : B, \Delta \vdash s : C}{\Gamma, \Delta \vdash \mathbf{let}\ x \otimes y = t\ \mathbf{in}\ s : C}\ \text{let-pair}$$

Sum type

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathbf{inj}_{\ell}^{A \oplus B}\ t : A \oplus B}\ \text{left-injection} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathbf{inj}_{r}^{A \oplus B}\ t : A \oplus B}\ \text{right-injection}$$

$$\frac{\Gamma \vdash t : A \oplus B \qquad x : A, \Delta \vdash s_1 : C \qquad y : B, \Delta \vdash s_2 : C}{\Gamma, \Delta \vdash \delta\ (t,\ x^{A}.s_1,\ y^{B}.s_2) : C}\ \text{case}$$

Table 1.1: Typing Rules for Λ

Function type

$$\begin{array}{rcl@{\qquad}rcl} \textbf{let } x = t \textbf{ in } s & := & (\lambda x.s)\ t & \textbf{Unit type} & & \\ \textbf{let } f\ x = t \textbf{ in } s & := & (\lambda f.s)\ (\lambda x.t) & \lambda().t := \lambda x.x\ ;\ t & & \end{array}$$

Tensor type

$$\begin{array}{rcl} A_1 \otimes \ldots \otimes A_n & := & (A_1 \otimes \ldots) \otimes A_n \\ x_1 \otimes \ldots \otimes x_n & := & (x_1 \otimes \ldots) \otimes x_n \\ \textbf{let } x_1 \otimes \ldots \otimes x_n = t \textbf{ in } s & := & \begin{cases} \textbf{let } y \otimes x_n = t \textbf{ in} \\ \textbf{let } x_1 \otimes \ldots = y \textbf{ in} \\ s \end{cases} \\ \lambda(x_1 \otimes \ldots \otimes x_n).t & := & \lambda z.\textbf{let } x_1 \otimes \ldots \otimes x_n = z \textbf{ in } t \end{array}$$

Sum type

$$\begin{array}{rcl} \textbf{bit} & := & \mathbf{1} \oplus \mathbf{1} \\ \textbf{ff} & := & \textbf{inj}_\ell\ () \\ \textbf{tt} & := & \textbf{inj}_r\ () \\ \textbf{if } t \textbf{ then } s_t \textbf{ else } s_f & := & \delta\ (t,\ x.x; s_f,\ y.y; s_t) \\ \textbf{match } t \textbf{ with } \begin{array}{l} \mid \textbf{inj}_\ell\ x \ \mapsto\ s_\ell \\ \mid \textbf{inj}_r\ y \ \mapsto\ s_r \end{array} & := & \delta\ (t,\ x.s_\ell,\ y.s_r) \end{array}$$

Table 1.2: Syntactic Sugar for Λ

1.3.2 Linearity of the Language

The application rule (like many other rules of Λ) is multiplicative, meaning that variables used in t cannot be used in s, and reciprocally. This prevents duplication of variables, a central characteristic of linear languages. However, there are two conflicting visions of linearity, one in which every variable of the typing context must be used *at most once*, which we call *affine*, and one in which every variable in the typing context must be used *exactly once*, which we call *strictly linear*. We write AΛ for Λ together with an affine typing system, *i.e.*, we have the following weakening rule, and we write LΛ for Λ together with a strictly linear typing system, *i.e.*, without the following weakening rule. We use respectively $\vdash_{\mathbb{A}}$ and $\vdash_{\mathbb{L}}$ for typing judgements, and keep $\vdash$ for rules that apply to both.

$$\frac{\Gamma \vdash_{\mathbb{A}} t : B \qquad x \notin \mathrm{FV}(t)}{\Gamma, x : A \vdash_{\mathbb{A}} t : B}\ \begin{array}{l}(\mathrm{A}\Lambda \text{ only})\\ \text{weakening}\end{array}$$

As we expect in LΛ for variables to be used exactly once, it is practical to have the divergence $\bot$ indexed by the set of variables it "uses". When multiple $\bot$ occur in the same term, this allows to know which variable is used by which $\bot$, information required for computing the semantics of the term. This is unnecessary in AΛ, as variables can remain unused.

We note that linearity could be relaxed to not apply to non-functional variables, *i.e.*, variables of type built from $\mathbf{1}$, $\otimes$ and $\oplus$ (but not $\multimap$). Indeed, for such a type A, we can define two terms $\vdash_{\mathbb{L}} \mathbf{destr}_A : A \multimap \mathbf{1}$ and $\vdash_{\mathbb{L}} \mathbf{dupl}_A : A \multimap (A \otimes A)$ which could be inserted in the term anytime we do not want to use a variable, or anytime we want to use a variable more than once. For example

$$\mathbf{destr_{bit}} := \lambda x.\mathbf{if}\ x\ \mathbf{then}\ ()\ \mathbf{else}\ () \qquad \mathbf{dupl_{bit}} := \lambda x.\mathbf{if}\ x\ \mathbf{then}\ \mathbf{tt} \otimes \mathbf{tt}\ \mathbf{else}\ \mathbf{ff} \otimes \mathbf{ff}$$

In the forthcoming extension with quantum primitives, the quantum type **qubit** will not have any duplication term, as it is physically impossible to perfectly duplicate quantum data.

1.3.3 Typing Derivations

A typing derivation for $\Gamma \vdash t : A$ is a finite "upward" tree with the typing judgement $\Gamma \vdash t : A$ as a root. For example, a typing derivation T for $\vdash (\lambda x^{\mathbf{1}}.x) \otimes \mathbf{tt} : (\mathbf{1} \multimap \mathbf{1}) \otimes \mathbf{bit}$ is:

$$\begin{array}{c}\mathrm{T}\\ \vdots\\ \vdash (\lambda x^{\mathbf{1}}.x) \otimes \mathbf{tt} : (\mathbf{1} \multimap \mathbf{1}) \otimes \mathbf{bit}\end{array} = \dfrac{\dfrac{\dfrac{}{x : \mathbf{1} \vdash x : \mathbf{1}}}{\vdash \lambda x^{\mathbf{1}}.x : \mathbf{1} \multimap \mathbf{1}} \quad \dfrac{\dfrac{}{\vdash () : \mathbf{1}}}{\vdash \mathbf{inj}_r^{\mathbf{1}\oplus\mathbf{1}}\ () : \mathbf{1} \oplus \mathbf{1}}}{\vdash (\lambda x^{\mathbf{1}}.x) \otimes \mathbf{inj}_r^{\mathbf{1}\oplus\mathbf{1}}\ () : (\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \oplus \mathbf{1})}$$

While the typing of a term is unique, multiple derivations might exist for the same judgement $\Gamma \vdash t : A$. Non uniqueness of typing derivations comes from the following rules:

$$\frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, y : B, x : A, \Delta \vdash t : C} \quad \text{permutation} \quad \text{(strict or affine)}$$

$$\frac{\Gamma \vdash_{\mathbb{A}} t : B \quad x \notin \mathrm{FV}(t)}{\Gamma, x : A \vdash_{\mathbb{A}} t : B} \quad \text{weakening} \quad \text{(affine only)}$$

As soon as the typing context contains at least two elements, it is always possible to chain permutations in the context in pointless ways, which breaks uniqueness. Additionally, it is often possible to "move" permutation rules around in the typing derivation. Another problem appears in the AΛ because of the weakening rule: when deconstructing an application (or a similar construct), we might be able to split the context in multiple ways. In other words, when a variable of the typing context is not used in the term, each time there is a branching in the typing derivation, we have to choose if the variable will "not be used" by the left branch or by the right one.

1.3.4 Operational Semantics

We define a reduction system on terms, using call-by-value left-then-right evaluation contexts. We choose this reduction strategy as one of the objectives of this thesis is find a fully abstract model for the call-by-value quantum λ-calculus defined in [PSV14]. We note that we give the same operational semantics for AΛ and LΛ. We start by defining the values of Λ as $v, w ::= () \mid x \mid \lambda x.t \mid \mathbf{inj}_\ell\ v \mid \mathbf{inj}_r\ v \mid v \otimes w$. The reduction rules are given by Table 1.3.

This reduction system satisfies subject reduction and is deterministic (hence confluent), normalising and always progresses. More precisely, we have the following proposition.

Proposition 1.3.2. *The following properties hold:*

Subject Reduction *If $\Gamma \vdash t : A$ and $t \to s$ then $\Gamma \vdash s : A$.*

Determinism *For any term t, there exists at most one term s such that $t \to s$.*

Normalisation *For any term t, there is no infinite sequence $t \to t_1 \to t_2 \to \dots$.*

Progress *For any closed term $\vdash t : A$, either t is a value, or $\bot$, or there exists a term s such that $t \to s$.*

We will now define observational equivalence. Two terms are observationally equivalent if they behave the same in every context. Observational equivalence of two terms is not easy to check, as one must quantify over every possible context. One of the objectives of denotational semantics is to build models allowing us to reason on observational equivalence more directly.

Definition 1.3.3 (Convergence)**.** *When we have $t \to^* v$, with t a term and v a value, we write $t \Downarrow v$.*

$$
\begin{array}{rcll}
() \ ; \ t & \to & t & \\
(\lambda x.t) \ v & \to & t\{x \leftarrow v\} & \\
\textbf{let } x \otimes y = v \otimes w \textbf{ in } t & \to & t\{x \leftarrow v, y \leftarrow w\} & \\
\delta \ (\textbf{inj}_\ell \ v, \ x.t, \ y.s) & \to & t\{x \leftarrow v\} & \\
\delta \ (\textbf{inj}_r \ v, \ x.t, \ y.s) & \to & s\{y \leftarrow v\} & \\
E[t] & \to & E[s] & \text{whenever} \quad t \to s \\
E[\bot_V] & \to & \bot_{\mathrm{FV}(E[\bot_V])} & \text{whenever} \quad E[_] \neq _
\end{array}
$$

$$
\begin{array}{rcccccl}
\text{with } E[_] & ::= & _ & | & E[_] \ ; \ t & & \\
& | & E[_] \ t & | & v \ E[_] & & \\
& | & E[_] \otimes t & | & v \otimes E[_] & | & \textbf{let } x \otimes y = E[_] \textbf{ in } t \\
& | & \textbf{inj}_\ell \ E[_] & | & \textbf{inj}_r \ E[_] & | & \delta \ (E[_], \ x.t_1, \ y.t_2)
\end{array}
$$

Table 1.3: Reduction Rules for Λ terms

If $\vdash t : \mathbf{1}$, then we write $t \Downarrow$ for $t \Downarrow ()$.

Definition 1.3.4 (Observation Context)**.** *An observation context for $\Gamma \vdash A$, with Γ a typing context and A a type, is a term with a unique hole $\mathcal{O}[_]$ such that for every $\Gamma \vdash t : A$, we have $\vdash \mathcal{O}[t] : \mathbf{1}$.*

Definition 1.3.5 (Observational Equivalence)**.** *We say that two terms $\Gamma \vdash t : A$ and $\Gamma \vdash t' : A$ are observationally equivalent (for $\Gamma \vdash A$), and we write $t =_{\mathrm{obs}}^{\Gamma \vdash A} t'$, if for every observation context $\mathcal{O}[_]$ (for $\Gamma \vdash A$), we have*

$$\mathcal{O}[t] \Downarrow \iff \mathcal{O}[t'] \Downarrow$$

We will often keep the annotation $\Gamma \vdash A$ implicit.

1.4 Freyd Categories as a Categorical Model for Λ

1.4.1 Denotational Semantics

Definition 1.4.1. *A denotational semantics for Λ is an SPC $(\mathcal{C}, \otimes, \mathbf{1})$ and an operation $[\![-]\!]$ which associates to every type A of Λ an object $[\![A]\!] \in \mathcal{C}$ and to every typed term $\Gamma \vdash t : A$ of Λ a morphism $[\![t]\!]^{\Gamma \vdash A} \in \mathcal{C}(\bigotimes_{(x_i : A_i) \in \Gamma} [\![A_i]\!], [\![A]\!])$.*

Similarly to $=_{\mathrm{obs}}^{\Gamma \vdash A}$, we will sometimes keep the typing annotation of $[\![-]\!]^{\Gamma \vdash A}$ implicit. One of the uses of denotational semantics is to characterise convergence and observational equivalence. The gold standard of denotational semantics is full abstraction, which means the semantics exactly characterise them. More formally:

Definition 1.4.2. *The semantics is said to be*

Sound *if for every term* $\vdash t : \mathbf{1}$*:*

$$t \Downarrow \implies [\![t]\!] = [\![()]\!]$$

Sound and Adequate *if additionally for every term* $\vdash t : \mathbf{1}$*:*

$$[\![t]\!] = [\![()]\!] \implies t \Downarrow$$

Fully Abstract *if for every pair of terms* $\Gamma \vdash t : A$ *and* $\Gamma \vdash s : A$*:*

$$[\![t]\!] = [\![s]\!] \iff t =_{\text{obs}} s$$

Proposition 1.4.3. *If the semantics is fully abstract, then it is sound and adequate.*

To those standard properties, we add some useful intermediate properties:

Value Substituting if for every term $\Gamma, x : A \vdash t : B$, and every value $\Delta \vdash v : A$:

$$[\![t]\!] \circ \left(\left(\bigotimes_{(x_i:A_i)\in\Gamma} [\![A_i]\!] \right) \otimes [\![v]\!] \right) = [\![t\{x \leftarrow v\}]\!]$$

Invariant if for every term $\Gamma \vdash t : A$, if $t \to s$ then

$$[\![t]\!] = [\![s]\!]$$

1.4.2 Freyd Categories as a Model

We now build a denotational semantics using Freyd categories. We take a distributive CFC $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ respecting the following additional constraints:

Bottom We have a distinguished central morphism $\bot \in \mathcal{C}(\mathbf{1}, \mathbf{0})$

Non-Trivial The objects $\mathbf{0}$ and $\mathbf{1}$ are not isomorphic. In particular, $\mathbf{id_1} \neq 0_\mathbf{1} \circ \bot$.

When interpreting AΛ, we will also require the affine property (*i.e.*, $\mathbf{1}$ is a final object in $\mathcal{V}$). We do not require it when interpreting LΛ.

Example 1.9

Both **Rel** and **WRel** are distributive CpCCs (hence CFCs) which are non-trivial, and have a bottom morphism, which is the empty relation. None of the two are affine.

We start by giving the interpretation of the types, then the interpretation of typing derivations, and finally the interpretation of typed terms. The semantics of types is quite straightforward, and is described in Table 1.4. We also define the semantics of a typing context as $[\![\Gamma]\!] := \bigotimes_{(x_i:A_i)\in\Gamma} [\![A_i]\!]$. In Tables 1.5 and 1.6, we give to every typing derivation

$$\begin{array}{rclcrcl} [\![\mathbf{1}]\!] & := & \mathbf{1} & & [\![A \oplus B]\!] & := & [\![A]\!] \oplus [\![B]\!] \\ [\![A \multimap B]\!] & := & [\![A]\!] \multimap [\![B]\!] & & [\![A \otimes B]\!] & := & [\![A]\!] \otimes [\![B]\!] \end{array}$$

Table 1.4: Denotational Semantics for Λ types

Structural Rules:

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x : A, y : B, \Delta \vdash t : C\end{array}}{\Gamma, y : B, x : A, \Delta \vdash t : C} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x : A, y : B, \Delta \vdash t : C\end{array} \right]\!\!\right] \circ ([\![\Gamma]\!] \otimes \mathrm{br}_{[\![B]\!],[\![A]\!]} \otimes [\![\Delta]\!])$$

$$\left[\!\!\left[\dfrac{}{\Gamma \vdash \bot_V^A : A} \right]\!\!\right] := 0_{[\![A]\!]} \circ 0^{-1}_{\mathbf{0} \otimes [\![\Gamma]\!]} \circ (\bot \otimes [\![\Gamma]\!]) \circ \mathrm{lu}^{-1}_{[\![\Gamma]\!]}$$

AΛ only: $\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{A}} t : B\end{array}}{\Gamma, x : A \vdash_{\mathbb{A}} t : B} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{A}} t : B\end{array} \right]\!\!\right] \circ ([\![\Gamma]\!] \otimes J(\mathbf{destr}_{[\![A]\!]}))$

Functions type:

$$\left[\!\!\left[\dfrac{}{x : A \vdash x : A} \right]\!\!\right] := \mathbf{id}_{[\![A]\!]}$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x : A \vdash t : B\end{array}}{\Gamma \vdash \lambda x^A.t : A \multimap B} \right]\!\!\right] := J\left([\![A]\!] \multimap \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x : A \vdash t : B\end{array} \right]\!\!\right] \right) \circ \mathrm{fun}_{[\![\Gamma]\!],[\![A]\!]}$$

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ \Gamma \vdash t : A \multimap B & \Delta \vdash s : A\end{array}}{\Gamma, \Delta \vdash t\ s : B} \right]\!\!\right] := \mathrm{eval}_{[\![B]\!],[\![A]\!]} \circ \left(\left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : A \multimap B\end{array} \right]\!\!\right] \otimes^{\ell} \left[\!\!\left[\begin{array}{c}\mathrm{S}\\ \vdots \\ \Delta \vdash s : A\end{array} \right]\!\!\right] \right)$$

Table 1.5: Denotational Semantics of Λ Typing Derivations, Part 1

of $\Gamma \vdash t : A$ a semantics in $\mathcal{C}([\![\Gamma]\!], [\![A]\!])$. We will then use Theorem 1.4.6 to define the semantics of typing judgements. There is no subtlety involved in those definitions: we proceed by structural induction on the typing derivations, and we use the structure of the distributive CFC. The use of the left-then-right tensor $\otimes^{\ell}$ correspond to the left-then-right reduction strategy. Note that we kept the associator isomorphism implicit in the definition of the semantics.

Definition 1.4.4. *If V is a typing derivation for a value $\Gamma \vdash v : A$, then we define its value-semantics $[\![V]\!]_{\mathbf{v}} \in \mathcal{V}([\![\Gamma]\!], [\![A]\!])$ as in Table 1.7.*

It is the same inductive definition as $[\![V]\!]$, up to the following changes: we propagate the $[\![-]\!]_{\mathbf{v}}$ in every rule (except in the λ-abstraction rule where do not replace the $[\![-]\!]$ by a $[\![-]\!]_{\mathbf{v}}$), and we remove the occurrences of J.

Proposition 1.4.5. *We always have $J([\![V]\!]_{\mathbf{v}}) = [\![V]\!]$*

Unit type:

$$\left[\!\!\left[\frac{}{\vdash () : \mathbf{1}} \right]\!\!\right] := \mathbf{id_1}$$

$$\left[\!\!\left[\frac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots\\ \Gamma \vdash t : \mathbf{1} & \Delta \vdash s : A\end{array}}{\Gamma, \Delta \vdash t \ ; \ s : A} \right]\!\!\right] := \mathrm{lu}_{[\![A]\!]} \circ \left(\left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : \mathbf{1}\end{array} \right]\!\!\right] \otimes^{\ell} \left[\!\!\left[\begin{array}{c}\mathrm{S}\\ \vdots\\ \Delta \vdash s : A\end{array} \right]\!\!\right] \right)$$

Tensor type:

$$\left[\!\!\left[\frac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots\\ \Gamma \vdash t : A & \Delta \vdash s : B\end{array}}{\Gamma, \Delta \vdash t \otimes s : A \otimes B} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : A\end{array} \right]\!\!\right] \otimes^{\ell} \left[\!\!\left[\begin{array}{c}\mathrm{S}\\ \vdots\\ \Delta \vdash s : B\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\frac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots\\ \Gamma \vdash t : A \otimes B & x : A, y : B \Delta \vdash s : C\end{array}}{\Gamma, \Delta \vdash \mathbf{let}\ x^A \otimes y^B = t\ \mathbf{in}\ s : C} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{S}\\ \vdots\\ x : A, y : B, \Delta \vdash s : C\end{array} \right]\!\!\right] \circ \left(\left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : A \otimes B\end{array} \right]\!\!\right] \otimes [\![\Delta]\!] \right)$$

Sum type:

$$\left[\!\!\left[\frac{\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : A\end{array}}{\Gamma \vdash \mathbf{inj}_{\ell}^{A \oplus B}\ t : A \oplus B} \right]\!\!\right] := \iota_{\ell}^{[\![A]\!] \oplus [\![B]\!]} \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : A\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\frac{\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : B\end{array}}{\Gamma \vdash \mathbf{inj}_{r}^{A \oplus B}\ t : A \oplus B} \right]\!\!\right] := \iota_{r}^{[\![A]\!] \oplus [\![B]\!]} \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : B\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\frac{\begin{array}{ccc}\mathrm{T} & \mathrm{S1} & \mathrm{S2}\\ \vdots & \vdots & \vdots\\ \Gamma \vdash t : A \oplus B & x : A, \Delta \vdash s_1 : C & y : B, \Delta \vdash s_2 : C\end{array}}{\Gamma, \Delta \vdash \delta\ (t,\ x^A.s_1,\ y^B.s_2) : C} \right]\!\!\right]$$

$$:= \left[\left[\!\!\left[\begin{array}{c}\mathrm{S1}\\ \vdots\\ x : A, \Delta \vdash s_1 : C\end{array} \right]\!\!\right] ; \left[\!\!\left[\begin{array}{c}\mathrm{S2}\\ \vdots\\ y : B, \Delta \vdash s_2 : C\end{array} \right]\!\!\right] \right] \circ \mathrm{dis}^{-1}_{[\![A]\!], [\![B]\!], [\![\Delta]\!]} \circ \left(\left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots\\ \Gamma \vdash t : A \oplus B\end{array} \right]\!\!\right] \otimes [\![\Delta]\!] \right)$$

Table 1.6: Denotational Semantics of Λ Typing Derivations, Part 2

Structural Rules:

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{V}\\ \vdots \\ \Gamma, x : A, y : B, \Delta \vdash v : C\end{array}}{\Gamma, y : B, x : A, \Delta \vdash v : C} \right]\!\!\right]_{\mathbf{v}} := [\![V]\!]_{\mathbf{v}} \circ ([\![\Gamma]\!] \otimes \mathrm{br}_{[\![B]\!],[\![A]\!]} \otimes [\![\Delta]\!])$$

AΛ only: $\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{V}\\ \vdots \\ \Gamma \vdash_{\mathbb{A}} v : B\end{array}}{\Gamma, x : A \vdash_{\mathbb{A}} v : B} \right]\!\!\right]_{\mathbf{v}} := [\![V]\!]_{\mathbf{v}} \circ ([\![\Gamma]\!] \otimes \mathbf{destr}_{[\![A]\!]})$

Functions type:

$$\left[\!\!\left[\dfrac{}{x : A \vdash x : A} \right]\!\!\right]_{\mathbf{v}} := \mathbf{id}_{[\![A]\!]}$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{V}\\ \vdots \\ \Gamma, x : A \vdash t : B\end{array}}{\Gamma \vdash \lambda x^A.t : A \multimap B} \right]\!\!\right]_{\mathbf{v}} := ([\![A]\!] \multimap [\![V]\!]_{\mathbf{v}}) \circ \mathrm{fun}_{[\![\Gamma]\!],[\![A]\!]}$$

Unit type:

$$\left[\!\!\left[\dfrac{}{\vdash () : \mathbf{1}} \right]\!\!\right]_{\mathbf{v}} := \mathbf{id}_{\mathbf{1}}$$

Tensor type:

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{V} & \mathrm{W}\\ \vdots & \vdots \\ \Gamma \vdash v : A & \Delta \vdash w : B\end{array}}{\Gamma, \Delta \vdash v \otimes w : A \otimes B} \right]\!\!\right]_{\mathbf{v}} := [\![V]\!]_{\mathbf{v}} \otimes [\![W]\!]_{\mathbf{v}}$$

Sum type:

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{V}\\ \vdots \\ \Gamma \vdash v : A\end{array}}{\Gamma \vdash \mathbf{inj}_{\ell}^{A \oplus B}\ v : A \oplus B} \right]\!\!\right]_{\mathbf{v}} := \iota_{\ell}^{[\![A]\!] \oplus [\![B]\!]} \circ [\![V]\!]_{\mathbf{v}}$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{V}\\ \vdots \\ \Gamma \vdash v : B\end{array}}{\Gamma \vdash \mathbf{inj}_{r}^{A \oplus B}\ v : A \oplus B} \right]\!\!\right]_{\mathbf{v}} := \iota_{r}^{[\![A]\!] \oplus [\![B]\!]} \circ [\![V]\!]_{\mathbf{v}}$$

Table 1.7: Value Denotational Semantics of Λ Typing Derivations

Theorem 1.4.6. *If $\Gamma \vdash t : A$ has two typing derivations T and T', then we have $[\![T]\!] = [\![T']\!]$. As the interpretation is independent from the typing derivation, we write it $[\![t]\!]^{\Gamma \vdash A}$. We define $[\![-]\!]_{\mathbf{v}}^{\Gamma \vdash A}$ similarly.*

This theorem is quite standard. One simple way to prove it is: for every typing judgement, we choose a canonical typing derivation (where the permutations and weakening are postponed as much as possible, *etc.*) and show that all the other typing derivations can be transformed into this canonical one through permutations of rules that preserve the semantics. To show that permuting rules preserve the semantics, we rely on the fact that all the structural morphisms (braiding, ...) are value morphisms (morphisms that are in the image of $\mathcal{V}$ by J), as it means they are in the centre of the premonoid $\otimes$, and that the naturality of $\mathrm{fun}_{A,B}$ applies.

1.4.3 Proof of Soundness and Adequacy

We prove that our model is value-substituting, satisfies context factorisation, is invariant, sound, adequate and the direct implication of the full abstraction equivalence. The reverse implication does not hold for any arbitrary non-trivial distributive CFC with a bottom, though we could prove with a method similar to Theorem 3.2.14 that it holds for **Rel** in the case of LΛ.

Lemma 1.4.7 (Value Substitution)**.** *For every term $\Gamma, x : A \vdash t : B$ and every value $\Delta \vdash v : A$:*

$$[\![t]\!] \circ ([\![\Gamma]\!] \otimes [\![v]\!]) = [\![t\{x \leftarrow v\}]\!]$$

Proof. We choose a typing derivation for t and v, and take the corresponding typing derivation for $t\{x \leftarrow v\}$ (*i.e.*, we replace the axiom[a] for x, in the typing derivation of t by the typing derivation of v). We proceed inductively on the typing derivation of t. The base case is $x : A \vdash x : A$, and we indeed have:

$$[\![x]\!] \circ (\mathbf{1} \otimes [\![v]\!]) = [\![x\{x \leftarrow v\}]\!]^{\Delta \vdash A}$$

For the inductive case, since v is a value, we have $[\![v]\!] = J([\![v]\!]_{\mathbf{v}})$. Going through all the typing rules, we need to use the following properties that every $f \in \mathcal{V}(A, B)$ respects:

- Since $J(f)$ is a central morphism, we always have

$$(B \otimes h) \circ (J(f) \otimes C) = (J(f) \otimes C) \circ (A \otimes h)$$

- Since $\mathrm{fun}_{A,C}$ is natural in its first argument, we have

$$\mathrm{fun}_{B,C} \circ f = (C \multimap f \otimes C) \circ \mathrm{fun}_{A,C}$$

- Since $\mathbf{0}$ is an initial object,

$$0^{-1}_{\mathbf{0}\otimes B} \circ (\mathbf{0} \otimes J(f)) = 0^{-1}_{\mathbf{0}\otimes A}$$

- Since $\mathrm{dis}_{C_\ell,C_r,A}$ is natural in all its arguments, we have

$$\mathrm{dis}^{-1}_{C_\ell,C_r,B} \circ (C_\ell \oplus C_r) \otimes J(f) = ((C_\ell \otimes J(f)) \oplus (C_r \otimes J(f))) \circ \mathrm{dis}^{-1}_{C_\ell,C_r,A}$$

□

[a] Linearity of the language ensures that there is exactly one axiom for the free variable x.

The value substitution lemma ensures that for simple β-reductions $(\lambda x.t)\ v \to t\{x \leftarrow v\}$, the semantics is preserved. This however does not directly proves the case $E[(\lambda x.t)\ v] \to E[t\{x \leftarrow v\}]$ for $E[-]$ an evaluation context. We remark that evaluation contexts never capture variables, so $\mathrm{FV}(t) \subseteq \mathrm{FV}(E[t])$. This allows us to state the following lemma.

Lemma 1.4.8 (Context Factorisation)**.** *For every pair of terms $\Gamma \vdash s : A$ and $\Gamma, \Delta \vdash E[s] : B$, with $E[-]$ an evaluation context, we have a morphism $[\![E]\!] \in \mathcal{C}([\![A]\!] \otimes [\![\Delta]\!], [\![B]\!])$ such that*

$$[\![E[s]]\!] = [\![E]\!] \circ ([\![s]\!] \otimes [\![\Delta]\!])$$

Proof. We follow the same induction as in the previous lemma, with s instead of v, $E[x]$ instead of t, and $E[s]$ instead of $[\![t\{x \leftarrow v\}]\!]$. All the cases are trivial. □

We now have all the ingredients to prove invariance.

Lemma 1.4.9 (Invariance)**.** *For every pair of terms $\Gamma \vdash t : A$ and $\Gamma \vdash s : A$*

$$t \to s \implies [\![t]\!] = [\![s]\!]$$

Proof. We proceed by induction on $\to$. The rule for the sequence is true. The rules for the λ-abstraction, the pairs, and the discriminator directly follow from the value substitution lemma. Remains the evaluation context rules. We use the context factorisation lemma and obtain $[\![E[t]]\!]^{\Gamma,\Delta\vdash B} = [\![E]\!] \circ \left([\![t]\!]^{\Gamma\vdash A} \otimes [\![\Delta]\!]\right)$. Invariance by the reduction rule for evaluation context and bottom follow immediately from this factorisation property. □

In the simple case of Λ, soundness and adequacy are a direct consequence of the invariance lemma.

Theorem 1.4.10 (Soundness and Adequacy)**.** *For every term* $\vdash t : \mathbf{1}$*, we have*

$$t \Downarrow () \iff [\![t]\!] = [\![()]\!]$$

Proof. Using strong normalisation, we obtain that either $t \to^* ()$ or $t \to^* \bot$. Since the category is non-trivial, $()$ and $\bot$ have different semantics, so we have the equivalence. □

It follows that we have the direct implication of the full abstraction:

Corollary 1.4.11. *For every pair of terms* $\Gamma \vdash t : A$ *and* $\Gamma \vdash s : A$*, we have*

$$[\![t]\!] = [\![s]\!] \implies t =_{\text{obs}} s$$

Proof. We define the type $P = \bigotimes_{(x_i:A_i)\in\Gamma} A_i$, and the values $v_t = \lambda\left(\bigotimes_{(x_i:A_i)\in\Gamma} x_i^{A_i}\right).t$ and $v_s = \lambda\left(\bigotimes_{(x_i:A_i)\in\Gamma} x_i^{A_i}\right).s$. We assume that $[\![t]\!] = [\![s]\!]$. From the definition of $[\![-]\!]$, it follows that $[\![v_t]\!] = [\![v_s]\!]$. We now prove that $v_t =_{\text{obs}}^{P\multimap A} v_s$. We take an observation context $\mathcal{O}[_]$ for $\vdash P \multimap A$, and we use the value-substituting lemma to obtain that

$$[\![\mathcal{O}[v_t]]\!] = [\![\mathcal{O}[x]]\!] \circ [\![v_t]\!] = [\![\mathcal{O}[x]]\!] \circ [\![v_s]\!] = [\![\mathcal{O}[v_s]]\!]$$

Using adequacy, it follows that $\mathcal{O}[v_t] \Downarrow \iff \mathcal{O}[v_s] \Downarrow$, hence $v_t =_{\text{obs}}^{P\multimap A} v_s$. We note that $t =_{\text{obs}} s \iff v_t =_{\text{obs}} v_s$. As we can transform an observation context of one into the other by adding application or λ-abstractions around the hole. □

Chapter 2

Introduction to Quantum Computation

2.1 Hilbert spaces

In this section, we go through some basic mathematical notions central in quantum physics. Namely complex numbers, Hilbert spaces, tensor products and positive operators.

2.1.1 Complex Numbers

We write $\mathbf{i}$ and $-\mathbf{i}$ for the two complex numbers of square equal to minus one. Every complex number $z \in \mathbb{C}$ can be written as $z = \mathrm{Re}(z) + \mathbf{i}\,\mathrm{Im}(z)$, with $\mathrm{Re}(z), \mathrm{Im}(z) \in \mathbb{R}$ being its *real part* (or abscissa) and its *imaginary part* (or ordinate). As such, every complex number can be seen as a point of the plan $\mathbb{R}^2$, as in Fig. 2.1. The representation with real and imaginary parts is not the most meaningful one in our case, we will more interested in the polar representation $z = |z| \cdot \mathbf{e}^{\mathbf{i}\mathrm{Arg}(z)}$ with $|z| \in \mathbb{R}_{\geq 0}$ and $\mathrm{Arg}(z) \in [0, 2\pi)$ being its *module* (or radius) and its *argument* (or angle).

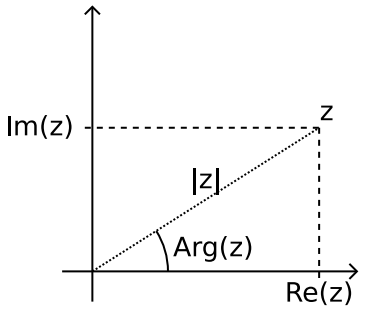


Figure 2.1: The Complex Plane

Indeed, in quantum mechanics, we frequently need to represent a probability $p \in [0,1]$ together with a cyclic information $\theta \in [0, 2\pi)$ (modulo 2π) called phase. For that, we use the complex number $\sqrt{p} \cdot \mathbf{e}^{\mathbf{i}\theta}$.

Properties 2.1.1. *We recall here some basic properties of complex numbers.*

$$\begin{array}{lcll} z & = & |z| \cdot \mathbf{e}^{\mathbf{i}\mathrm{Arg}(z)+2\mathbf{i}k\pi} & (\forall k \in \mathbb{Z}) \\ \bar{z} & = & \mathrm{Re}(z) - \mathbf{i}\ \mathrm{Im}(z) & = & |z| \cdot \mathbf{e}^{-\mathbf{i}\mathrm{Arg}(z)} \\ z \cdot \bar{z} & = & \mathrm{Re}(z)^2 + \mathrm{Im}(z)^2 & = & |z|^2 \end{array}$$

We will often use matrices of complex numbers, hence we recall here the notation $A^\dagger$ for the conjugate transpose of A:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix}^\dagger = \begin{pmatrix} \overline{a_{1,1}} & \cdots & \overline{a_{m,1}} \\ \vdots & & \vdots \\ \overline{a_{1,n}} & \cdots & \overline{a_{m,n}} \end{pmatrix}$$

We say that a matrix has size $m \times n$ if it has m rows and n columns. We say that a square matrix has size n if it has n rows and n colmuns. A particular kind of matrix that will come back often is a unitary matrix. A matrix U is called unitary if and only if both $U \times U^\dagger$ and $U^\dagger \times U$ are identity matrices.

2.1.2 Hilbert Spaces

A single complex number is quite limited in terms of the information it can represent. To represent more information at once, we will use vectors in a Hilbert space. So, we will first recall some definitions about vector spaces and linear operators.

Definition 2.1.2. *For $(V, \ldots, +, \mathbf{0})$ a complex vector space, an hermitian sesquilinear form $\langle -|-\rangle$ is a function from $V \times V$ to $\mathbb{C}$ which respects:*

$$\underline{\text{Sesquilinear:}} \quad \begin{array}{lcl} \langle a \cdot v_1 + b \cdot v_2 | w \rangle & = & \overline{a} \cdot \langle v_1 | w \rangle + \overline{b} \cdot \langle v_2 | w \rangle \\ \langle v | a \cdot w_1 + b \cdot w_2 \rangle & = & a \cdot \langle v | w_1 \rangle + b \cdot \langle v | w_2 \rangle \end{array}$$

$\underline{\text{Hermitian:}}$ $\langle w|v\rangle = \overline{\langle v|w\rangle}$

It is said to be an inner product if additionally it is positive definite:

$\underline{\text{Positive:}}$ $\langle v|v\rangle \in \mathbb{R}_{\geq 0}$

$\underline{\text{Definite:}}$ $\langle v|v\rangle = 0 \iff v = \mathbf{0}$

A Hilbert space is by definition a complete complex inner-product space, however, since we will only use finite dimensional Hilbert spaces, the definition simplifies to the following one.

Definition 2.1.3. *A (finite dimensional) Hilbert space is a finite dimensional complex vector space* $(H, \cdot, +, \mathbf{0})$ *together with an operator* $\langle -|-\rangle : H \times H \to \mathbb{C}$ *which is an inner product. Hilbert spaces come with a norm, defined as* $||v|| = \sqrt{\langle v|v\rangle}$.

In other words, Hilbert spaces are simply the equivalent of Euclidean spaces for complex numbers. Similarly to Euclidean spaces, Hilbert spaces admit *orthonormal bases*. An orthonormal basis of H is a basis $(h_1, \ldots, h_{\dim(H)})$ such that for all $1 \leq i, j \leq \dim(H)$ we have $\langle h_i|h_j\rangle = 1$ if $i = j$ and 0 otherwise. Given an orthonormal basis $(h_1, \ldots, h_n)$ of H, every vector v of H can be written as a column matrix of complex numbers:

$$\mathcal{M}(v) = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \langle v|h_1\rangle \\ \vdots \\ \langle v|h_n\rangle \end{pmatrix}$$

Given $(h_1, \ldots, h_n)$ and $(k_1, \ldots, k_m)$ orthonormal bases of H and K, any linear operator $f : H \to K$ can be represented as a complex matrix:

$$\mathcal{M}(f) = \begin{pmatrix} f_{1,1} & \cdots & f_{1,n} \\ \vdots & & \vdots \\ f_{m,1} & \cdots & f_{m,n} \end{pmatrix} = \begin{pmatrix} \langle k_1|f(h_1)\rangle & \cdots & \langle k_1|f(h_n)\rangle \\ \vdots & & \vdots \\ \langle k_m|f(h_1)\rangle & \cdots & \langle k_m|f(h_n)\rangle \end{pmatrix}$$

With those notations, application and composition can be computed using matrix products as follows:

$$\begin{array}{rcl} \mathcal{M}(f(v)) & = & \mathcal{M}(f) \times \mathcal{M}(v) \\ \mathcal{M}(g \circ f) & = & \mathcal{M}(g) \times \mathcal{M}(f) \\ \langle v|w\rangle & = & \mathcal{M}(v)^\dagger \times \mathcal{M}(w) \end{array}$$

In the following, we always consider Hilbert spaces together with an orthonormal basis, allowing us to go back and forth between vectorial and matricial representations.

The most commonly used Hilbert spaces are the $\mathbb{C}^n$ together with their canonical inner product $\langle (v_1, \ldots, v_n)|(w_1, \ldots, w_n)\rangle := \sum_{i=1}^n \overline{v_i} \cdot w_i$ and the canonical basis

$$\{(1, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)\}$$

In fact, without loss of generality, one could only use spaces of the form $\mathbb{C}^n$, since any Hilbert space of dimension $n \in \mathbb{N}$ is isomorphic to $\mathbb{C}^n$. In particular the dual space H^* of H is isomorphic to H, and we recall here its definition:

- H^* is the set of linear forms over H, *i.e.*, $\{\langle h|-\rangle : H \to \mathbb{C} \mid h \in H\}$.
- Its inner product is $\langle\langle h|-\rangle|\langle k|-\rangle\rangle = \langle k|h\rangle$. The inversion is required because we have $\langle \lambda h|-\rangle = \overline{\lambda}\langle h|-\rangle$.
- Its orthonormal basis is the set of $\langle h|-\rangle$ for h in the orthonormal basis of H.

However, this isomorphism between H and H^* is not *natural*[1] when the dimension of H is at least two, in other words the isomorphism depends on the bases chosen for H and H^*. We still have that H^{**} and H are naturally isomorphic, and so are $\mathbb{C}$ and $\mathbb{C}^*$.

In practice, we will identify objects that are related by a natural isomorphism, hence we will identify H^{**} with H, and $\mathbb{C}$ with $\mathbb{C}^*$, but we will consider H and H^* as distinct spaces when of dimension at least two. This choice to work with every finite dimensional Hilbert space, rather than only Hilbert spaces of the form $\mathbb{C}^n$, should allow for the results presented in this thesis to be easily generalised to a more algebraic setting such as dagger compact closed categories as defined in [Sel07].

The dagger operation on matrices extend to an operation on linear operator. For $f : H \to K$, we write $f^\dagger : K \to H$ for the unique linear operator such that $\mathcal{M}(f)^\dagger = \mathcal{M}(f^\dagger)$. Equivalently, $f^\dagger : K \to H$ is the unique linear operator such that $\langle u|f(v)\rangle = \langle f^\dagger(u)|v\rangle$ for every $u \in H, v \in K$.

2.1.3 Tensor Product

A very useful operation on Hilbert spaces is the *tensor product* $\otimes$. Intuitively, if vectors of the Hilbert space H represent the different states of a first system, and K of a second system, then the vectors of the Hilbert space $H \otimes K$ represent the state of each of the systems *and* how they are correlated to each other. Mathematically, $H \otimes K$ is a Hilbert space defined as follows:

1. We consider the free vector space F over pairs of elements of H and K, *i.e.*, $F = \{\sum_i c_i(v_i, w_i) \mid \forall i, c_i \in \mathbb{C}, v_i \in H, w_i \in K\}$.

2. We quotient it by the reflexive transitive closure of $\equiv$ defined as

$$\begin{array}{ll} (v + v', w) \equiv (v, w) + (v', w) & (c \cdot v, w) \equiv c(v, w) \\ (v, w + w') \equiv (v, w) + (v, w') & (v, c \cdot w) \equiv c(v, w) \end{array}$$

   We write $v \otimes w$ the equivalence class of (v, w).

3. The inner product is simply $\langle \sum_i c_i \cdot v_i \otimes w_i | \sum_j c'_j \cdot v'_j \otimes w'_j \rangle = \sum_i \sum_j \overline{c_i} c'_j \langle v_i | v'_j \rangle \langle w_i | w'_j \rangle$

It follows that $\otimes : H \times K \to H \otimes K$ is a bilinear function, and that $H \otimes K$ has dimension $\dim(H) \dim(K)$. If H has $(h_1, \ldots, h_n)$ as an orthonormal basis, and K has $(k_1, \ldots, k_m)$, then $(h_1 \otimes k_1, \ldots, h_n \otimes k_m)$ is an orthonormal basis of $H \otimes K$.

If we take a vector $u \in H \otimes K$, then two situations arise: either $u = v \otimes w$, then we say u represents separable states, or u is a non-trivial sum of $v_i \otimes w_i$, then we say u represents an entangled state.

[1] This notion of natural isomorphism coincide with the categorical notion of natural isomorphism.

We now recall some basic constructions and properties of the tensor product. From two linear operators $f : H \to H'$ and $g : K \to K'$, we define $f \otimes g : H \otimes K \to H' \otimes K'$ as follows:

$$(f \otimes g)\left(\sum_i c_i \cdot v_i \otimes w_i\right) = \sum_i c_i \cdot f(v_i) \otimes g(w_i)$$

On matrix representations, the tensor product can be computed using the Kronecker product $\otimes$ of matrices:

$$\mathcal{M}(v \otimes w) = \mathcal{M}(v) \otimes \mathcal{M}(w) = \begin{pmatrix} v_1 \cdot \mathcal{M}(w) \\ \vdots \\ v_n \cdot \mathcal{M}(w) \end{pmatrix}$$

$$\mathcal{M}(f \otimes g) = \mathcal{M}(f) \otimes \mathcal{M}(g) = \begin{pmatrix} f_{1,1} \cdot \mathcal{M}(g) & \cdots & f_{1,n} \cdot \mathcal{M}(g) \\ \vdots & & \vdots \\ f_{m,1} \cdot \mathcal{M}(g) & \cdots & f_{m,n} \cdot \mathcal{M}(g) \end{pmatrix}$$

Finally, we state a very simple but important property, which will allow us to represent functions as vectors.

Proposition 2.1.4. *For H, K two Hilbert spaces, linear operators from H to K form a Hilbert space, written $\mathcal{L}(H, K)$. We have $\mathcal{L}(H, K) \cong H^* \otimes K$. More generally, for H, K, L three Hilbert spaces, we have $\mathcal{L}(H \otimes K, L) \cong \mathcal{L}(H, K^* \otimes L)$.*

For $f \in \mathcal{L}(H, H')$ and $g \in \mathcal{L}(K, K')$, and consider their tensor product $f \otimes g \in \mathcal{L}(H, H') \otimes \mathcal{L}(K, K')$, it coincides (up to isomorphism) with $f \otimes g \in \mathcal{L}(H \otimes H', K \otimes K')$ defined above.

2.1.4 Positive Operators

We now introduce positive operators, which are used in quantum mechanics to represent *mixed states*, *i.e.*, probability distributions of regular quantum states, which are called *pure states*. We recall that an hermitian sesquilinear form is a function $\langle -|- \rangle : H \times H \to \mathbb{C}$ which respects the sesquilinear and hermitian conditions described in Definition 2.1.2.

Definition 2.1.5. *We write* $\mathrm{Herm}(H)$ *for the* ***real*** *vector space of hermitian operators over H, where $f : H \to H$ is an hermitian operator if and only if $f^\dagger = f$. Its dimension is* $\dim(\mathrm{Herm}(H)) = \dim(H)^2$. *We define* $\mathbf{tr} : \mathrm{Herm}(H) \to \mathbb{R}$ *as follows:*

$$\mathbf{tr}(f) = \mathbf{tr}(\mathcal{M}(f)) = \mathbf{tr}\begin{pmatrix} m_{1,1} & \cdots & m_{1,\dim(H)} \\ \vdots & & \vdots \\ m_{\dim(H),1} & \cdots & m_{\dim(H),\dim(H)} \end{pmatrix} = \sum_{i=1}^{\dim(H)} m_{i,i} \in \mathbb{R}$$

We write $\mathrm{Pos}(H)$ *for the real convex cone (in* $\mathrm{Herm}(H)$*) of positive operators.*

We recall that f is a positive operator if and only if $\mathcal{M}(f) = \mathcal{M}(f)^\dagger$, and all the eigenvalues of $\mathcal{M}(f)$ are positive or null. Using the fact that hermitian operators are always unitarily diagonalisable, we can obtain the following result

Proposition 2.1.6 (Eigenvector Decomposition)**.** *If $f \in \mathrm{Pos}(H)$ and $n = \dim(H)$, then we have*

$$\mathcal{M}(f) = \sum_{i=1}^{n} \lambda_i \mathcal{M}(v_i) \times \mathcal{M}(v_i)^\dagger$$

with λ_i being the eigenvalues and v_i being a corresponding eigenvector of norm one. Choosing a square root for each of the eigenvalues, it follows that there exists $f_i : \mathbb{C} \to H$ such that

$$f = \sum_{i=1}^{n} f_i \circ f_i^\dagger$$

This is a well-known result, and we can refer to [Rob19] for one of the multiple proofs. The trace $\mathbf{tr}$ is a linear operation on $\mathrm{Herm}(H)$, and $\mathbf{tr}(f \otimes g) = \mathbf{tr}(f) \cdot \mathbf{tr}(g)$. Moreover, we have for every $f \in \mathrm{Pos}(H)$ and every coefficient $a \in \mathbb{C}$ of $\mathcal{M}(f)$, $|a| \leq \mathbf{tr}(f)$.

Positive operators of trace lesser or equal to one are called subdensity operators, we write the corresponding set $\mathrm{Pos}_{\leq 1}(H)$. Since $\mathrm{Pos}(H)$ is a real convex cone, and the trace is linear, it follows that subdensity operators are stable under subprobability distributions $\sum_i p_i f_i$ with $\sum_i p_i \leq 1$.

Quantum states[2] will later be represented by normalised vectors of a Hilbert space H, *i.e.*, vectors $v \in H$ such that $||v|| = \langle v|v\rangle = 1$. To each of those vectors we associate a subdensity operator $f_v \in \mathrm{Pos}(H)$ defined as $f_v(u) := \langle v|u\rangle \cdot v$ or in other words, $\mathcal{M}(f_v) := \mathcal{M}(v) \times \mathcal{M}(v)^\dagger$. Since subdensity operators are stable under subprobability distributions, it follows that given a subprobability distribution $\{(p_i, v_i) \mid i \in I\}$ of normalised vectors of H, *i.e.*, quantum states, we can associate to it a canonical subdensity operator $\sum_{i\in I} p_i f_{v_i}$.

An important note is that multiple subprobability distributions can correspond to the same subdensity operator, see Example 2.1. This is by design, as in Theorem 2.3.2, we show that such subprobability distributions of quantum states are indistinguishable through experiments.

Proposition 2.1.7. *A basis for the real vector space $\mathrm{Herm}(\mathbb{C}^2)$ is, in matrix form:*

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \quad \begin{pmatrix} 1/2 & \mathbf{i}/2 \\ -\mathbf{i}/2 & 1/2 \end{pmatrix}$$

Proof. We consider an element of $\mathrm{Herm}(\mathbb{C}^2)$. Its associated matrix is $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with $a, b, c, d \in \mathbb{C}$ satisfying $M^\dagger = M$, *i.e.*, $\overline{a} = a$, $\overline{b} = c$, $\overline{c} = b$ and $\overline{d} = d$. In other

[2] We refer here to pure quantum states as defined in Section 2.3.2.

Example 2.1 Probability distributions corresponding to the same density operator

We consider the probability distributions p_1 and p_2 which can be written with matrix representation as

$$p_1 = \left\{ \left(\frac{1}{2}, \begin{pmatrix}1\\0\end{pmatrix}\right), \left(\frac{1}{2}, \begin{pmatrix}0\\1\end{pmatrix}\right) \right\} \qquad p_2 = \left\{ \left(\frac{1}{2}, \begin{pmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{pmatrix}\right), \left(\frac{1}{2}, \begin{pmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{pmatrix}\right) \right\}$$

They both correspond to the following density operator $\frac{1}{2}\mathbf{id}$ as:

$$\frac{1}{2}\begin{pmatrix}1&0\\0&0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0&0\\0&1\end{pmatrix} = \begin{pmatrix}1/2&0\\0&1/2\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1/2&1/2\\1/2&1/2\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1/2&-1/2\\-1/2&1/2\end{pmatrix}$$

words, $M = \begin{pmatrix}\alpha & \beta + \mathbf{i}\gamma\\ \beta - \mathbf{i}\gamma & \delta\end{pmatrix}$ with $\alpha, \beta, \gamma, \delta \in \mathbb{R}$. It follows that

$$M = (\alpha - 2\beta - 2\gamma)\begin{pmatrix}1&0\\0&0\end{pmatrix} + (\delta - 2\beta - 2\gamma)\begin{pmatrix}0&0\\0&1\end{pmatrix} + 2\beta\begin{pmatrix}1/2&1/2\\1/2&1/2\end{pmatrix} + 2\gamma\begin{pmatrix}1/2&\mathbf{i}/2\\-\mathbf{i}/2&1/2\end{pmatrix}$$

and this decomposition is unique. This proves that the four associated vectors form a basis. □

Another notable property is the existence of a partial order called the Loewner order, defined on positive operators as: $f \sqsubseteq g$ whenever $g - f$ is a positive operator.

2.1.5 The Categories of Hilbert Spaces

Two different categories arise from the previous definitions, the rather simple category of Hilbert spaces and linear operators between them, and the much richer category of Hilbert spaces and completely positive maps.

The Category Hilb

Proposition 2.1.8. *The category* **Hilb** *whose objects are (finite dimensional) Hilbert spaces and whose morphisms are linear operators is a compact closed category* $(\mathbf{Hilb}, \otimes, \mathbf{1}, (_)^*)$ *with* $\otimes$ *being the usual tensor product,* $\mathbf{1} = \mathbb{C}$*, and* $(_)^*$ *being the usual dual of Hilbert spaces.*

To avoid confusion with the category defined next, we write $\mathbf{id}_H^{\mathbf{Hilb}}, \mathrm{as}_{H,K,L}^{\mathbf{Hilb}}, \ldots$ for the identity, the associator, and all the other structural morphisms. Using Proposition 2.1.4, we can rewrite the unit and counit in a much more intuitive form:

- The unit $\eta_H^{\mathbf{Hilb}}$ can be seen as a morphism of $\mathbf{Hilb}(\mathbb{C}, \mathcal{L}(H^*, H^*))$ that maps a scalar z to z times the identity operator on H^*.
- The counit $\epsilon_H^{\mathbf{Hilb}}$ can be seen as a morphism of $\mathbf{Hilb}(\mathcal{L}(H, H), \mathbb{C})$ that maps an operator on H to its trace.

Since we do not identify H and H^*, we see spaces "without a star" as positives and containing actual vectors and spaces "with a star" as negatives and containing linear forms expecting a vector as an input.

The Category $\mathbf{Hilb}_{\leq 1}$

A relevant subcategory of **Hilb** is $\mathbf{Hilb}_{\leq 1}$, the category of Hilbert spaces and contraction morphisms, *i.e.*, morphisms $f \in \mathbf{Hilb}(H, K)$ such that $f \circ f^\dagger \sqsubseteq \mathbf{id}_H^{\mathbf{Hilb}}$

Proposition 2.1.9. $(\mathbf{Hilb}_{\leq 1}, \otimes, \mathbf{1})$ *is an SMC, and it is the minimal sub-SMC of* **Hilb** *containing all the morphisms f such that $\mathcal{M}(f) = \begin{pmatrix} U & 0 \\ 0 & 0 \end{pmatrix}$ with U a unitary matrix and the* 0*s some rectangular null matrices.*

In Section 2.3.2, we give a more precise characterisation when the spaces are all of dimension a power of two, and show that $\mathbf{Hilb}_{\leq 1}$ can be seen as the subcategory of **Hilb** excluding all the operations that are not physically realisable. The operations we keep in $\mathbf{Hilb}_{\leq 1}$ precisely correspond to changing the orthonormal basis in which we consider our vectors, projecting our vectors onto a smaller space, embedding our vectors into a bigger space, or any sequence of those operations.

The Category CPM

We will now define **CPM** of completely positive maps, which is the category we will actually use to represent quantum information. More information about the category **CPM** can be found in [Sel04, Sel07]. Informally, maps of **CPM** are linear functions from matrices to matrices that preserve positivity in a strong sense. To formally define it, we first define a temporary notion of *matrix operator*.

A matrix operator f from a Hilbert space H to a Hilbert space K is simply a linear operator from $\mathcal{L}(H, H)$ to $\mathcal{L}(K, K)$. The category of Hilbert spaces and matrix operators is an SMC with $\otimes$ for monoidal product and $\mathbf{1}$ for unit. Indeed, Proposition 2.1.4 implies that $\mathcal{L}(H, H) \otimes \mathcal{L}(K, K) \cong \mathcal{L}(H \otimes K, H \otimes K)$.

Definition 2.1.10. *The objects of the category* **CPM** *are Hilbert spaces, and a morphism f from H to K is a matrix operator which is:*

Positive: *the image of* $\mathrm{Pos}(H)$ *by f is included in* $\mathrm{Pos}(K)$*,*

Completely Positive: *for every Hilbert space* L, $\mathbf{id}_L^{\mathbf{CPM}} \otimes f$ *is positive, i.e., the image of* $\mathrm{Pos}(L \otimes H)$ *by* $\mathbf{id}_L^{\mathbf{CPM}} \otimes f$ *is included in* $\mathrm{Pos}(L \otimes K)$.

We recall that the Loewner order is defined as $f \sqsubseteq g$ whenever $g - f$ is completely positive. This means that for f a matrix operator, $f \in \mathbf{CPM}$ if and only if $f \sqsupseteq 0$. In particular, if we consider a linear combination of **CPM** maps (with positive and negative real coefficients), to prove that it is a **CPM** maps, we just need to prove that it is greater than 0 for the Loewner order.

Since matrix operators are linear operators, the definition of dagger extend to them and we have that if $f \in \mathbf{CPM}(H, K)$ then $f^\dagger \in \mathbf{CPM}(K, H)$. We can lift morphisms of **Hilb** into morphisms of **CPM** as follows

$$\begin{array}{rccc} (\!|-|\!) : & \mathbf{Hilb}(H,K) & \to & \mathbf{CPM}(H,K) \\ & f & \mapsto & (\!|f|\!) : (\phi \mapsto f \circ \phi \circ f^\dagger) \end{array}$$

Proposition 2.1.11. *The category* **CPM** *is a compact closed category* $(\mathbf{CPM}, \otimes, \mathbf{1}, (_)^*)$*, with* $\otimes$ *being the usual tensor product,* $\mathbf{1} = \mathbb{C}$*, and* $(_)^*$ *being the usual dual of Hilbert spaces.*

The structure of the SMC comes from $\mathbf{Hilb}_{\leq 1}$ through $(\!|-|\!)$, which will be a strong symmetric monoidal functor. We write $\mathbf{id}_H^{\mathbf{CPM}}, \mathrm{as}_{H,K,L}^{\mathbf{CPM}}, \ldots$ for the images of $\mathbf{id}_H^{\mathbf{Hilb}}, \mathrm{as}_{H,K,L}^{\mathbf{Hilb}}, \ldots$ Proving that **CPM** is compact closed is not immediate, but is a direct consequence of the following theorem.

Theorem 2.1.12 (Choi-Jamiołkowski isomorphism)**.** *The set of completely positive maps* $\mathbf{CPM}(H, K)$ *is isomorphic to the set of positive operators* $\mathrm{Pos}(H^* \otimes K)$*. This isomorphism preserves and reflects*[3] *the Loewner order* $\sqsubseteq$*.*

This theorem is quite standard, and its proof relies on the isomorphism between $\mathcal{L}(H, K)$ and $H^* \otimes K$ from Proposition 2.1.4 to obtain that $\mathcal{L}(\mathcal{L}(H, H), \mathcal{L}(K, K)) \cong (H^*)^* \otimes H^* \otimes K^* \otimes K \cong \mathcal{L}(H^* \otimes K, H^* \otimes K)$. Then, proving that the completely positive map on the left hand side is sent to a positive operator on the right hand side is a direct verification once we make explicit this isomorphism. Proving that we reach all the positive operators relies on Proposition 2.1.6.

An important consequence of this theorem is Corollary 2.1.13, which justifies the use of **CPM** as an extension of **Hilb**. Indeed, morphisms of **Hilb** describe quantum operations on pure states, *i.e.*, along one branch of the execution, whereas morphisms of **CPM** describe operations over mixed states, *i.e.*, over multiple branches of the execution at once.

Corollary 2.1.13 (Krauss Representation)**.** *For* $f \in \mathbf{CPM}(H, K)$*, there exist* $n \in \mathbb{N}$ *and* $f_i \in \mathbf{Hilb}(H, K)$ *for all* $1 \leq i \leq n$*, such that* $f = \sum_{i=1}^n (\!|f_i|\!)$*.*

[3] An isomorphism reflects a relation if its inverse preserves it.

Proof. We consider $f \in \mathbf{CPM}(H,K)$, and use Theorem 2.1.12 to obtain $\hat{f} \in \mathrm{Pos}(H^* \otimes K)$. We then use Proposition 2.1.6 and obtain $\hat{f} = \sum_i g_i \circ g_i^\dagger$. It follows that there exists some $g'_i \in \mathcal{L}(\mathbf{1}, H^* \otimes K)$ such that $f = \sum_i g'_i \circ g_i'^\dagger$. Using Proposition 2.1.4 we obtain $f_i \in \mathcal{L}(H,K)$ such that $f = \sum_{i=1}^n (\!| f_i |\!)$. □

The Category $\mathbf{CPM}_{\leq 1}$

We now introduce the notion of superoperator. Similarly to $\mathbf{Hilb}_{\leq 1}$ being the subcategory of **Hilb** containing the physically realisable operations, $\mathbf{CPM}_{\leq 1}$ is the subcategory of **CPM** containing the physically realisable operations. Superoperators are sometimes called quantum operations or quantum channels, and come from the work of Sudarshan [SMR61] on general stochastic transformations for density matrices, though we use the more modern formalism of Choi and Kraus.

Definition 2.1.14. *We say that $f \in \mathbf{CPM}(H,K)$ is a* superoperator *if the image of $\mathrm{Pos}_{\leq 1}(H)$ by f is included in $\mathrm{Pos}_{\leq 1}(K)$. We write $\mathbf{CPM}_{\leq 1}$ for the subcategory of superoperators.*

We note that the trace defined in Definition 2.1.5 is a superoperator, and we write $\mathbf{Tr}_H \in \mathbf{CPM}_{\leq 1}(H, \mathbf{1})$ for the associated map $f \mapsto \mathbf{tr}(f)$ and $\mathbf{Tr}_H^K$ for the map associated to the partial trace, *i.e.*, $\mathbf{Tr}_H^K = \mathrm{lu}_K^{\mathbf{CPM}} \circ (\mathbf{Tr}_H \otimes \mathbf{id}_K^{\mathbf{CPM}}) \in \mathbf{CPM}_{\leq 1}(H \otimes K, K)$. Its dagger $\mathbf{Tr}_H^\dagger \in \mathbf{CPM}(\mathbf{1}, H)$ generates the identity matrix. We write $\mathbf{1}_H \in \mathbf{CPM}_{\leq 1}(\mathbf{1}, H)$ for the map that generates the normalised identity matrix.

$$\mathbf{Tr}_H^\dagger : z \mapsto z \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \mathbf{1}_H : z \mapsto z \begin{pmatrix} \frac{1}{\dim H} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\dim H} \end{pmatrix}$$

The subcategory $\mathbf{CPM}_{\leq 1}$ is an SMC, but is not compact closed. We still have a correspondence between superoperators and a subclass of positive operators.

Corollary 2.1.15 (Bounded Choi-Jamiołkowski isomorphism)**.** *The set of superoperators $\mathbf{CPM}_{\leq 1}(H,K)$ is isomorphic to the set $\{f \in \mathrm{Pos}(H^* \otimes K) \mid \mathbf{Tr}_K^{H^*}(f) \sqsubseteq \mathbf{id}_{H^*}^{\mathrm{Pos}}\}$.*

Proof. We refine Theorem 2.1.12 by noting that

$$\begin{aligned} \mathbf{Tr}_K^{H^*}(\hat{f}) &= (\mathbf{Tr}_K^{H^*} \circ (\mathbf{id}_{\mathcal{L}(H^*,H^*)} \otimes f)) \left(\textstyle\sum_{i,i'} H^*_{i,i'} \otimes H_{i,i'}\right) \\ &= (\mathbf{id}_{\mathcal{L}(H^*,H^*)} \otimes (\mathbf{Tr}_K \circ f)) \left(\textstyle\sum_{i,i'} H^*_{i,i'} \otimes H_{i,i'}\right) \\ &= \widehat{\mathbf{Tr}_K \circ f} \end{aligned}$$

Using linearity of $\hat{-}$, it follows that $\mathbf{Tr}_H - \mathbf{Tr}_K \circ f \in \mathbf{CPM}(H,K)$ if and only if

$\hat{\mathbf{Tr}}_H - \mathbf{Tr}_K^{H^*}(\hat{f}) \in \mathrm{Pos}(H^*)$, *i.e.*, $\mathbf{Tr}_K^{H^*}(\hat{f}) \sqsubseteq \mathbf{id}_{H^*}^{\mathrm{Pos}}$. □

In the same way that **CPM** morphisms are sums of morphisms of **Hilb**, $\mathbf{CPM}_{\leq 1}$ morphisms will appear as a probability distribution of morphisms of $\mathbf{Hilb}_{\leq 1}$.

Proposition 2.1.16. *For $f \in \mathbf{Hilb}_{\leq 1}$, we always have $(\!|f|\!) \in \mathbf{CPM}_{\leq 1}$. Conversely, for $f \in \mathbf{CPM}_{\leq 1}(H, K)$, there exists $n \in \mathbb{N}$ and $(f_i, p_i) \in \mathbf{Hilb}_{\leq 1}(H, K) \times [0, 1]$ for all $1 \leq i \leq n$, with $\sum_{i=1}^n p_i = 1$, such that*

$$f = \sum_{i=1}^{n} p_i \cdot (\!|f_i|\!)$$

Lastly, we note that $\mathbf{CPM}_{\leq 1}(\mathbf{1}, \mathbf{1}) \cong [0, 1]$, as every morphism of $\mathbf{CPM}_{\leq 1}(\mathbf{1}, \mathbf{1})$ is of the form $\lambda \cdot \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$ for $\lambda \in [0, 1]$.

Observational Characterisation of CPM Morphisms

We are now interested in ways to "observe" the difference between **CPM** morphisms. This will be a central tool for later full abstraction results. We first note the following property:

Proposition 2.1.17. *Two morphisms $f, g \in \mathbf{CPM}(H, K)$ such that $f(M) = g(M)$ for every $M \in \mathrm{Pos}_{\leq 1}(H)$ are necessarily equal.*

Proof. We take $f, g \in \mathbf{CPM}(H, K)$ such that $f(M) = g(M)$ for every $M \in \mathrm{Pos}_{\leq 1}(H)$. For $M_0 \in \mathrm{Pos}(H)$, if $\mathbf{tr}(M_0) > 1$ then $\frac{M_0}{\mathbf{tr}(M_0)} \in \mathrm{Pos}_{\leq 1}(H)$ so

$$\frac{f(M_0)}{\mathbf{tr}(M_0)} = f\left(\frac{M_0}{\mathbf{tr}(M_0)}\right) = g\left(\frac{M_0}{\mathbf{tr}(M_0)}\right) = \frac{g(M_0)}{\mathbf{tr}(M_0)}$$

It follows that for every $M \in \mathrm{Pos}(H)$, $f(M) = g(M)$. For $M_1 \in \mathrm{Herm}(H)$, if we diagonalise M_1 and split the positive and negative eigenvalues, we obtain $M_1 = M_1^+ - M_1^-$ with $M_1^+, M_1^- \in \mathrm{Pos}(H)$. So

$$f(M_1) = f(M_1^+) - f(M_1^-) = g(M_1^+) - g(M_1^-) = g(M_1)$$

It follows that for every $M \in \mathrm{Herm}(H)$, $f(M) = g(M)$. For $M_2 \in \mathcal{L}(H, H)$. We write $M_2^+ = \frac{M_2 + M_2^\dagger}{2} \in \mathrm{Herm}(H)$ and $M_2^- = \frac{M_2 - M_2^\dagger}{2} \in \mathrm{Herm}(H)$. So

$$f(M_2) = f(M_2^+) + \mathbf{i} f(M_2^-) = g(M_2^+) + \mathbf{i} g(M_2^-) = g(M_2)$$

It follows that for every $M \in \mathcal{L}(H, H)$, $f(M) = g(M)$. □

Testing every $M \in \mathrm{Pos}_{\leq 1}(H)$ to determine if f and g are different is impractical, so we search a more limited number of subdensity operators that are "enough". If $\mathbf{CPM}(H, K)$

was a vector space, we could just take elements of its basis. But the set $\mathbf{CPM}(H,K)$ is not a complex vector space, as positivity is not preserved when multiplying by a complex number. It is not a real vector space either, but it can be seen as the positive fragment of the real vector space of linear operators from $\mathbf{Hilb}(H,H)$ to $\mathbf{Hilb}(K,K)$. Using a real basis well-chosen to contain only superoperators, we obtain the following results.

Proposition 2.1.18 (Basis for **CPM**)**.** *We write $\mathcal{G}^{\mathbb{C}^2}$ for the set composed of the four following morphisms of $\mathbf{CPM}_{\leq 1}(\mathbf{1},\mathbb{C}^2)$:*

$$G_0^{\mathbb{C}^2} : z \mapsto z\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} \qquad G_1^{\mathbb{C}^2} : z \mapsto z\begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}$$
$$G_2^{\mathbb{C}^2} : z \mapsto z\begin{pmatrix}1/2 & 1/2\\ 1/2 & 1/2\end{pmatrix} \quad G_3^{\mathbb{C}^2} : z \mapsto z\begin{pmatrix}1/2 & \mathbf{i}/2\\ -\mathbf{i}/2 & 1/2\end{pmatrix}$$

And write $\mathcal{T}^{\mathbb{C}^2}$ for the set composed of the four following morphisms of $\mathbf{CPM}_{\leq 1}(\mathbb{C}^2,\mathbf{1})$:

$$T_0^{\mathbb{C}^2} : \begin{pmatrix}a & b\\ c & d\end{pmatrix} \mapsto a \qquad T_1^{\mathbb{C}^2} : \begin{pmatrix}a & b\\ c & d\end{pmatrix} \mapsto d$$
$$T_2^{\mathbb{C}^2} : \begin{pmatrix}a & b\\ c & d\end{pmatrix} \mapsto \tfrac{a+b+c+d}{2} \quad T_3^{\mathbb{C}^2} : \begin{pmatrix}a & b\\ c & d\end{pmatrix} \mapsto \tfrac{a-\mathbf{i}b+\mathbf{i}c+d}{2}$$

We have for every $f,g \in \mathbf{CPM}(\mathbb{C}^2,\mathbb{C}^2)$:

$$f = g \iff \forall b \in \mathcal{G}^{\mathbb{C}^2}, \forall b' \in \mathcal{T}^{\mathbb{C}^2}, b' \circ f \circ b = b' \circ g \circ b$$

To prove this result, we just need to note that (1) for every $M \in \mathrm{Pos}_{\leq 1}(\mathbb{C}^2)$, there exists a linear combination $\ell = \sum_{0\leq n\leq 3} c_n G_n^{\mathbb{C}^2}$ with $c_n \in \mathbb{R}$ such that $\ell : z \mapsto z \cdot M$, and (2) for every $M,N \in \mathrm{Pos}_{\leq 1}(\mathbb{C}^2)$, if $M \neq N$ then there exists $b' \in \mathcal{T}^{\mathcal{C}^2}$ such that $b'(M) \neq b'(N)$.

This proposition also hold for the dual space $\mathbb{C}^2$. We write $\mathcal{G}^{(\mathbb{C}^2)^*}$ for the set composed of the four morphisms of $\mathbf{CPM}(\mathbf{1},\mathbb{C}^2)$ defined as $G_i^{(\mathbb{C}^2)^*} := \left(G_i^{\mathbb{C}^2}\right)^*$, and similarly $\mathcal{T}^{(\mathbb{C}^2)^*}$ for the set composed of the four morphisms of $\mathbf{CPM}(\mathbb{C}^2,\mathbf{1})$ defined as $T_i^{(\mathbb{C}^2)^*} := \left(T_i^{\mathbb{C}^2}\right)^*$. We keep the natural isomorphism between $\mathbb{C}$ and $\mathbb{C}^*$ implicit.

We now want to generalise it to higher dimensions than 2. For $H = H_1 \otimes \ldots \otimes H_n$ with each of the H_k being $\mathbb{C}^2$ or $(\mathbb{C}^2)^*$, we write $\mathcal{G}^H$ and $\mathcal{T}^H$ for the respective sets of morphisms defined as:

$$G^H_{i_1\ldots i_n} := G^{H_1}_{i_1} \otimes \ldots \otimes G^{H_n}_{i_n} \in \mathbf{CPM}(\mathbf{1},H) \qquad T^H_{i_1\ldots i_n} := T^{H_1}_{i_1} \otimes \ldots \otimes T^{H_n}_{i_n} \in \mathbf{CPM}(H,\mathbf{1})$$

Corollary 2.1.19. *For every $f,g \in \mathbf{CPM}(H,K)$, with H and K being tensors of multiple $\mathbb{C}^2$ and $(\mathbb{C}^2)^*$:*

$$f = g \iff \forall b \in \mathcal{G}^H, \forall b' \in \mathcal{T}^K, b' \circ f \circ b = b' \circ g \circ b$$

2.2 Preliminaries on Partial Orders

We will eventually need to take limits within **CPM**. For that, we need to define the completion of **CPM** with infinitary elements. This completion relies on the D-completion of partials orders as in [ZF10], which we define here. Indeed, **CPM** is already a complete metric space, meaning that some limits are already defined, and naive completion methods[4] fail to preserve the pre-existing limits.

2.2.1 Partial Orders

Definition 2.2.1. *A partially ordered set $(S, \leq)$, or poset, is a set S together with a binary relation $\leq$ on S which is reflexive (i.e., $s \leq s$), transitive (i.e., $s \leq s' \leq s'' \implies s \leq s''$) and anti-symmetric (i.e., $s \leq s' \leq s \implies s = s'$).*

We extend the unions of sets to posets. We consider two posets $(S, \leq_S)$ and $(T, \leq_T)$, and define $\leq_{S \cup T}$, $\leq_{S \sqcup T}$ and $\leq_{S \uplus T}$ as follows:

- $a \leq_{S \cup T} b \iff \begin{cases} a, b \in S \\ a \leq_S b \end{cases}$ or $\begin{cases} a, b \in T \\ a \leq_T b \end{cases}$.
- $\leq_{S \sqcup T}$ is identical to $\leq_{S \cup T}$, but assumes that S and T are disjoint.
- $(i, a) \leq_{S \uplus T} (j, b) \iff \begin{cases} i = 0 = j \\ a \leq_S b \end{cases}$ or $\begin{cases} i = 1 = j \\ a \leq_T b \end{cases}$

We note that $S \cup T$ might not be a poset, as we might obtain a pre-order, or a relation which is not transitive. However, $S \sqcup T$ and $S \uplus T$ are always a poset.

In a poset $(S, \leq)$, we say that $X \subseteq S$ is *down-closed* if for all $x' \leq x \in X$ we have $x' \in X$.

2.2.2 Directed Complete Partial Orders

Definition 2.2.2. *A poset $(S, \leq)$ is said to be*

Directed *if it is non-empty and for every $s, s' \in S$, there exists an upper bound of $\{s, s'\}$ in S (i.e., $b \in S$ such that $s \leq b \geq s'$).*

Directed Complete *if each of its directed subsets has a unique supremum (i.e., least upper bound).*

Note that directed complete posets, or dcpos, are not necessarily directed. In a dcpo, we write $\sup X$ for the supremum of the directed subset X, and $\lim_n s_n$ for $\sup\{s_n \mid n \in \mathbb{N}\}$ when $s_0 \leq s_1 \leq \dots$.

[4] Such as ideal completion [Plo81].

Example 2.2

$(\mathbb{R}_{\geq 0}, \leq)$ is a directed poset, but not a dcpo. $(\overline{\mathbb{R}_{\geq 0}}, \leq)$ is a dcpo.

In a poset $(S, \leq)$, we say that $X \subseteq S$ is Scott-closed if it is down-closed and if for each of its directed subsets $D \subseteq X$ that has a supremum in S, we have $\sup D \in X$.

A function f from a poset to another poset is said to be Scott-continuous if it is monotone ($x \leq y \implies f(x) \leq f(y)$) and preserves all the existing suprema of directed subsets. We now define the D-completion of a poset, which is the smallest dcpo that contains the posets.

Definition 2.2.3. *To define the D-completion of a poset $(S, \leq)$, we proceed as follows:*

- *We write $(Scott(S), \subseteq)$ for the dcpo of all Scott-closed subsets of S.*
- *We write $(D(S), \subseteq)$ for the smallest dcpo included in $(Scott(S), \subseteq)$ and containing all the $\{x \mid x \leq s\}$ for $s \in S$.*
- *We define the D-completion of $(S, \leq)$, written $(\overline{S}, \leq)$, as a chosen superposet of $(S, \leq)$ which is order-isomorphic to $(D(S), \subseteq)$.*
- *We say that $s \in \overline{S}$ is finitary if $s \in S$, and infinitary otherwise.*

We can choose such a superposet because the function $s \mapsto \{x \mid x \leq s\}$ from $(S, \leq)$ to $(D(S), \subseteq)$ induces an order-isomorphism between $(S, \leq)$ and a down-closed subset of $(D(S), \subseteq)$. We refer to [ZF10] for more details about D-completions.

Example 2.3

We have $D(\mathbb{R}_{\geq 0}) = \{[0, x] \mid x \in \mathbb{R}_{\geq 0}\} \sqcup \{\mathbb{R}_{\geq 0}\} \cong \overline{\mathbb{R}_{\geq 0}}$. This means the dcpo $(\overline{\mathbb{R}_{\geq 0}}, \leq)$ is a D-completion of the poset $(\mathbb{R}_{\geq 0}, \leq)$.

2.2.3 Positive Monoids

Definition 2.2.4. *A commutative monoid $(C, +, 0)$ is a set C together with a commutative and associative operator $+ : C \times C \to C$ and a neutral element for this operator. Such a monoid is said to be*

Cancellative *if whenever we have $x + y = x + z$, we necessarily have $y = z$.*

Positive *if whenever we have $x + y = 0$, we necessarily have $x = 0 = y$.*

Strongly Positive *if whenever we have $z + x + y = z$, we necessarily have $z + x = z = z + y$.*

Example 2.4

$(\mathbb{R}_{\geq 0}, +, 0)$ is a cancellative positive (hence strongly positive) commutative monoid. $(\overline{\mathbb{R}_{\geq 0}}, +, 0)$ is a strongly positive commutative monoid but not a cancellative one, since $+\infty + 1 = +\infty + 2$.

Note that a positive cancellative commutative monoid is necessarily strongly positive and that a strongly positive monoid is positive but *not* necessarily cancellative.

In a strongly positive commutative monoid $(C, +, 0)$ we define the induced (partial) order as the relation

$$x \sqsubseteq y \iff \exists z, x + z = y$$

While we can define this relation in every commutative monoid, it is a partial order if and only if the commutative monoid is strongly positive. We say that the strongly positive commutative monoid is directed complete when the induced order is. When it is, we can define countably infinite sums as the supremum of finite sums:

$$\sum_{i \in \mathbb{N}} c_i := \sup \left\{ \sum_{i \in I} c_i \mid I \subseteq_{\text{fin}} \mathbb{N} \right\}$$

Lemma 2.2.5. *In a directed complete strongly positive commutative monoid, infinite sums are independent of the order of summing, i.e., for every finite subsets of* $\mathbb{N}$ $I_0 \subset I_1 \subset \ldots$ *with* $\bigcup_{n \in \mathbb{N}} I_n = \mathbb{N}$*, we have*

$$\sum_{i \in \mathbb{N}} c_i = \sup \left\{ \sum_{i \in I_n} c_i \mid n \in \mathbb{N} \right\}$$

Proof. Since $\{\sum_{i \in I_n} c_i \mid n \in \mathbb{N}\}$ is included in $\{\sum_{i \in I} c_i \mid I \subseteq_{\text{fin}} \mathbb{N}\}$, the supremum of the former is lesser or equal to the supremum of the latter. We now consider $I \subseteq_{\text{fin}} \mathbb{N}$. Since $\bigcup_{n \in \mathbb{N}} I_n = \mathbb{N}$, there is a $k \in \mathbb{N}$ such that $I \subseteq I_k$. By definition of $\sqsubseteq$, it means

$$\sum_{i \in I} c_i \sqsubseteq \sum_{i \in I_k} c_i$$

Every element of $\{\sum_{i \in I} c_i \mid I \subseteq_{\text{fin}} \mathbb{N}\}$ is lesser or equal to one of $\{\sum_{i \in I_n} c_i \mid n \in \mathbb{N}\}$, which means that the supremum of the former is smaller or equal than the supremum of the latter. □

Example 2.5

$(\overline{\mathbb{R}_{\geq 0}}, +, 0)$ is a directed complete strongly positive commutative monoid.

Proposition 2.2.6. *A positive cancellative commutative monoid $(C, +, 0)$ which is directed complete is necessarily the trivial monoid $(\{0\}, +, 0)$.*

Proof. We take $c \in C$. Using the directed completion, we consider $c' = \sum_{i \in \mathbb{N}} c$. We have $c' + c = c'$, so using the cancellative property, we have $c = 0$. □

In a cancellative commutative monoid $(C, +, 0)$, we define the subtraction as the partial operator $- : C \times C \rightharpoonup C$ such that $(x - y)$ is the necessarily unique element such that $y + (x - y) = x$, if it exists. For $\epsilon_1, \ldots, \epsilon_n \in \{-, +\}$ and $x_1, \ldots, x_n \in C$, we write

$$\sum_{i=1}^{n} \epsilon_i x_i := \left(\sum_{\epsilon_i = +} x_i - \sum_{\epsilon_i = -} x_i \right) \text{ when it is defined}$$

2.2.4 Positive Cones

Definition 2.2.7. *A positive convex cone $(C, +, \cdot, 0)$ is a cancellative positive commutative monoid $(C, +, 0)$ together with an external product $\cdot : (\mathbb{R}_{\geq 0} \times C) \to C$, respecting the following equations:*

$$\begin{array}{rcl} \lambda \cdot (x + y) & = & \lambda \cdot x + \lambda \cdot y \\ \lambda \cdot 0 & = & 0 \\ (\lambda + \kappa) \cdot x & = & \lambda \cdot x + \kappa \cdot x \\ 0 \cdot x & = & 0 \\ (\lambda \times \kappa) \cdot x & = & \lambda \cdot \kappa \cdot y \\ 1 \cdot x & = & x \end{array}$$

The induced order and the subtraction partial operator of the monoid extend to the cone.

Example 2.6

For any (finite dimensional) Hilbert spaces H and K, $(\mathbf{CPM}(H, K), +, \cdot, 0)$ is a positive convex cone. The induced order $\sqsubseteq$ is the Loewner order. We note that the composition and tensor of **CPM** are linear, in other words **CPM** is a category enriched over positive convex cones.

Definition 2.2.8. *A completed positive convex cone $(C, +, \cdot, 0)$ is a directed complete strongly positive commutative monoid $(C, +, 0)$ together with an external product $\cdot : \overline{\mathbb{R}_{\geq 0}} \times C \to C$, bilinear with respect to the monoid and the additive monoid of completed non-*

negative real numbers:

$$\begin{array}{rcl} \lambda \cdot (x+y) & = & \lambda \cdot x + \lambda \cdot y \\ \lambda \cdot 0 & = & 0 \\ (\lambda + \kappa) \cdot x & = & \lambda \cdot x + \kappa \cdot x \\ 0 \cdot x & = & 0 \\ (\lambda \times \kappa) \cdot x & = & \lambda \cdot \kappa \cdot y \\ 1 \cdot x & = & x \\ \lambda \cdot \sup X & = & \sup\{\lambda \cdot x \mid x \in X\} \end{array}$$

The induced order and the infinite sums of the monoid extend to the cone.

A completed positive convex cone is *not* a positive convex cone (unless it is trivial).

Proposition 2.2.9. *The D-completion of positive convex cones for the induced order is a completed positive convex cone, where* $\infty \cdot f$ *is defined as* $\lim_n n \cdot f$.

Example 2.7

We write $\overline{\mathbf{CPM}}(H, K)$ for the D-completion of $\mathbf{CPM}(H, K)$ for the induced order. It is a completed positive convex cone. We note that the composition and tensor of **CPM** are linear and continuous, in other words **CPM** is a category enriched over completed positive convex cones. We note that if $F \subseteq \mathbf{CPM}_{\leq 1}(H, K)$, then $\sup F \in \mathbf{CPM}_{\leq 1}(H, K)$. It follows that **CPM** already contains all trace bounded suprema, in other words:

$$\forall f \in \overline{\mathbf{CPM}}(H, K), \left[f \in \mathbf{CPM}(H, K) \iff \mathbf{Tr}_H \circ f \circ \mathbf{Tr}_K^\dagger \in \mathbf{CPM}(\mathbf{1}, \mathbf{1}) \cong \mathbb{R}_{\geq 0} \right]$$

2.3 Quantum Computation

We start by giving some basic notions of quantum computation. The basic datatype used in quantum computation is the quantum bit, written **qubit** or less frequently **qbit**. The exact state of a qubit is physically inaccessible, and only a few operations are available to interact with them: (1) the preparation, which creates a qubit initialised to any classical state, (2) the measurement, which destroys a qubit and probabilistically obtains the **bit** true or false, with probabilities depending on the state of the qubit and (3) a set of operations called unitary operations, which modify the state of potentially multiple qubits at once. While we will give more intuitions later on what those unitary operations are, we can already note here that the *permutation*, which exchanges the states of two qubits, is a unitary operation, while the *duplication*, which would copy the state of a qubit to another qubit is not a unitary, and cannot be defined through any combination of creation, measurement and unitary operations.

2.3.1 A First Language for Quantum Computation

In order to be able to give examples, we introduce here MiniQ, a linear first-order fragment of the quantum λ-calculus of Section 7.1 or [PSV14]. In this language, we have data types $A, B ::= \mathbf{bit} \mid \mathbf{qubit} \mid A \otimes B$, and we have linear first-order operations. Since operations on **qubit** cannot duplicate information, we consider here that functions always consume their argument (hence the argument cannot be reused by another function). This prevention of duplication is very alike behaviours observed in linear logic, see [Gir87], as such we will borrow notations from it: we use $A \otimes B$ for pairing, later we use $A \multimap B$ for functions that can be used only once, and $!(A \multimap B)$ for functions that can be used multiple times. The terms of MiniQ are the following

- variable x, and let-binding $\mathbf{let}\ x^A = t_0\ \mathbf{in}\ t_1$,
- booleans $\mathbf{ff}$ and $\mathbf{tt}$, and conditional $\mathbf{if}\ t_0\ \mathbf{then}\ t_1\ \mathbf{else}\ t_2$,
- pairing $t \otimes s$ and destruction $\mathbf{let}\ x^A \otimes y^B = t_0\ \mathbf{in}\ t_1$,
- measurement $\mathbf{meas}\ t_0$, creation $\mathbf{new}\ t_0$ and unitary operations $\mathbf{U}\ t_0$.

Where variables are taken in a set of variables $\mathbb{V}$, terms being considered up to the usual renaming of variables through α-equivalence. The three quantum primitives we allow are **new** which takes a **bit** and creates a **qubit** initialised according to the value of the **bit**, **meas** which measures[5] a **qubit** and returns a **bit** according to the result of the measurement, and **U** which range over every unitary matrix and represent the corresponding unitary operation on quantum data. We recall that unitary matrices are square matrices of complex numbers that are invertible, with $U^{-1} = U^\dagger$.

We write typing rules with the usual syntax $\Gamma \vdash t : A$, with the typing context Γ being a sequence of typed variables $(x : B)$. Typing is derived from the rules in Table 2.1.

In all those rules, we ensure that no variable appears twice in the context. Similarly to Λ, LΛ and AΛ, in AMiniQ, all the variables of the context must be used at most once in the term and in LMiniQ, we exclude the weakening rule (at the top left of Table 2.1), meaning that all variables of the context must be used exactly once.

To give a semantics to this very simple language, we need a mathematical representation of quantum computation. There are two levels at which we can consider quantum computation: as acting on *pure states*, or on *mixed states*. These two are analogous to respectively classical states and probabilistic states: the first representation encompasses a single possibility, while the second one encompasses a probabilistic sum of outcomes. We start by defining pure states, which can be used to define an operational semantics for

[5] In real experiments, results of a measurement depends on a basis used for the measurement. We assume a canonical basis is chosen and used. Measures in any other basis can be simulated by applying a well-chosen unitary and then measuring in the canonical basis.

Structural Rules

$$\frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, y : B, x : A, \Delta \vdash t : C}\ \text{permutation} \qquad \frac{\Gamma \vdash_{\mathbb{A}} t : B \qquad x \notin \mathrm{FV}(t)}{\Gamma, x : A \vdash_{\mathbb{A}} t : B}\ \text{(AMiniQ only) weakening}$$

Variables

$$\frac{}{x : A \vdash x : A}\ \text{axiom} \qquad \frac{\Gamma \vdash t_0 : A \qquad x : A, \Delta \vdash t_1 : B}{\Gamma, \Delta \vdash \mathbf{let}\ x^A = t_0\ \mathbf{in}\ t_1 : B}\ \text{let}$$

Tensor type

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash s : B}{\Gamma, \Delta \vdash t \otimes s : A \otimes B}\ \text{pair} \qquad \frac{\Gamma \vdash t : A \otimes B \qquad x : A, y : B, \Delta \vdash s : C}{\Gamma, \Delta \vdash \mathbf{let}\ x \otimes y = t\ \mathbf{in}\ s : C}\ \text{let-pair}$$

Boolean type

$$\frac{}{\vdash \mathbf{ff} : \mathbf{bit}}\ \text{false} \qquad \frac{}{\vdash \mathbf{tt} : \mathbf{bit}}\ \text{true} \qquad \frac{\Gamma \vdash t_0 : \mathbf{bit} \qquad \Delta \vdash t_1 : A \qquad \Delta \vdash t_2 : A}{\Gamma, \Delta \vdash \mathbf{if}\ t_0\ \mathbf{then}\ t_1\ \mathbf{else}\ t_2 : A}\ \text{conditional}$$

Quantum type

$$\frac{\Gamma \vdash t : \mathbf{qubit}}{\Gamma \vdash \mathbf{meas}\ t : \mathbf{bit}}\ \text{meas} \qquad \frac{\Gamma \vdash t : \mathbf{bit}}{\Gamma \vdash \mathbf{new}\ t : \mathbf{qubit}}\ \text{new}$$

$$\frac{\Gamma \vdash t : \mathbf{qubit}^{\otimes n} \qquad U \text{ unitary of size } 2^n}{\Gamma \vdash \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}}\ \text{unitary}$$

Table 2.1: Typing Rules for MiniQ

MiniQ. We then define mixed states, which can be used to define a denotational semantics for MiniQ. We will not detail here these operational and denotational semantics, and will instead directly follow with the extension of MiniQ with higher order, which we call QΛ. One can recover an operational semantics and a denotational semantics for MiniQ by considering the ones of QΛ in Section 3.1.3 and Section 3.2, and simply restricting them to the syntax of MiniQ.

2.3.2 Pure States

In the pure state approach, the state of a qubit is represented as a normalised vector $q \in \mathbb{C}^2$ (with $||q|| = 1$). Similarly, the state of a collection of n qubits is represented as a normalised vector $q \in \mathbb{C}^{2^n}$. For convenience, we use the following shorthand for these Hilbert spaces

$$\mathbf{1} = \mathbb{C} \qquad \mathfrak{Q} = \mathbb{C}^2 \qquad \mathfrak{Q}^{\otimes n} = \mathbb{C}^{2^n}$$

We use the usual bra-ket notation from quantum physics for them, hence:

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix} \in \mathfrak{Q}^{\otimes 2} \text{ is written } a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$$

We note that $|a_{ij}|^2$ in $[0, 1]$, and can be understood as the probability of obtaining ij when fully measuring the state. Some simple examples of quantum states are:

- The trivial quantum states $|0\rangle$ and $|1\rangle$. If measured[6], they will return with probability one a single result, respectively false and true.
- A fair "quantum coin" $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, which is a quantum superposition of the two trivial states. If measured, the results will be false or true with probability 1/2 each.
- Another fair "quantum coin" $\frac{\mathbf{i}}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$, which also yields false or true with probability 1/2 each when measured, but can be distinguished from the previous one if we apply a well-chosen unitary operation to both before measuring them.
- A maximally entangled pair $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, where if we independently measure each of the two qubits, we are guaranteed to obtain the same result on both sides.

Since operations on **qubit** cannot duplicate quantum information, we consider the input of an operation on quantum data is always consumed, and cannot be used elsewhere. Hence the operation of "doing nothing" will be represented by the identity function from $\mathfrak{Q}$ to $\mathfrak{Q}$. In fact operations from a state containing n qubits to a state containing m qubits can

[6] We recall that we only consider measurements in a canonical basis.

be represented by a map in $\mathbf{Hilb}(\mathfrak{Q}^{\otimes n}, \mathfrak{Q}^{\otimes m})$. We recall that $\mathbf{Hilb}_{\leq 1}$ informally contains "all the quantum operations that can be physically realised", and we can here refine that statement:

Proposition 2.3.1. *We write* $\mathbf{Hilb}^2_{\leq 1}$ *for* $\mathbf{Hilb}_{\leq 1}$ *restricted to objects of dimension a power of two. The category* $\mathbf{Hilb}^2_{\leq 1}$ *is the smallest sub-SMC containing all the following morphisms,*

- *The creation maps* $\mathbf{new}^{\mathbf{Hilb}}_{\mathbf{ff}} : z \mapsto z|0\rangle$ *and* $\mathbf{new}^{\mathbf{Hilb}}_{\mathbf{tt}} : z \mapsto z|1\rangle$*, both in* $\mathbf{Hilb}(\mathbf{1}, \mathfrak{Q})$*.*
- *The measurement maps* $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{ff}} : \alpha|0\rangle + \beta|1\rangle \mapsto \alpha$ *and* $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{tt}} : \alpha|0\rangle + \beta|1\rangle \mapsto \beta$*, both in* $\mathbf{Hilb}(\mathfrak{Q}, \mathbf{1})$*.*
- *Any unitary map* $\mathbf{U} \in \mathbf{Hilb}(A, B)$*, i.e., such that* $\mathbf{U} \circ \mathbf{U}^\dagger = \mathbf{id}^{\mathbf{Hilb}}_B$ *and* $\mathbf{U}^\dagger \circ \mathbf{U} = \mathbf{id}^{\mathbf{Hilb}}_A$*. Note that it implies* $\dim(A) = \dim(B)$*.*

This follows from Lemma 6.13 of [Sel04].

Morphisms of $\mathbf{Hilb}_{\leq 1}$ that are norm-preserving can directly be thought of as operations from quantum states to quantum states. However, morphisms of $\mathbf{Hilb}_{\leq 1}$ are in general norm non-increasing, which can be thought of as a probability to fail. In particular, $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{ff}}$ physically corresponds to "measuring a qubit, and assuming the result is false", while $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{tt}}$ is the dual operation. It will be practical to step outside of $\mathbf{Hilb}_{\leq 1}$ when using the measurement, so that we can talk about both outputs at once. We define k-$\mathbf{meas}^{\mathbf{Hilb}}$ which measure the k-th qubit of a (normalised) state and returns the non-normalised quantum state obtained from $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{ff}}$, and the non-normalised quantum state obtained from $\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{tt}}$.

$$k\text{-}\mathbf{meas}^{\mathbf{Hilb}} : \begin{array}{ccc} \mathfrak{Q}^{\otimes n}_{||-||=1} & \to & \mathfrak{Q}^{\otimes n-1} \times \mathfrak{Q}^{\otimes n-1} \\ \text{initial state} & \mapsto & \binom{\text{final state if } \mathbf{ff}}{\text{final state if } \mathbf{tt}} \\ q & \mapsto & \Big(k\text{-}\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{ff}}(q), k\text{-}\mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{tt}}(q)\Big) \end{array}$$

$$\text{where } k\text{-}\mathbf{meas}^{\mathbf{Hilb}}_b := \mathfrak{Q}^{\otimes k-1} \otimes \mathbf{meas}^{\mathbf{Hilb}}_b \otimes \mathfrak{Q}^{\otimes n-k}$$

The Biased Coin

As a first illustration, we will explain how to simulate a coin with probability p of landing on heads (*i.e.*, $\mathbf{tt}$) and $1-p$ for tails (*i.e.*, $\mathbf{ff}$). The protocol is very simple.

1. Create a qubit using $\mathbf{new}^{\mathbf{Hilb}}_{\mathbf{ff}}$. The current state is $|0\rangle$.
2. Apply the unitary operation of corresponding matrix $\begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix}$. The resulting state is $\sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle$.

3. Measure the qubit. The result is **ff** with probability $1-p$ and **tt** with probability p.

Using the MiniQ language, we can also write it as the term:

$$\vdash_{\mathbb{L}} \mathrm{Coin}_p() : \mathbf{bit}$$

$$\mathrm{Coin}_p() := \mathbf{meas}\ \left(\begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix}\ (\mathbf{new}\ \mathbf{ff})\right)$$

The Bell States Protocol

To illustrate the quantum effects, we will take the well-known protocol of creating Bell states. This protocol is also known under the name of the Einstein-Podolsky-Rosen protocol, or EPR. The Bell states are the following four maximally entangled pairs of qubits:

$$\begin{array}{ll} \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle & \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle \\ \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle & \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \end{array}$$

If we take the first of those states as an example, and measure its first qubit, then the result of the measurement will be true or false with probability 1/2 each. If the result is true then we know that the state of the second qubit is $|1\rangle$, whereas if it is false we know that the state of the second qubit is $|0\rangle$. One notable physical property here is that the entanglement between those two qubits holds whatever the distance between them, as such, the Bell states have a central role in quantum cryptography where each of the two qubits can be owned by a different agent.

The Bell States protocol takes as an input a pair of booleans, and creates one of the four Bell states according to the value of those booleans. On an input (b, b'), the protocol is the following:

1. Create two qubits using $\mathbf{new}_b^{\mathbf{Hilb}}$ and $\mathbf{new}_{b'}^{\mathbf{Hilb}}$. On input $(\mathbf{ff}, \mathbf{ff})$ the current state would be $|00\rangle$.

2. Apply to the first qubit the Hadamard unitary, associated to the matrix $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. On input $(\mathbf{ff}, \mathbf{ff})$ the current state would be $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$.

3. Apply to both the controlled-not unitary operation, associated to the matrix $Nc = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$. On input $(\mathbf{ff}, \mathbf{ff})$ the current state would be $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$.

Using the MiniQ language, we can also write it as the term:

$$b : \mathbf{bit}, b' : \mathbf{bit} \vdash_{\mathbb{L}} \mathrm{Bell}(b, b') : \mathbf{qubit} \otimes \mathbf{qubit}$$

$$\mathrm{Bell}(b, b') := \mathbf{let}\ q_1 \otimes q_2 = (\mathbf{new}\ b) \otimes (\mathbf{new}\ b')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in\ Nc}\ (q_3 \otimes q_2)$$

The Bell Measure Protocol

Another simple example is the Bell measurement, which allows us to recover from a maximally entangled pair which pair of booleans created it. In other words, on an input created by $\mathrm{Bell}(b, b')$, the Bell measurement will return (b, b') with probability one. On an input (q, q'), the protocol is the following:

1. Apply to both the controlled-not unitary operation, associated to the matrix Nc. On input $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ the current state would be $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$.

2. Apply to the first qubit the Hadamard unitary, associated to the matrix H. On input $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ the current state would be $|00\rangle$.

3. Measure both qubits, using **meas** twice. On input $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ the current state would be $(\mathbf{ff}, \mathbf{ff})$ with probability 1, and probability 0 for the three other possibilities.

Using the MiniQ language, we can also write it as the term:

$$q : \mathbf{qubit}, q' : \mathbf{qubit} \vdash_{\mathbb{L}} \mathrm{BellM}(q, q') : \mathbf{bit} \otimes \mathbf{bit}$$

$$\mathrm{BellM}(q, q') := \mathbf{let}\ q_1 \otimes q_2 = \mathbf{Nc}\ (q \otimes q')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in}\ (\mathbf{meas}\ q_3) \otimes (\mathbf{meas}\ q_2)$$

The Quantum Teleportation Protocol

Finally, we give a less simple example: the quantum teleportation protocol. In this protocol, an agent has a qubit he wishes to transmit to another agent, only using classical communication methods and a previously shared quantum state. For that, he will create a pair of booleans from his qubit, send them to the other agent, and the other agent will be able to recreate this exact qubit from those two booleans.

This protocol requires a setup phase, where the two agents meet to create some Bell states. Indeed, for each teleportation the agents will consume one entangled pair of qubits. The full protocol is the following:

1. The agents Alice and Bob use $\mathrm{Bell}(\mathbf{ff}, \mathbf{ff})$ to obtain a pair of qubits q_A and q_B. They each take one of the qubits.

2. Alice has a qubit q she wishes to communicate. For that, she computes $\mathrm{BellM}(q_A, q)$, then send the results to Bob.

3. Bob receives two bits (b, b'), and applies to q_B one unitary $U_{b,b'}$ with:

$$U_{\mathbf{ff},\mathbf{ff}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad U_{\mathbf{ff},\mathbf{tt}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad U_{\mathbf{tt},\mathbf{ff}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad U_{\mathbf{tt},\mathbf{tt}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

At the end of this protocol, Bob has a qubit of identical behaviour as the (now destroyed) qubit q. This includes possible entanglements of q with other qubits, which are also preserved by this "teleportation". Mathematically, the composition of all those operations gives the identity map. As the protocol has multiple branching points depending on the result of measurements, with the tools we have defined, it is for the moment unclear how to actually compute this composition. But we will come back to this example in Section 3.2.2 once we will have introduced the necessary tools.

2.3.3 Mixed States

Mixed states can be understood as subprobability distributions of pure states, where subprobability distributions that are indistinguishable are represented by the same mixed state.

In the mixed state approach, the state of a qubit is represented by a subdensity operator $q \in \mathrm{Pos}_{\leq 1}(\mathfrak{Q})$. Similarly, the state of a collection of n qubits is represented by a subdensity operator $q \in \mathrm{Pos}_{\leq 1}(\mathfrak{Q}^{\otimes n})$. We will often use the matrix $\mathcal{M}(q)$ (for the canonical basis) instead of q. The general form of a qubit becomes:

$$\begin{pmatrix} a & b \\ \overline{b} & c \end{pmatrix} \text{ with } \begin{cases} a, c \in \mathbb{R}_{\geq 0} \\ a + c \leq 1 \end{cases} \quad \text{and} \quad \begin{cases} b \in \mathbb{C} \\ |b|^2 \leq a \cdot c \end{cases}$$

Note that $a+c$ can be less than one. This happens when the computation has a probability of failing or diverging, never returning a result.

Pure states appear as a particular case of mixed states, the pure state $a|0\rangle + b|1\rangle$ corresponding to $\begin{pmatrix} |a|^2 & a\overline{b} \\ \overline{a}b & |b|^2 \end{pmatrix}$. More generally the embedding is the following:

$$\begin{array}{rccc} (\!|_|\!) : & H & \to & \mathrm{Pos}(H) \\ & q & \mapsto & (q' \mapsto \langle q|q'\rangle \cdot q) \\ & \mathcal{M}(q) & \mapsto & \mathcal{M}(q) \times \mathcal{M}(q)^\dagger \end{array}$$

We said earlier that mixed states are subprobability distributions of pure states, up to indistinguishability. We formalise it in the following theorem. Note that this theorem relies on the operational semantics of LMiniQ that we have yet to define. Since LMiniQ is a fragment of the LQΛ language defined in next chapter, we simply refer to Section 3.1.3 for the operational semantics of LMiniQ.

Theorem 2.3.2. *For every mixed state $m \in \mathrm{Pos}_{\leq 1}(\mathfrak{Q}^{\otimes n})$, there exist some pure states $q_i \in \mathfrak{Q}^{\otimes n}$ and some probabilities p_i such that:*

$$m = \sum_i p_i (\!|q_i|\!)$$

Two subprobability distribution of quantum states $\{(p_i, q_i) \mid i \in I\}$ *and* $\{(p'_j, q'_j) \mid j \in J\}$ *cannot be distinguished through terms of LMiniQ, with operational semantics defined in Section 3.1.3, if and only if*

$$\sum_i p_i (\!| q_i |\!) = \sum_j p_j (\!| q_j |\!)$$

Proof. For the first part, we simply diagonalise $\mathcal{M}(m)$ into $SDS^\dagger$ with D being a diagonal matrix of coefficients d_i all in $[0, 1]$, of sum lesser or equal to 1. We remark that $\mathcal{M}(m) = \sum_i d_i (Se_i)(Se_i)^\dagger$ with e_i the i-th vector of the canonical basis of $\mathfrak{Q}^{\otimes n}$. We take q_i such that $\mathcal{M}(q_i) = Se_i$ and we have $m = \sum_i d_i (\!| q_i |\!)$. As a direct consequence of the full abstraction Theorem 3.2.14, we obtain that two subprobability distributions of quantum states cannot be distinguished through terms of LQΛ if and only if they have the same mixed states. □

Operations from a state with n qubits to a state with m qubits will be represented by maps in $\mathbf{CPM}(\mathfrak{Q}^{\otimes n}, \mathfrak{Q}^{\otimes m})$. For every map f in $\mathbf{Hilb}_{\leq 1}(\mathfrak{Q}^{\otimes n}, \mathfrak{Q}^{\otimes m})$, we can lift it to $\mathbf{CPM}_{\leq 1}$ using the functorial extension of $(\!|_|\!)$ as follows:

$$\begin{array}{rccc} (\!| f |\!) : & \mathrm{Pos}_{\leq 1}(\mathfrak{Q}^{\otimes n}) & \to & \mathrm{Pos}_{\leq 1}(\mathfrak{Q}^{\otimes m}) \\ & q & \mapsto & f \circ q \circ f^\dagger \end{array}$$

In particular, we have

$$\mathbf{new}^{\mathbf{CPM}}_{\mathbf{ff}} = (\!| \mathbf{new}^{\mathbf{Hilb}}_{\mathbf{ff}} |\!) : z \mapsto z \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \mathbf{new}^{\mathbf{CPM}}_{\mathbf{tt}} = (\!| \mathbf{new}^{\mathbf{Hilb}}_{\mathbf{tt}} |\!) : z \mapsto z \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{meas}^{\mathbf{CPM}}_{\mathbf{ff}} = (\!| \mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{ff}} |\!) : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a \quad \mathbf{meas}^{\mathbf{CPM}}_{\mathbf{tt}} = (\!| \mathbf{meas}^{\mathbf{Hilb}}_{\mathbf{tt}} |\!) : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto d$$

Chapter 3

Relational Model for the Linear Quantum λ-calculus

3.1 The Linear Quantum λ-calculus

We define here the language QΛ, which is both an extension of Λ with quantum primitives, and MiniQ with higher order. Like them, it has a strict variant LQΛ and an affine variant AQΛ, with for only difference the presence of the weakening typing rule. We use $\vdash_{\mathbb{L}}$ and $\vdash_{\mathbb{A}}$ for their respective typing judgements, and $\vdash$ for definitions and propositions that apply to both. The types are the types of Λ extended with quantum bits. We also add lists as an example of how to handle recursive types. We recall that lists A^ℓ of elements of type A are the smallest solution of the recursive equation $X = \mathbf{1} \oplus (A \otimes X)$.

$$A, B ::= \mathbf{1} \mid A \multimap B \mid A \oplus B \mid A \otimes B \mid \mathbf{qubit} \mid A^\ell$$

To the terms of QΛ, we add the following:

- the quantum primitives **meas**, **new** and **U** (for any unitary operation U),
- the list operations **fold** and **unfold**.

The typing rules are the same as Λ, augmented by the following ones. The defined typing system still respects the uniqueness of typing.

Theorem 3.1.1 (Uniqueness of Typing)**.** *For Γ a typing context and t a term, with $FV(t) \subseteq \Gamma$, there exists at most one type A such that the typing judgement $\Gamma \vdash t : A$ is valid.*

3.1.1 Quantum Primitives

Typing Rules:

$$\frac{\Gamma \vdash t : \mathbf{bit}}{\Gamma \vdash \mathbf{new}\ t : \mathbf{qubit}}\ \text{new} \qquad \frac{\Gamma \vdash t : \mathbf{qubit}}{\Gamma \vdash \mathbf{meas}\ t : \mathbf{bit}}\ \text{meas}$$

$$\frac{\Gamma \vdash t : \mathbf{qubit}^{\otimes n} \qquad U \text{ of arity } n}{\Gamma \vdash \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}}\ \text{unitary}$$

Syntactic Sugar:

$$\begin{array}{lcll} p \cdot t + (1-p) \cdot s & := & \mathbf{if\ meas} \begin{pmatrix} \sqrt{1-p} & \sqrt{1-p} \\ \sqrt{p} & -\sqrt{p} \end{pmatrix} (\mathbf{new\ ff})\ \mathbf{then}\ t\ \mathbf{else}\ s & \\ p \cdot t & := & p \cdot t + (1-p) \cdot \bot & \\ \sum_{i=1}^{n} p_i \cdot t_i & := & p_1 \cdot t_1 + \sum_{i=2}^{n} \frac{p_i}{1-p_1} \cdot t_i & \text{with } \frac{0}{0} = 0 \end{array}$$

We note that we are able to define a term $\mathbf{destr_{qubit}} := \lambda q.\mathbf{destr_{bit}}\ (\mathbf{meas}\ q)$, hence even in the LQΛ, the strictness of the linearity can be relaxed for **qubit**, *i.e.*, we can discard qubit without using them. However, we do not have a term $\mathbf{dupl_{qubit}}$, as quantum data cannot be duplicated[1].

3.1.2 Lists

Typing Rules:

$$\frac{\Gamma \vdash t : A^{\ell}}{\Gamma \vdash \mathbf{unfold}\ t : \mathbf{1} \oplus (A \otimes A^{\ell})}\ \text{unfold} \qquad \frac{\Gamma \vdash t : \mathbf{1} \oplus (A \otimes A^{\ell})}{\Gamma \vdash \mathbf{fold}\ t : A^{\ell}}\ \text{fold}$$

Syntactic Sugar:

$$\begin{array}{rcl} [\,] & := & \mathbf{fold}\ (\mathbf{inj}_{\ell}\ ()) \\ t :: s & := & \mathbf{fold}\ (\mathbf{inj}_{r}\ (t \otimes s)) \\ \mathbf{match}\ t\ \mathbf{with} \begin{array}{|lcl} [\,] & \mapsto & s_1 \\ x :: y & \mapsto & s_2 \end{array} & := & \delta\ (\mathbf{unfold}\ t,\ z.z; s_1,\ z'.\mathbf{let}\ x \otimes y = z'\ \mathbf{in}\ s_2) \end{array}$$

We note that the impossibility to erase (resp. duplicate) A^{ℓ} even whenever A is erasable (resp. duplicable) is more a syntactic restriction than a fundamental one. In the extension QΛ! (see Section 7.1), those are definable using recursion.

[1] More precisely, we do not have a comonoid for **qubit**, *i.e.*, we cannot define a duplication term such that both "duplicating a **qubit** and then destroying the left hand side **qubit**" and "duplicating a **qubit** and then destroying the right hand side **qubit**" are observationally equivalent to the identity.

$$
\begin{array}{lcll}
() \ ; \ t & \to & t & \\
(\lambda x.t)\ v & \to & t\{x \leftarrow v\} & \\
\mathbf{let}\ x \otimes y = v \otimes w\ \mathbf{in}\ t & \to & t\{x \leftarrow v, y \leftarrow w\} & \\
\delta\ (\mathbf{inj}_\ell\ v,\ x.t,\ y.s) & \to & t\{x \leftarrow v\} & \\
\delta\ (\mathbf{inj}_r\ v,\ x.t,\ y.s) & \to & s\{y \leftarrow v\} & \\
\mathbf{unfold}\ (\mathbf{fold}\ v) & \to & v & \\
E[t] & \to & E[s] & \text{whenever} \quad t \to s \\
E[\bot_V] & \to & \bot_{\mathrm{FV}(E[\bot_V])} & \text{whenever} \quad E[_] \neq _
\end{array}
$$

$$
\begin{array}{lcllll}
\text{with } E[_] & ::= & _ & \mid\ E[_]\ ;\ t & & \\
& \mid & E[_]\ t & \mid\ v\ E[_] & & \\
& \mid & E[_] \otimes t & \mid\ v \otimes E[_] & \mid\ \mathbf{let}\ x \otimes y = E[_]\ \mathbf{in}\ t & \\
& \mid & \mathbf{inj}_\ell\ E[_] & \mid\ \mathbf{inj}_r\ E[_] & \mid\ \delta\ (E[_],\ x.t_1,\ y.t_2) & \\
& \mid & \mathbf{fold}\ E[_] & \mid\ \mathbf{unfold}\ E[_] & & \\
& \mid & \mathbf{meas}\ E[_] & \mid\ \mathbf{new}\ E[_] & \mid\ \mathbf{U}\ E[_] &
\end{array}
$$

Table 3.1: Reduction Rules for QΛ terms

3.1.3 Operational Semantics

We first extend the reduction system of Λ to QΛ by adding reductions for list folding and unfolding. We start by defining the values of QΛ as $v, w ::= () \mid x \mid \lambda x.t \mid \mathbf{inj}_\ell\ v \mid \mathbf{inj}_r\ v \mid v \otimes w \mid \mathbf{fold}\ v$. The reduction rules are given by Table 3.1.

We do not specify any reduction rules for the quantum primitives yet. Indeed, quantum variables can be entangled with one another, hence locally describing their value is not enough, we need a global store to describe the quantum effects that link the different quantum variables. This will lead us to add a global quantum state tracked along the reduction sequence of a term, but before tackling this problem, we will recall some standard properties of the reduction system we just defined.

This reduction system satisfies subject reduction, is deterministic (hence confluent) and is normalising. Closed terms that do not contain quantum primitives satisfy progress. More precisely, the following properties hold.

Proposition 3.1.2.

Subject Reduction *If* $\Gamma \vdash t : A$ *and* $t \to s$ *then* $\Gamma \vdash s : A$.

Determinism *For any term* t*, there exists at most one term* s *such that* $t \to s$.

Normalisation *For any term* t*, there is no infinite sequence* $t \to t_1 \to t_2 \to \dots$.

Partial Progress *For any closed term* $\vdash t : A$ *which does not contain* $\mathbf{meas}, \mathbf{new}, \mathbf{U}$ *, either* t *is a value, or* $\bot$*, or there exists a term* s *such that* $t \to s$.

We now define the concept of closure, which is a term together with a global memory state tracking the values of the different quantum variables, including the entanglement between them. This notion is sometimes called "configuration", however we reserve this name for configurations of event structures in Section 4.2.

Definition 3.1.3. *A simple (quantum) closure is a triple $[q, \ell, t]$ where:*

- *t is a term.*
- *q is the quantum store, $q \in \mathfrak{Q}^{\otimes n}$ for some $n \in \mathbb{N}$. We write $|q| = n$.*
- *ℓ is a sequence of n variables written $|x_1 \dots x_n\rangle$. It is an ordering of all the free variables of t, and it can be seen as a function which to each free variable of the term indicates where its value is stored.*

We say that the simple closure is terminal *if t is a value or $\bot$. We write $\vdash [q, \ell, t] : A$ whenever*

$$x_1 : \mathbf{qubit}, \dots, x_n, \mathbf{qubit} \vdash t : A$$

For simplicity of notation, we assumed that every free variable of the term is bound by the store, hence closures are always closed.

The second subtlety in the definition of this semantics is that reductions are probabilistic. Indeed, the measurements **meas** lead to two different results, each of them with a different probability. There are two different ways of handling probabilistic reduction systems. The first one is to annotate the reductions by a probability, for example "$t \to_{0.5} s_1$ and $t \to_{0.5} s_2$", and the second is to use probabilistic sums, for example "$t \to \frac{1}{2}s_1 + \frac{1}{2}s_2$". For most practical uses, those two approaches are equivalent. We choose the second approach.

Definition 3.1.4. *A closure c is a discrete probability distribution of simple closures, written as a formal sum $\sum_i p_i c_i$ with c_i simple closures, $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. We say it is terminal if all the c_i are terminal. We write $\vdash \sum_i p_i c_i : A$ if $\vdash c_i : A$ for every simple closure c_i.*

As the formal sum $\sum_i p_i c_i$ is just the representation of a discrete probability distribution, it follows that $\sum$ is associative, commutative, and has 0 for neutral. We describe in Table 3.2 the operational semantics of quantum closures.

Proposition 3.1.5. *This reduction system on closures*

- *satisfies subject reduction,*
- *is deterministic (hence confluent),*
- *is normalising,*

$$\frac{t_0 \to t_1}{[q, \ell, E[t_0]] \to [q, \ell, E[t_1]]}$$

$$\frac{}{[q, |x_1 \ldots x_n\rangle, E[\mathbf{new\ ff}]] \to [q \otimes |0\rangle, |x_1 \ldots x_{n+1}\rangle, E[x_{n+1}]]}$$

$$\frac{}{[q, |x_1 \ldots x_n\rangle, E[\mathbf{new\ tt}]] \to [q \otimes |1\rangle, |x_1 \ldots x_{n+1}\rangle, E[x_{n+1}]]}$$

$$\frac{k\text{-}\mathbf{meas}^{\mathbf{Hilb}}(q) = (q_0, q_1)}{[q, |x_1 \ldots x_n\rangle, E[\mathbf{meas}\ x_k]] \to \begin{array}{lll} & ||q_0||^2 & \left[\frac{q_0}{||q_0||}, |x_1 \ldots x_{k-1} x_{k+1} \ldots x_n\rangle, E[\mathbf{ff}]\right] \\ + & ||q_1||^2 & \left[\frac{q_1}{||q_1||}, |x_1 \ldots x_{k-1} x_{k+1} \ldots x_n\rangle, E[\mathbf{tt}]\right] \end{array}}$$

$$\frac{\hat{\sigma} \circ \left(U \otimes \mathfrak{Q}^{\otimes n-k}\right) \circ \hat{\sigma}^{-1}(q) = q' \qquad \hat{\sigma} \text{ permutation map induced by } \sigma \in \mathfrak{S}_n}{[q, |x_1 \ldots x_n\rangle, E[\mathbf{U}\ (x_{\sigma(1)} \otimes \ldots \otimes x_{\sigma(k)})]] \to [q', |x_1 \ldots x_n\rangle, E[x_{\sigma(1)} \otimes \ldots \otimes x_{\sigma(k)}]]}$$

$$\frac{c \to c' \quad d \text{ terminal} \quad 0 < p \leq 1}{pc + (1-p)d \to pc' + (1-p)d} \qquad \frac{c \to c' \quad d \to d' \qquad 0 \leq p \leq 1}{pc + (1-p)d \to pc' + (1-p)d'}$$

Table 3.2: Reduction Rules for QΛ closures

- *and closures always progress, where value closures are defined as probability distributions of closures whose term part are values or* $\bot$.

We emphasise that the determinism is at the level of closures, *i.e.*, probability distributions of simple closures. The reduction system describes non-deterministic behaviours by encapsulating them in a probability distribution.

We extend the notion of convergence and observational equivalence from Λ, taking into account that our reductions are now probabilistic.

Definition 3.1.6 (Convergence)**.** *For c a closure and v a value, we define the probability that c converges to v, written $\mathbb{P}(c \Downarrow v)$, as the supremum of the $p \in [0,1]$ such that $p = \sum_{i=1}^{n} p_i$ and $c \to^* \sum_{i=1}^{n} p_i[q_i, \ell_i, v] + (1-p)c'$ with c' any closure.*

By strong normalisation, the supremum is always reached. Whenever $\vdash c : \mathbf{1}$, we write $\mathbb{P}(c \Downarrow)$ for $\mathbb{P}(c \Downarrow ())$. Whenever $\varnothing \vdash t : \mathbf{1}$, we write $\mathbb{P}(t \Downarrow)$ for $\mathbb{P}([\varnothing, \varnothing, t] \Downarrow)$.

An observation context $\mathcal{O}[_]$ for $\Gamma \vdash A$ with Γ a typing context and A a type, is a term with a unique hole $\mathcal{O}[_]$ such that for every $\Gamma \vdash t : A$, we have $\vdash \mathcal{O}[t] : \mathbf{1}$.

We say that two terms $\Gamma \vdash t_1 : A$ and $\Gamma \vdash t_2 : A$ are observationally equivalent, and we write $t_1 =_{\mathrm{obs}}^{\Gamma \vdash A} t_2$, if for every observation context $\mathcal{O}[_]$ for $\Gamma \vdash A$, we have

$$\mathbb{P}(\mathcal{O}[t_1] \Downarrow) = \mathbb{P}(\mathcal{O}[t_2] \Downarrow)$$

We will keep the annotation $\Gamma \vdash A$ implicit in $=_{\mathrm{obs}}^{\Gamma \vdash A}$.

3.2 The Linear Quantum Relational Model

We want to define a denotational semantics for QΛ. The model presented here is a reformulation of the model of Selinger and Valiron in [SV08], with lists requiring an infinitary completion as in their later paper [PSV14] with Pagani. Rather than presenting the model as a generalisation of **CPM** with coproducts, we formulate it as a variant of the weighted relational model **WRel** using **CPM** annotations instead of probabilistic annotations. We note that this is a model of LQΛ, but we believe it could be tweaked into a model for AQΛ.

3.2.1 Definition of the Model

We follow the paradigm of quantum data over classical control flow, so rather than trying to integrate quantum data within the existing framework, *e.g.*, having the web for **qubit** list all the possible values for **qubit** similarly to how the web for **bit** lists $\{\mathbf{tt}, \mathbf{ff}\}$, we keep the classical framework unchanged and quantum data over it with the use of "quantum annotations". We define the category **QRel** as follows:

- The objects are pairs $A = (|A|, \mathcal{H}_A)$ with $|A|$ a finite or countable set, called the web of A, and $\mathcal{H}_A(a)$ a finite dimensional Hilbert space for every element $a \in |A|$.

- The morphisms from an object A to an object B are the functions f that map $(a,b) \in |A| \times |B|$ to an annotation $f(a,b) \in \overline{\mathbf{CPM}}(\mathcal{H}_A(a), \mathcal{H}_B(b))$, which is the D-completion of the positive convex cone $\mathbf{CPM}(\mathcal{H}_A(a), \mathcal{H}_B(b))$.

- The identity on $(|A|, \mathcal{H}_A)$ has the annotation $\mathbf{id}_A(a,a) = \mathbf{id}^{\mathbf{CPM}}_{\mathcal{H}_A(a)}$ for every $a \in |A|$ and $\mathbf{id}_A(a,b) = 0$ when $a \neq b$.

- The composition is the relational composition:

$$(g \circ f)(a,c) := \sum_{b \in |B|} g(b,c) \circ f(a,b)$$

  We note that this sum might be infinite, hence the need for the D-completion.

Assuming a morphism $f \in \mathbf{QARel}(A,B)$ represent a programs taking inputs described by A and producing outputs described by B, the morphism f read as: if the state of the input is a classical state $a \in |A|$ together with a quantum state $q \in \mathrm{Pos}(\mathcal{H}_A(a))$, and the state of the output is $b \in |B|$ together with $q' \in \mathrm{Pos}(\mathcal{H}_B(b))$, then $f(a,b)(q) = q'$.

We note the similarity between **QRel** and the previously mentioned categories of weighted relations **WRel** where the composition was

$$(g \circ f)(a,c) := \sum_{b \in |B|} g(b,c) \times f(a,b)$$

and of relation **Rel** where the composition was

$$(a,c) \in (g \circ f) : \ \exists b \in |B|, (b,c) \in g \text{ and } (a,b) \in f$$

In fact, we have a full and faithful functor from **WRel** to **QRel** which send an object A to the pair $(A, \mathcal{H} : a \mapsto \mathbf{1})$ and a morphism R to $f(a,b) := R(a,b) \cdot \mathbf{id_1}$.

While we will prove in Proposition 3.2.12 that we never use the elements added by the D-completion when interpreting LQΛ, we cannot avoid using it in the definition of **QRel**, as the composition involves infinite sums which, a priori, have no reason to converge without the completion. The presence of lists in our language is the only source of objects with infinite web, hence of infinite sums in the composition.

Theorem 3.2.1. **QRel** *is a distributive CpCC which is non-trivial and has a bottom:*

- *The monoidal product $\otimes$ on objects is simply given by $(|A|, \mathcal{H}_A) \otimes (|B|, \mathcal{H}_B) := (|A| \times |B|, (a,b) \mapsto \mathcal{H}_A(a) \otimes \mathcal{H}_B(b))$. The unit is $(\{\star\}, \star \mapsto \mathbf{1})$. On morphisms the monoidal product is:*

$$(f \otimes g)(a,b) := f(a) \otimes g(b)$$

  The category $(\mathbf{QRel}, \otimes, \mathbf{1})$ is an SMC.

- *The dual is* $(|A|, \mathcal{H}_A)^* := (|A|, a \mapsto \mathcal{H}_A(a)^*)$*. The unit and counit are:*

$$\eta_A^{\mathbf{QRel}} : (\star, (a,b)) \mapsto \begin{cases} \eta_A^{\mathbf{CPM}} & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \qquad \epsilon_A^{\mathbf{QRel}} : ((a,b), \star) \mapsto \begin{cases} \epsilon_A^{\mathbf{CPM}} & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

  The category $(\mathbf{QRel}, \otimes, \mathbf{1}, (-)^*)$ *is a CpCC.*

- *The coproduct* $\oplus$ *is defined on objects as*

$$(|A|, \mathcal{H}_A) \oplus (|B|, \mathcal{H}_B) := \left(|A| \uplus |B|, \begin{matrix} (0,a) \mapsto \mathcal{H}_A(a) \\ (1,b) \mapsto \mathcal{H}_B(b) \end{matrix} \right)$$

  The initial object is $(\varnothing, \varnothing)$*. On morphisms* $f : A \to C$ *and* $g : B \to C$*, the copairing* $[f; g] : A \oplus B \to C$ *is:*

$$[f;g] : \begin{array}{lcl} ((0,a),c) & \mapsto & f(a,c) \\ ((1,b),c) & \mapsto & g(b,c) \end{array}$$

  The coproduct is distributive with respect to the monoidal product.

- *The bottom morphism* $\bot$ *is the unique morphism from* $(\{\star\}, \star \mapsto \mathbf{1})$ *to* $(\varnothing, \varnothing)$*.*

- *It is non-trivial, i.e.,* $(\{\star\}, \star \mapsto \mathbf{1})$ *and* $(\varnothing, \varnothing)$ *are not isomorphic.*

It follows that **QRel** is a distributive CFC, non-trivial and with a bottom, taking the category of computations and values to be the same, the Freyd inclusion to be the identity functor, and $A \multimap B := A^* \otimes B$.

We can use the semantics described in Section 1.4.2 for most of LQΛ, completed with the following semantics for types (we recall that $\mathfrak{Q} := \mathbb{C}^2$):

$$\begin{array}{rcl} [\![\mathbf{qubit}]\!] & := & (\{\mathbf{qb}\}, \mathbf{qb} \mapsto \mathfrak{Q}) \\ \left[\!\left[A^\ell\right]\!\right] & := & [\![A]\!]^\ell \end{array}$$

Where for any object A, we define A^ℓ as $\lim_n F_A^n(\varnothing, \varnothing)$ with $F_A(X) := \mathbf{1} \oplus (A \otimes X))$ and for the following dcpo on objects:

$$(|A|, \mathcal{H}_A) \leq (|B|, \mathcal{H}_B) \iff |A| \subseteq |B| \text{ and } \forall a \in |A|, \mathcal{H}_A(a) = \mathcal{H}_B(a)$$

We note that we indeed have $F_A^n(\varnothing, \varnothing) \leq F_A^{n+1}(\varnothing, \varnothing)$ for this dcpo. By definition of A^ℓ, we have $F_A(A^\ell) = A^\ell$.

We complete the semantics for typing derivations as described in Table 3.3.

Theorem 3.2.2. *If* $\Gamma \vdash_{\mathbb{L}} t : A$ *has two typing derivations* T *and* T'*, then we have* $[\![T]\!] = [\![T']\!]$*. As it is independent from the typing derivation, we write it* $[\![t]\!]^{\Gamma \vdash_{\mathbb{L}} A}$*.*

The proof of this theorem is very similar to the proof for LΛ earlier in Theorem 1.4.6, as it extends without problems to LQΛ.

Quantum primitives:

$$[\![\mathbf{qubit}]\!] := (\{\mathbf{qb}\}, \mathbf{qb} \mapsto \mathfrak{Q})$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{bit}\end{array}}{\Gamma \vdash_{\mathbb{L}} \mathbf{new}\ t : \mathbf{qubit}} \right]\!\!\right](\gamma, \mathbf{qb}) := \mathbf{new}^{\mathbf{CPM}}_{\mathbf{tt}} \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{bit}\end{array} \right]\!\!\right](\gamma, \mathbf{tt}) + \mathbf{new}^{\mathbf{CPM}}_{\mathbf{ff}} \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{bit}\end{array} \right]\!\!\right](\gamma, \mathbf{ff})$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{qubit}\end{array}}{\Gamma \vdash_{\mathbb{L}} \mathbf{meas}\ t : \mathbf{bit}} \right]\!\!\right](\gamma, b) := \mathbf{meas}^{\mathbf{CPM}}_{b} \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{qubit}\end{array} \right]\!\!\right](\gamma, \mathbf{qb})$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{qubit}^{\otimes n}\end{array}}{\Gamma \vdash_{\mathbb{L}} \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}} \right]\!\!\right](\gamma, q) := (\!|U|\!) \circ \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{qubit}^{\otimes n}\end{array} \right]\!\!\right](\gamma, q)$$

Lists:

$$[\![A^{\ell}]\!] := [\![A]\!]^{\ell}$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : A^{\ell}\end{array}}{\Gamma \vdash_{\mathbb{L}} \mathbf{unfold}\ t : \mathbf{1} \oplus (A \otimes A^{\ell})} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : A^{\ell}\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{1} \oplus (A \otimes A^{\ell})\end{array}}{\Gamma \vdash_{\mathbb{L}} \mathbf{fold}\ t : A^{\ell}} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{L}} t : \mathbf{1} \oplus (A \otimes A^{\ell})\end{array} \right]\!\!\right]$$

Table 3.3: Denotational Semantics of LQΛ Typing Derivations

Definition 3.2.3. *If* $\vdash_{\mathbb{L}} [q, \ell, t] : A$*, we define* $[\![[q, \ell, t]]\!]^{\vdash_{\mathbb{L}} A} \in \mathbf{QRel}(\mathbf{1}, [\![A]\!])$ *as follows:*

- *we know we have* $\Delta \vdash_{\mathbb{L}} t : A$ *with* $\Delta = x_1 : \mathbf{qubit}, \ldots, x_n : \mathbf{qubit}$.
- *we recall that* $(\!|q|\!) \in \mathbf{CPM}(\mathbf{1}, \mathfrak{Q}^{\otimes n})$ *is defined in Section 2.3.3.*
- $[\![[q, \ell, t]]\!](\star, a) := [\![t]\!]((\star, \mathbf{qb}, \ldots, \mathbf{qb}), a) \circ (\!|q|\!)$

and then we define $[\![\sum_i p_i[q_i, \ell_i, t_i]]\!](\star, a)$ *as* $\sum_i p_i [\![[q_i, \ell_i, t_i]]\!](\star, a)$*, using the fact that the set* $\overline{\mathbf{CPM}}(\mathbf{1}, \mathcal{H}_{[\![A]\!]}(a))$ *is a completed positive convex cone.*

3.2.2 Examples

As an illustration, we go through the same examples as in Section 2.3.2, showing their denotational semantics in **QRel**. We sum up semantics of a term $\Gamma \vdash_{\mathbb{L}} t : A$ in tables similar to the following one. The column *Web* lists all the element (γ, a) of the web $|\Gamma| \times |A|$; the column *Operator* gives the corresponding morphism $[\![t]\!](\gamma, A)$; and the column *Space* describes the objects forming the domain and codomain of the operator, *i.e.*, $\mathcal{H}_\Gamma(\gamma) \to \mathcal{H}_A(a)$.

Web	Space	Operator
(γ, a)	$\mathcal{H}_\Gamma(\gamma) \to \mathcal{H}_A(a)$	$[\![t]\!](\gamma, a)$

The Biased Coin

This protocol simulates a probabilistic choice between the boolean true and false. We recall its QΛ term here

$$\mathrm{Coin}_p() := \mathbf{meas} \begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix}$$

We sum up its semantics in the following table, and note that as expected the probability associated to $\mathbf{tt}$ is p, and the one associated to $\mathbf{ff}$ is $(1-p)$.

Web	Space	Operator
$(\star, \mathbf{ff})$	$\mathbf{1} \to \mathbf{1}$	$(1-p) \cdot \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$
$(\star, \mathbf{tt})$	$\mathbf{1} \to \mathbf{1}$	$p \cdot \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$

The Bell States

This protocol creates pairs of maximally entangled qubits, called Bell states. We recall its QΛ term and sum up its semantics.

$$\mathrm{Bell}(b, b') := \mathbf{let}\ q_1 \otimes q_2 = (\mathbf{new}\ b) \otimes (\mathbf{new}\ b')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in\ Nc}\ (q_3 \otimes q_2)$$

Web	Space		Operator
$((\mathbf{ff},\mathbf{ff}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^{\otimes 2}$	$(\!\|\frac{1}{\sqrt{2}}\|00\rangle + \frac{1}{\sqrt{2}}\|11\rangle\|\!) :$	$z \mapsto \frac{z}{2}\cdot\begin{pmatrix}1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{pmatrix}$
$((\mathbf{ff},\mathbf{tt}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^{\otimes 2}$	$(\!\|\frac{1}{\sqrt{2}}\|01\rangle + \frac{1}{\sqrt{2}}\|10\rangle\|\!) :$	$z \mapsto \frac{z}{2}\cdot\begin{pmatrix}0&0&0&0\\0&1&1&0\\0&1&1&0\\0&0&0&0\end{pmatrix}$
$((\mathbf{tt},\mathbf{ff}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^{\otimes 2}$	$(\!\|\frac{1}{\sqrt{2}}\|00\rangle - \frac{1}{\sqrt{2}}\|11\rangle\|\!) :$	$z \mapsto \frac{z}{2}\cdot\begin{pmatrix}1&0&0&-1\\0&0&0&0\\0&0&0&0\\-1&0&0&1\end{pmatrix}$
$((\mathbf{tt},\mathbf{tt}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^{\otimes 2}$	$(\!\|\frac{1}{\sqrt{2}}\|01\rangle - \frac{1}{\sqrt{2}}\|10\rangle\|\!):$	$z \mapsto \frac{z}{2}\cdot\begin{pmatrix}0&0&0&0\\0&1&-1&0\\0&-1&1&0\\0&0&0&0\end{pmatrix}$

For each point of the web, its operator is the image by $(\!|-|\!)$ of one of the four Bell states.

The Bell Measure

This protocol, when applied to a Bell state, recovers the pair of booleans that created it. We recall its QΛ term and sum up its semantics.

$$\mathrm{BellM}(q,q') := \mathbf{let}\ q_1 \otimes q_2 = \mathbf{Nc}\ (q \otimes q')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in}\ (\mathbf{meas}\ q_3) \otimes (\mathbf{meas}\ q_2)$$

Web	Space	Operator
$((\mathbf{qb},\mathbf{qb}),(\mathbf{ff},\mathbf{ff}))$	$\mathfrak{Q}^{\otimes 2} \to \mathbf{1}$	$M \mapsto \frac{m_{11}+m_{14}+m_{41}+m_{44}}{2}$
$((\mathbf{qb},\mathbf{qb}),(\mathbf{ff},\mathbf{tt}))$	$\mathfrak{Q}^{\otimes 2} \to \mathbf{1}$	$M \mapsto \frac{m_{22}+m_{23}+m_{32}+m_{33}}{2}$
$((\mathbf{qb},\mathbf{qb}),(\mathbf{tt},\mathbf{ff}))$	$\mathfrak{Q}^{\otimes 2} \to \mathbf{1}$	$M \mapsto \frac{m_{11}-m_{14}-m_{41}+m_{44}}{2}$
$((\mathbf{qb},\mathbf{qb}),(\mathbf{tt},\mathbf{tt}))$	$\mathfrak{Q}^{\otimes 2} \to \mathbf{1}$	$M \mapsto \frac{m_{22}-m_{23}-m_{32}+m_{33}}{2}$

One can check that $[\![\mathrm{BellM}(q,q')]\!] \circ [\![\mathrm{Bell}(b,b')]\!] = \mathbf{id}^{\mathbf{QRel}}_{[\![\mathbf{bit}^{\otimes 2}]\!]}$.

The Quantum Teleportation

In Section 2.3.2, we did not give a term for the quantum teleportation protocol, as multi-agent protocols are not easily described in our MiniQ language. We will instead use the quantum teleportation as a way to illustrate higher order terms by reformulating it as in [PSV14]:

- Alice and Bob create a pair of entangled qubit, then create the functions $f : \mathbf{qubit} \multimap \mathbf{bit}^{\otimes 2}$ which makes a Bell measurement on its input and one of the two qubit of the pair, and $g : \mathbf{bit}^{\otimes 2} \multimap \mathbf{qubit}$ which applies some well-chosen unitary function to the second qubit of the pair. We describe them in the term QTelep() below.

- Alice takes the function f and Bob takes g.
- Alice has a qubit q she wishes to communicate. For that, she computes $f(q)$ and sends the two resulting booleans to Bob.
- Bob receives two booleans b, b' and wishes to recover the qubit q. For that, he computes $g(b \otimes b')$.

Before defining the term QTelep(), we first define the Bell unitary and give its semantics in **QRel**. We note that the semantics of the Bell unitary and the semantics of the Bell States are identical up to a factor half. This justifies the choice of those specific unitary matrices.

$$\begin{aligned}\mathrm{BellU}(b,b') := \lambda q.\mathbf{if}\ b \quad \mathbf{then} \quad & \left(\mathbf{if}\ b'\ \mathbf{then}\ \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}\ q\ \mathbf{else}\ \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\ q\right)\\ \mathbf{else} \quad & \left(\mathbf{if}\ b'\ \mathbf{then}\ \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\ q\ \mathbf{else}\ \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\ q\right)\end{aligned}$$

We sum up the semantics of the Bell unitary protocol:

Web	Space	Operator
$((\mathbf{ff},\mathbf{ff}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix}1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{pmatrix}$
$((\mathbf{ff},\mathbf{tt}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix}0&0&0&0\\0&1&1&0\\0&1&1&0\\0&0&0&0\end{pmatrix}$
$((\mathbf{tt},\mathbf{ff}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix}1&0&0&-1\\0&0&0&0\\0&0&0&0\\-1&0&0&1\end{pmatrix}$
$((\mathbf{tt},\mathbf{tt}),(\mathbf{qb},\mathbf{qb}))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix}0&0&0&0\\0&1&-1&0\\0&-1&1&0\\0&0&0&0\end{pmatrix}$

We now define the term QTelep(). This term is typed by $\vdash_{\mathbb{L}} \mathrm{QTelep}() : (\mathbf{qubit} \multimap \mathbf{bit}^{\otimes 2}) \otimes (\mathbf{bit}^{\otimes 2} \multimap \mathbf{qubit})$, and its semantics cannot be described just by describing separately the two functions it creates: the two functions are linked through quantum entanglement.

$$\mathrm{QTelep}() := \mathbf{let}\ q_1 \otimes q_2 = \mathrm{Bell}(\mathbf{ff},\mathbf{ff})\ \mathbf{in}\ (\lambda q.\mathrm{BellM}(q_1,q)) \otimes (\lambda(b \otimes b').\mathrm{BellU}(b,b'))$$

We sum up the semantics of this term in the following table. In this table, we assume b and b' to be any booleans, and $\neg b$ and $\neg b'$ to be their negations.

Web	Space	Operator
$(\star, ((\mathbf{qb}, (b, b)), ((b, b), \mathbf{qb})))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$
$(\star, ((\mathbf{qb}, (b, b')), ((b, \neg b'), \mathbf{qb})))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
$(\star, ((\mathbf{qb}, (b, b')), ((\neg b, b'), \mathbf{qb})))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$
$(\star, ((\mathbf{qb}, (b, b')), ((\neg b, \neg b'), \mathbf{qb})))$	$\mathbf{1} \to \mathfrak{Q}^* \otimes \mathfrak{Q}$	$z \mapsto z \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

The first line here is the most important, as it can be rewritten as

$$[\![\mathrm{QTelep}()]\!]\,(\star, ((\mathbf{qb}, (b, b)), ((b, b), \mathbf{qb}))) = [\![\lambda x^{\mathbf{qubit}}.x]\!]\,(\star, (\mathbf{qb}, \mathbf{qb}))$$

which shows that if Bob follows the protocol and applies his functions to the bits Alice sent to him, the whole protocol has the same semantics as $[\![\lambda x^{\mathbf{qubit}}.x]\!]$, in other words the whole protocol simplifies to an identity function and Bob obtains the qubit Alice wanted to send.

3.2.3 Soundness and Adequacy for LQΛ

In this section, we state the different properties of **QRel**: value-substituting, invariant, sound, adequate and the direct implication of fully abstract. The proofs do not significantly differ from the proofs in the case of Λ in Section 1.4.3. As such, we refer to them for more details.

Lemma 3.2.4 (Value Substitution)**.** *For every term $\Gamma, x : A \vdash_{\mathbb{L}} t : B$ and every value $\Delta \vdash_{\mathbb{L}} v : A$:*

$$[\![t]\!] \circ ([\![\Gamma]\!] \otimes [\![v]\!]) = [\![t\{x \leftarrow v\}]\!]$$

The proof is then the same as for Λ in Section 1.4.3.

Lemma 3.2.5 (Context Factorisation)**.** *For every term $\Gamma \vdash_{\mathbb{L}} s : A$ and $\Gamma, \Delta \vdash_{\mathbb{L}} E[s] : B$, with $E[-]$ an evaluation context, we have a morphism $[\![E]\!] \in \mathbf{QRel}([\![A]\!] \otimes [\![\Delta]\!], [\![B]\!])$ such that*

$$[\![E[s]]\!] = [\![E]\!] \circ ([\![s]\!] \otimes [\![\Delta]\!])$$

The proof is then the same as for Λ in Section 1.4.3.

Lemma 3.2.6 (Invariance)**.** *For every closures $\Gamma \vdash_{\mathbb{L}} c : A$ and $\Gamma \vdash_{\mathbb{L}} d : A$*

$$c \to d \implies [\![c]\!] = [\![d]\!]$$

Proof. Using the same proof as for Λ, we obtain that for every pair of terms $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma \vdash_{\mathbb{L}} s : A$ that do not use **fold** or **unfold**, we have $t \to s \implies [\![t]\!] = [\![s]\!]$. Since the rules for **fold** and **unfold** are trivial, we extend without difficulty to terms containing them.

Now, we consider reduction rules over closures. Using the context factorisation Lemma 3.2.5, we obtain that for $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma, \Delta \vdash_{\mathbb{L}} E[t] : B$ we have

$$[\![[q, \ell, E[t]]]\!] = [\![E]\!] \circ ([\![t]\!] \otimes [\![\Delta]\!]) \circ (\!|q|\!)$$

It follows that if we have $t \to s$, then we have:

$$[q, \ell, E[t]] \to [q, \ell, E[s]] \implies [\![[q, \ell, E[t]]]\!] = [\![[q, \ell, E[s]]]\!]$$

For the reduction of **new**, **meas** and **U**, we simply use the fact that the operational semantics uses morphisms of **Hilb** while the denotational semantics uses the corresponding morphisms of $\overline{\mathbf{CPM}}$.

The last two reductions relies on $\overline{\mathbf{CPM}}$ being a completed positive convex cone, and composition being linear for this cone. □

Theorem 3.2.7 (Soundness and Adequacy)**.** *For every term $\vdash_{\mathbb{L}} t : \mathbf{1}$, we have*

$$\forall p \in [0,1], \mathbb{P}(t \Downarrow) = p \iff [\![t]\!] = [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$$

In particular, we have that $[\![t]\!](\star, \star) \in \mathbf{CPM}(\mathbf{1}, \mathbf{1})$, so it is finitary in $\overline{\mathbf{CPM}}$.

Proof. We use strong normalisation: $[\varnothing, \varnothing, t] \to^* \sum_i p_i[q_i, \ell_i, ()] + \sum_j p'_j[q'_j, \ell'_j, \bot]$. For $p = \sum_i p_i$, we have $\left[\!\!\left[\sum_i p_i[q_i, \ell_i, ()] + \sum_j p'_j[q'_j, \ell'_j, \bot]\right]\!\!\right] = [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$. Using invariance, we obtain the expected equivalence. □

Corollary 3.2.8. *For every pair of terms $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma \vdash_{\mathbb{L}} s : A$, we have*

$$[\![t]\!] = [\![s]\!] \implies t =_{\text{obs}} s$$

Proof. We take an observation context $\mathcal{O}[_]$ for $\Gamma \vdash_{\mathbb{L}} A$. A simple proof by induction shows that there exists a function F such that for all term $\Gamma \vdash_{\mathbb{L}} t : A$ we have $[\![\mathcal{O}[t]]\!] = F([\![t]\!])$. When $\mathcal{O}[_]$ is an evaluation context, the context factorisation Lemma 3.2.5 shows that F is simply a post-composition, but this is not true in general[a]. We assume that $[\![t]\!] = [\![s]\!]$. We have $[\![\mathcal{O}[t]]\!] = F([\![t]\!]) = F([\![s]\!]) = [\![\mathcal{O}[s]]\!]$. Using adequacy, it follows that $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow)$, hence the result. □

[a] For example, if one consider $\mathcal{O}[t] = s; t$, F will be a pre-composition instead.

3.2.4 Full Abstraction for LQΛ

We now prove the reverse implication of Corollary 3.2.8, *i.e.*, full abstraction. A more direct proof can be found in [SV08], but here we set up some tools we will use for the full abstraction proof in the non-linear case in Part III. We utilise a method similar to [ETP14], *i.e.*, relying on test terms. While we do not yet have any formal parameters in our terms, this is because we are still in the linear fragment of the language. We will eventually in Section 10.3.2 define test terms very similar to those in [ETP14].

Consider the terms $t := \lambda x^{\mathbf{bit}}.x$ and $s := \lambda x^{\mathbf{bit}}\mathbf{if}\ x\ \mathbf{then}\ \mathrm{Coin}_{1/2}()\ \mathbf{else}\ \mathbf{ff}$. We have $[\![t]\!] \neq [\![s]\!]$, so we want to find an observation context $\mathcal{O}[_]$ such that $\mathbb{P}(\mathcal{O}[t] \Downarrow) \neq \mathbb{P}(\mathcal{O}[s] \Downarrow)$. Since $[\![t]\!] \neq [\![s]\!]$, this means that there exists a point of the web $a \in |\mathbf{bit} \multimap \mathbf{bit}|$ such that $[\![t]\!](\star, a) \neq [\![s]\!](\star, a)$. For example, we have here $a = (\mathbf{tt}, \mathbf{tt})$:

$$[\![t]\!](\star, (\mathbf{tt}, \mathbf{tt})) = \mathbf{id}_1^{\mathbf{CPM}} \qquad [\![s]\!](\star, (\mathbf{tt}, \mathbf{tt})) = \frac{1}{2}\mathbf{id}_1^{\mathbf{CPM}}$$

From this point of the web where they differ, we build the observation context:

$$\mathcal{O}[_] := \mathbf{if}\ _\ \mathbf{tt}\ \mathbf{then}\ ()\ \mathbf{else}\ \bot$$

And we obtain $\mathbb{P}(\mathcal{O}[t] \Downarrow) = 1$ and $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \frac{1}{2}$.

We detail the construction of $\mathcal{O}[_]$, and note that it has two main components: a generator part $\mathbf{tt}$ that feeds the adequate value into the term we want to observe, and a test part $\mathbf{if}\ -\ \mathbf{then}\ ()\ \mathbf{else}\ \bot$ that converge if and only if the output is the expected output. In this subsection, we generalise this approach and define for every type A and point of the web $a \in |A|$ a generator term $\vdash_{\mathbb{L}} \Uparrow_a^A : A$ and a test term $\vdash_{\mathbb{L}} \Downarrow_a^A : A \multimap \mathbf{1}$. Those terms are heavily inspired from the terms $\mathcal{P}$ and $\mathcal{N}$ of [ETP14].

There are some additional subtleties whenever A contains $\mathbf{qubit}$. Indeed, no single test term $\vdash_{\mathbb{L}} \Downarrow_{\mathbf{qb}}^{\mathbf{qubit}}$ can distinguish all the terms $\vdash_{\mathbb{L}} t : \mathbf{qubit}$ that have different semantics. But Proposition 2.1.18 ensures that four test terms can distinguish them. So we define in Definition 3.2.10 the test terms $\Downarrow_{\mathbf{qb},0}^{\mathbf{qubit}}$, $\Downarrow_{\mathbf{qb},1}^{\mathbf{qubit}}$, $\Downarrow_{\mathbf{qb},2}^{\mathbf{qubit}}$ and $\Downarrow_{\mathbf{qb},3}^{\mathbf{qubit}}$ according to the four morphisms of Proposition 2.1.18, and proceed similarly for generator terms.

We now formalise the extended points of the web, which for qubits are the pairs $\mathbf{qb}, i$ with $i \in \{0, 1, 2, 3\}$.

Definition 3.2.9. *For A a type, we define its extended web $|A|_e$ inductively is a similar way to $|A|$.*

$$\begin{array}{rcl} |\mathbf{1}|_e & := & |\mathbf{1}| \\ |\mathbf{qubit}|_e & := & |\mathbf{qubit}| \times \{0,1,2,3\} \\ |A \otimes B|_e & := & |A|_e \times |B|_e \\ |A \multimap B|_e & := & |A|_e \times |B|_e \\ |A \oplus B|_e & := & |A|_e \uplus |B|_e \\ |A^\ell|_e & := & |\mathbf{1}|_e \oplus (|A|_e \otimes |A^\ell|_e) \end{array}$$

$$
\begin{array}{lcllcl}
\Uparrow^{\mathbf{1}}_{\star} & := & () & & & \\
\Uparrow^{\mathbf{qubit}}_{\mathbf{qb},0} & := & \mathbf{new}\ \mathbf{ff} & \Uparrow^{\mathbf{qubit}}_{\mathbf{qb},1} & := & \mathbf{new}\ \mathbf{tt} \\
\Uparrow^{\mathbf{qubit}}_{\mathbf{qb},2} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\mathbf{i}}{\sqrt{2}} & -\frac{\mathbf{i}}{\sqrt{2}} \end{pmatrix} (\mathbf{new}\ \mathbf{ff}) & \Uparrow^{\mathbf{qubit}}_{\mathbf{qb},3} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{\mathbf{i}}{\sqrt{2}} \\ \frac{\mathbf{i}}{\sqrt{2}} & -\frac{\mathbf{i}}{\sqrt{2}} \end{pmatrix} (\mathbf{new}\ \mathbf{ff}) \\
\Uparrow^{A\otimes B}_{(a,b)} & := & \Uparrow^{A}_{a} \otimes \Uparrow^{B}_{b} & \Uparrow^{A\multimap B}_{(a,b)} & := & \lambda x.\ \Downarrow^{A}_{a}\ x; \Uparrow^{B}_{b} \\
\Uparrow^{A\oplus B}_{(0,a)} & := & \mathbf{inj}_{\ell}\ \Uparrow^{A}_{a} & \Uparrow^{A\oplus B}_{(1,b)} & := & \mathbf{inj}_{r}\ \Uparrow^{B}_{b} \\
\Uparrow^{A^{\ell}}_{(0,\star)} & := & [\,] & \Uparrow^{A^{\ell}}_{(1,(a,b))} & := & \Uparrow^{A}_{a} :: \Uparrow^{A^{\ell}}_{b}
\end{array}
$$

Table 3.4: Generator terms $\vdash_{\mathbb{L}} \Uparrow^{A}_{a} : A$

For every object $(A, \mathcal{H}_A)$, and point of its web $a \in |A|$, we define its number of **qubit**s $\#_{\mathbf{qubit}}(a)$ as $\log(\dim(\mathcal{H}_A(a)))$ (we consider the log in base 2). When this object comes from a type, this value will always be an integer. For a type A, from $e \in |A|_e$, we can canonically recover $a \in |A|$ by removing all the indices, and $i \in \{0,1,2,3\}^{\#_{\mathbf{qubit}}(a)}$ by collecting all of them. This is in fact a bijective operation, and we write $e = a|i$.

Definition 3.2.10. *For every element of the extended web of a type A, we define a generator term and a test term, written $\Uparrow^{A}_{a}$ and $\Downarrow^{A}_{a}$ as in Tables 3.4 and 3.5. They are typed by:*

$$\vdash_{\mathbb{L}} \Uparrow^{A}_{a} : A \qquad\qquad \vdash_{\mathbb{L}} \Downarrow^{A}_{a} : A \multimap \mathbf{1}$$

We note that the terms for **qubit** *are built from the morphisms of Proposition 2.1.18.*

The goal of test terms is to "extract" the coefficient corresponding to a point of the web. Formally, we expect that for any term $x : A \vdash_{\mathbb{L}} t : B$, we have:

$$\left[\!\left[\mathbf{let}\ x^{A} = \Uparrow^{A}_{a|i}\ \mathbf{in}\ \Downarrow^{B}_{b|j}\ t\right]\!\right](\star,\star) = T_j^{\mathcal{H}_{[\![B]\!]}(b)} \circ [\![t]\!]\,(a,b) \circ G_i^{\mathcal{H}_{[\![A]\!]}(a)}$$

where G and T are the morphisms of Proposition 2.1.18. We can deduce this property from the following lemma.

Lemma 3.2.11 (Semantics of Tests and Generators)**.** *For A a type and $(a|i) \in |A|_e$:*

$$a \neq b \implies \left[\!\left[\Uparrow^{A}_{a|i}\right]\!\right](\star,b) = 0_{\mathbf{1},[\![A]\!]} \text{ and } \left[\!\left[\Downarrow^{A}_{a|i}\ x\right]\!\right](b,\star) = 0_{[\![A]\!],\mathbf{1}}$$

And moreover:

$$
\begin{array}{rcll}
\left[\!\left[\Uparrow^{A}_{a|i}\right]\!\right](\star,a) & = & G_i^{\mathcal{H}_{[\![A]\!]}(a)} & \in\ \mathbf{CPM}(\mathbf{1}, \mathcal{H}_{[\![A]\!]}(a)) \\
\left[\!\left[\Downarrow^{A}_{a|i}\ x\right]\!\right](a,\star) & = & T_i^{\mathcal{H}_{[\![A]\!]}(a)} & \in\ \mathbf{CPM}(\mathcal{H}_{[\![A]\!]}(a), \mathbf{1})
\end{array}
$$

$$
\begin{array}{lcl}
\Downarrow^{\mathbf{1}}_{\star} & := & \lambda().() \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},0} & := & \lambda q.\mathbf{if\ meas}\ q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},1} & := & \lambda q.\mathbf{if\ meas}\ q\ \mathbf{then}\ ()\ \mathbf{else}\ \bot \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},2} & := & \lambda q.\mathbf{if\ meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},3} & := & \lambda q.\mathbf{if\ meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{\mathbf{i}}{\sqrt{2}} \\ -\frac{\mathbf{i}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
 & & \\
\Downarrow^{A\otimes B}_{(a,b)} & := & \lambda(x\otimes y).\ \Downarrow^{A}_{a}\ x\ ;\Downarrow^{B}_{b}\ y \\
\Downarrow^{A\multimap B}_{(a,b)} & := & \lambda f.\mathbf{let}\ x = f\ \Uparrow^{A}_{a}\ \mathbf{in}\ \Downarrow^{B}_{b}\ x \\
\Downarrow^{A\oplus B}_{(0,a)} & := & \lambda x.\delta\ (x,\ y.\ \Downarrow^{A}_{a}\ y,\ z.\bot) \\
\Downarrow^{A\oplus B}_{(1,b)} & := & \lambda x.\delta\ (x,\ y.\bot,\ z.\ \Downarrow^{B}_{b}\ z) \\
\Downarrow^{A^{\ell}}_{(0,\star)} & := & \lambda \ell.\mathbf{match}\ \ell\ \mathbf{with}\ ([\,]\mapsto () \mid x :: y \mapsto \bot) \\
\Downarrow^{A^{\ell}}_{(1,(a,b))} & := & \lambda \ell.\mathbf{match}\ \ell\ \mathbf{with}\ \left([\,]\mapsto \bot \;\middle|\; x :: y \mapsto \Downarrow^{A}_{a}\ x\ ;\Downarrow^{A^{\ell}}_{b}\ y\right)
\end{array}
$$

Table 3.5: Test terms $\vdash_{\mathbb{L}} \Downarrow^{A}_{a} : A \multimap \mathbf{1}$

Proof. The terms for $A = \mathbf{qubit}$ have been created such that it holds for them. We then simply proceed by induction on the type, using the compact closure of **CPM** for the function case. □

However, to be able to use the properties of G and T described in Proposition 2.1.18, we first need to ensure that $[\![c]\!]$ is in **CPM**, *i.e.*, is a finitary element of $\overline{\mathbf{CPM}}$.

Proposition 3.2.12 (Finitary Semantics)**.** *For $\Gamma \vdash_{\mathbb{L}} t : A$ a term, for every $(\gamma, a) \in |\Gamma| \times |A|$, $[\![t]\!](\gamma, a) \in \mathbf{CPM}([\![\Gamma]\!], [\![A]\!])$. In other words $[\![-]\!]$ never uses infinitary annotations.*

Proof. We first note that for every Hilbert space H of dimension n, there exists some $\alpha, \beta \in \mathbb{R}_{>0}$ such that

$$\sum_{i\in\{0,1,2,3\}^n} G_i^H \sqsupseteq \alpha \mathbf{Tr}_H^{\dagger} \qquad\qquad \sum_{i\in\{0,1,2,3\}^n} T_i^H \sqsupseteq \beta \mathbf{Tr}_H$$

We assume $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, and decompose $\gamma = (\gamma_1, \ldots, \gamma_n)$. We take $i_1, \ldots, i_n, j$ such that $\gamma_k|i_k \in |A_k|_e$ and $a|j \in |A|_e$. We define the term $s_{i_1,\ldots,i_n,j}$ defined as $s_{i_1,\ldots,i_n,j} := \mathbf{let}\ x_1 \otimes \ldots \otimes x_n = \Uparrow^{X_1}_{\gamma_1|i_1} \otimes \ldots \otimes \Uparrow^{X_n}_{\gamma_n|i_n}\ \mathbf{in}\ \Downarrow^{A}_{a|j}\ t$. Using Lemma 3.2.11, we have

$$\left[\!\left[s_{i_1,\ldots,i_n,j}\right]\!\right](\star,\star) = T_j^{\mathcal{H}_{[\![A]\!]}(a)} \circ [\![t]\!](\gamma, a) \circ \bigotimes_{k=1}^{n} G_{i_k}^{\mathcal{H}_{[\![A_k]\!]}(\gamma_k)}$$

Summing over all the possible $i_1, \dots, i_n, j$ (there are finitely many of them), it follows that there exist some $\alpha, \beta \in \mathbb{R}_{>0}$ such that

$$\sum_{i_1,\dots,i_n,j} \left[\!\left[s_{i_1,\dots,i_n,j} \right]\!\right](\star,\star) \sqsupseteq \beta \mathbf{Tr}_{\mathcal{H}_{[\![A]\!]}(a)} \circ [\![t]\!](\gamma, a) \circ \alpha \mathbf{Tr}^{\dagger}_{\mathcal{H}_{[\![\Gamma]\!]}(\gamma)}$$

We note that **CPM** already has all trace-bounded suprema (the trace is a norm), as such a morphism of $\overline{\mathbf{CPM}}(H, K)$ is in $\mathbf{CPM}(H, K)$ if and only its composition by the trace $\mathbf{Tr}_K$ and dagger-trace $\mathbf{Tr}^{\dagger}_H$ is in $\mathbf{CPM}(\mathbf{1}, \mathbf{1})$. The soundness and adequacy ensure that $[\![s]\!](\star,\star) \in \mathbf{CPM}(\mathbf{1}, \mathbf{1})$, so it follows that $[\![t]\!](\gamma, a) \in \mathbf{CPM}([\![\Gamma]\!], [\![A]\!])$. □

We then have all the tools to prove the reverse implication of the full abstraction, and conclude with the full abstraction theorem.

Lemma 3.2.13 (Characterisation by Tests and Generators)**.** *We define the set of observers* $O_{x:A \vdash_{\mathbb{L}} B}$ *as*

$$O_{x:A \vdash_{\mathbb{L}} B} = \left\{ \mathbf{let}\ x^A = \Uparrow^A_{a|i}\ \mathbf{in}\ \Downarrow^B_{b|j}\ _ \;\middle|\; (a|i) \in |A|_e, (b|j) \in |B|_e \right\}$$

For every pair of terms $x : A \vdash_{\mathbb{L}} t : B$ *and* $x : A \vdash_{\mathbb{L}} s : B$ *we have*

$$\forall \mathcal{O}[_] \in O_{x:A \vdash_{\mathbb{L}} B},\ \mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow) \implies [\![t]\!] = [\![s]\!]$$

Proof. Using Lemma 3.2.11, for $\mathcal{O}[_] = \mathbf{let}\ x^A = \Uparrow^A_{a|i}\ \mathbf{in}\ \Downarrow^B_{b|j}\ _$, we immediately have $[\![\mathcal{O}[t]]\!](\star,\star) = T_j^{\mathcal{H}_{[\![B]\!]}(b)} \circ [\![t]\!](a, b) \circ G_i^{\mathcal{H}_{[\![A]\!]}(a)}$. So if for all observers we have $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow)$, using soundness and adequacy (Theorem 3.2.7) we have for all $(a|i) \in |A|_e$ and all $(b|j) \in |B|_e$:

$$T_j^{\mathcal{H}_{[\![B]\!]}(b)} \circ [\![t]\!](a, b) \circ G_i^{\mathcal{H}_{[\![A]\!]}(a)} = T_j^{\mathcal{H}_{[\![B]\!]}(b)} \circ [\![s]\!](a, b) \circ G_i^{\mathcal{H}_{[\![A]\!]}(a)}$$

Using Proposition 3.2.12, we know that $[\![t]\!]$ and $[\![s]\!]$ are in **CPM**, so using Proposition 2.1.18 we deduce that for all $a \in |A|$ and $b \in |B|$ we have:

$$[\![t]\!](a, b) = [\![s]\!](a, b)$$ □

Theorem 3.2.14 (Full Abstraction)**.** *For every pair of terms* $\Gamma \vdash_{\mathbb{L}} t : A$ *and* $\Gamma \vdash_{\mathbb{L}} s : A$*, we have*

$$[\![t]\!] = [\![s]\!] \iff t =_{\text{obs}} s$$

Proof. The direct implication is exactly Corollary 3.2.8. We now assume $t =^{\Gamma \vdash_{\mathbb{L}} A}_{\text{obs}} s$. We write $P = \bigotimes_{(x:X)\in\Gamma} X$.

We consider $t' = \mathbf{let}\ y = \bigotimes_{(x:X)\in\Gamma} x\ \mathbf{in}\ t$ and $s' = \mathbf{let}\ y = \bigotimes_{(x:X)\in\Gamma} x\ \mathbf{in}\ s$. It follows that $c' =_{\text{obs}} d'$. In particular, for every $\mathcal{O}[_] \in O_{y:P \vdash_{\mathbb{L}} A}$, we have $\mathbb{P}(\mathcal{O}[t'] \Downarrow) = \mathbb{P}(\mathcal{O}[s'] \Downarrow)$. It follows from the previous lemma that $[\![t']\!] = [\![s']\!]$. From the definition of the semantics, it follows immediately that $[\![t]\!] = [\![s]\!]$. □

We recall that this theorem is essentially the same as the linear full abstraction theorem of [SV08]. However, the proof is new, and is designed so that it brings useful ingredients to generalise to the full language later on.

Part II

Quantum Game Semantics

Overview of Part II

We now build a fully abstract game semantics model for QΛ.

In the fourth chapter, we go through preliminaries about concurrent games and strategies. We define event structures, a partial-order structure introduced by Nielsen, Plotkin and Winskel in [NPW79], which we will use later to represent the classical control flow of programs. We then build on top of event structures the category of concurrent games and strategies. This chapter does not contain any original contribution, and a more detailed presentation of those notions can be found in [CCRW17].

In the fifth chapter, we quickly present the already existing notion of probabilistic strategies from [Win14][2], and then develop the category of quantum games and strategies. This is the first central key contribution of this thesis.

In the sixth chapter, we add the final ingredients needed to make a model for QΛ, including a notion of payoff to characterise which configurations are acceptable "stopping points" of the execution, a visibility condition ensuring the absence of deadlocks, and framing quantum games and strategies as a distributive closed Freyd category. We then prove a result of full abstraction for both LQΛ and AQΛ, and show links with the relational model for LQΛ. This chapter is almost entirely original work, though the notion of payoff can already be found in [Mel05], the notion of visibility is the same as in [CCW15b], and the link between game semantics and the relational model was already explored in the probabilistic case in [CP18].

[2] While this paper also presents a notion of quantum strategies, no simple link exists between this notion and ours.

Chapter 4

Preliminaries on Concurrent Game Semantics

4.1 Introduction to Game Semantics

In this section, we give a quick informal overview of game semantics. One might remark some differences with the most common definitions in the literature. This is due to two choices: firstly, the language we are considering is call-by-value, so is to be compared to pre-existing literature on call-by-value game semantics like [AM97, HY97], rather than the more traditional call-by-name game semantics; secondly, we are using the formalism of concurrent game semantics, which relies on partial orders rather than plays and pointers.

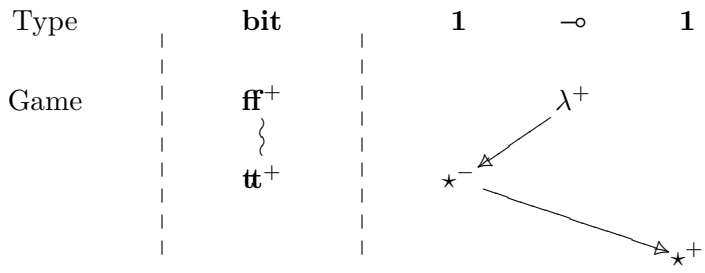
4.1.1 Player and Opponent Moves

The core concepts in game semantics are the concepts of moves, players, games and strategies. A *move* represents a computational event. As a rough approximation, moves are the atoms that compose a point of a web, for example in $((\star, \mathbf{ff}), (\mathbf{ff}, \star)) \in |(\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{bit} \multimap \mathbf{1})|$ each of the four $\star$ and $\mathbf{ff}$ will be a different move. Moves are split into two kinds, *Player* moves and *Opponent* moves. In game semantics, we represent the execution of a program by an interactive system between Player and Opponent. Player represents the program itself, is usually denoted by the positive polarity, and its moves represent outputs of the program. Opponent represents the environment, possibly the user of the program, is usually denoted by the negative polarity, and its moves represent the inputs of the program.

4.1.2 Games

The set of all available moves, together with their polarity and some additional structure discussed later, is called a game. In game semantics, we will associate such a game to every type of a language. For example, the game for the type $\mathbf{bit}$ will have two moves $\mathbb{tt}^+$ and

$\mathbf{ff}^+$, which are both Player moves as they correspond to the two different values that a program of type **bit** can produce; and those two moves will be said "in conflict", meaning that if one is played, the other can no longer be played. As another example, the game for the type $\mathbf{1} \multimap \mathbf{1}$ will have three moves λ^+, $\star^-$ and $\star^+$, the first and the third for Player and the second for Opponent, and a partial order $\lambda^+ \leq \star^- \leq \star^+$ called causality. The move λ^+ is an artefact of call-by-value, and stands for "the function is ready to be called", it is a signal sent from the program to a potential user. The move $\star^-$ stands for "the user called the function on input ()". Lastly, the move $\star^+$ stands for "the function returns with result ()". Those three events are ordered by causality, meaning that the first one is a prerequisite for the second, and the second a prerequisite for the third. Represented with figures, with the wiggly line standing for "conflict" and the arrows for "causality", we have



As we will see in Section 6.1, computing the game of a type is compositional, hence every move of the game corresponds to a syntactic component of the type. A convention we will follow in figures representing the game of a type is that we write the type at the top of the figure, and will usually put under every component of the type the moves associated to this component.

As a more advanced example, we consider the left hand side of Fig. 4.1, which is the game for $\mathbf{bit} \multimap \mathbf{bit}$. As for $\mathbf{1} \multimap \mathbf{1}$, if we read it from top to bottom it starts with a move λ^+ standing for "a function is ready to be called". It continues with two moves $\mathbf{ff}^-$ and $\mathbf{tt}^-$ standing for "the user called the function on input false" and "the user called the function on input true". Those two negative events are in conflict, as the user can only input one value, and causally depend on λ^+ as the user can only call a function which is ready to be called. Then, as the output of the function is also a boolean, we should have two positive moves $\mathbf{ff}^+$ and $\mathbf{tt}^+$ standing for "the function returns with result false", and "the function returns with result true" respectively, in conflict with each other, and causally depending on the input of the function. However, there are two possible inputs for the function, so we have a copy of those events for every possible input of the function.

To sum up, a game is a set of moves, each of them being either a Player move or an Opponent move, together with a causality relation and a conflict relation. It represents all the observable events of a given type. In Section 4.2, we define event structures which is the mathematical framework that we will use to formalise games.

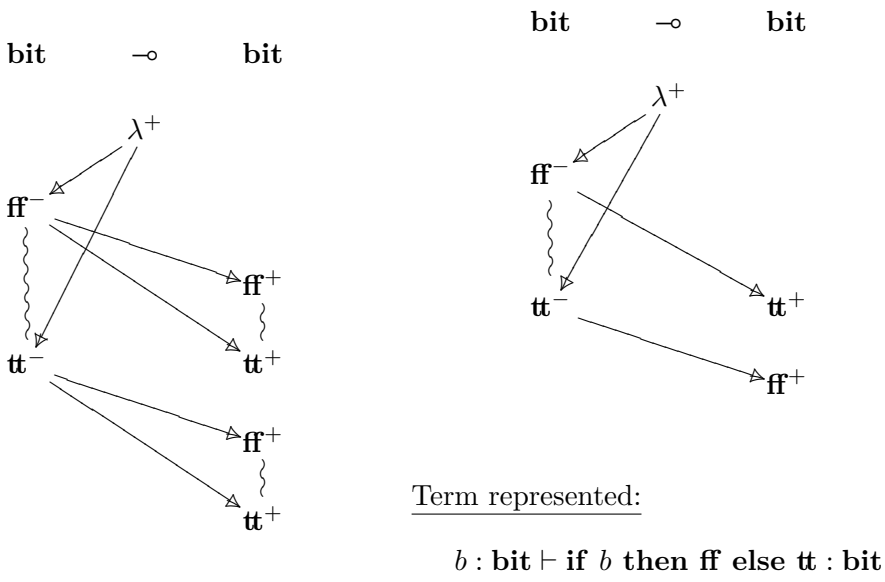


Figure 4.1: A game (left) and a strategy (right)

4.1.3 Strategies

The last core component of game semantics is the notion of strategy. If games are the representation of types, then strategies are the representation of programs. A strategy describes a possible behaviour of Player on a game. For example, on the game at the left hand side of Fig. 4.1, one can craft the strategy for the program that takes a boolean and returns its negation, and sum it up in this table:

Trigger:	Start	Opponent plays $\mathbf{ff}^-$	Opponent plays $\mathbf{tt}^-$
Reaction:	Play λ^+	Play the corresponding $\mathbf{tt}^+$	Play the corresponding $\mathbf{ff}^+$

Graphically, we represent this strategy with the figure at the right hand side of Fig. 4.1. Every move of the strategy corresponds to a move from the game, and the strategy just selects some moves of the game for Player to play in reaction to Opponent moves.

We will usually consider strategies from one game to another. To explain this concept we will take the example of a well-known strategy to not lose a game of chess against a grandmaster:

- Play two games of chess at once. In the first, you play Black against grandmaster G_1. In the second, you play White against grandmaster G_2.
- Whatever G_1 plays, immediately make the same move against G_2. Whatever G_2 plays, immediately make the same move against G_1. The execution of this strategy

should look alike "G_1 moves a pawn; You move the same pawn against G_2; G_2 moves a knight; You move the same knight against G_1; G_1 moves a knight too; You move the same knight against G_2; *etc.*".

- Eventually, you will either lose one of the two games, then you will immediately win the other one, or you will tie a game and immediately tie the other one.

This strategy is called the copy-cat strategy from Chess to Chess. It is the prime example of a strategy from a game to another: you play the two games at once, adapting your moves on one game depending on what happens on the other. Note that in the example, the strategy assumes you are Black in the first game and White in the second. More generally, a strategy from the game A to the game B plays on both $A^\perp$ and B at once, where $A^\perp$ is A with Player and Opponent exchanged.

In game semantics, a term typed $\Gamma \vdash t : A$ will be represented by a strategy from the game of Γ to the game of A. Indeed, the strategy should play on A to produce the expected values (including functions), while being able to use variables (including functions) from the game of Γ as if it was a user, *i.e.*, Opponent, with respect to this game.

4.2 The Category of Event Structures

To represent those games and strategies formally, we introduce the category of event structures. Event structures were first introduced in [NPW79] to represent concurrent systems and the unfolding of Petri Nets. Multiple kinds of event structures have been defined since then. In this thesis, we use prime event structures, as defined in [NPW79]. While the language QΛ does not have any concurrent behaviour a priori, we still have a limited use of concurrency: when representing terms of type $(A \multimap B) \otimes (\mathcal{C} \multimap D)$, it is unknown which function will be called first, so we represent them as parallel calls.

4.2.1 Event Structures

In order to represent executions of programs, we will use event structure diagrams like the one of Fig. 4.2. In this diagram, the nodes of the graph are events that can occur, the arrows are immediate causality relations between those events, and the wiggly lines are minimal conflicts between events. Immediate causality has to be interpreted in a conjunctive way, meaning that for an event to occur, every other event it depends on must have already occurred. Conversely, for an event to occur, none of the events it is in minimal conflict with must have already occurred. We want to formalise the underlying notions behind this diagram, however, while the relations $\rightarrowtriangle$ and $\sim$ are very practical for diagrams and examples, they are often impractical to use in proofs. We define event structures using causality, which is the reflexive and transitive closure of immediate causality, and consistency, which describes sets of events that are not in conflict with each others. We then redefine $\rightarrowtriangle$ and $\sim$ as syntactic sugar. We note that in Fig. 4.2, there are three boxes

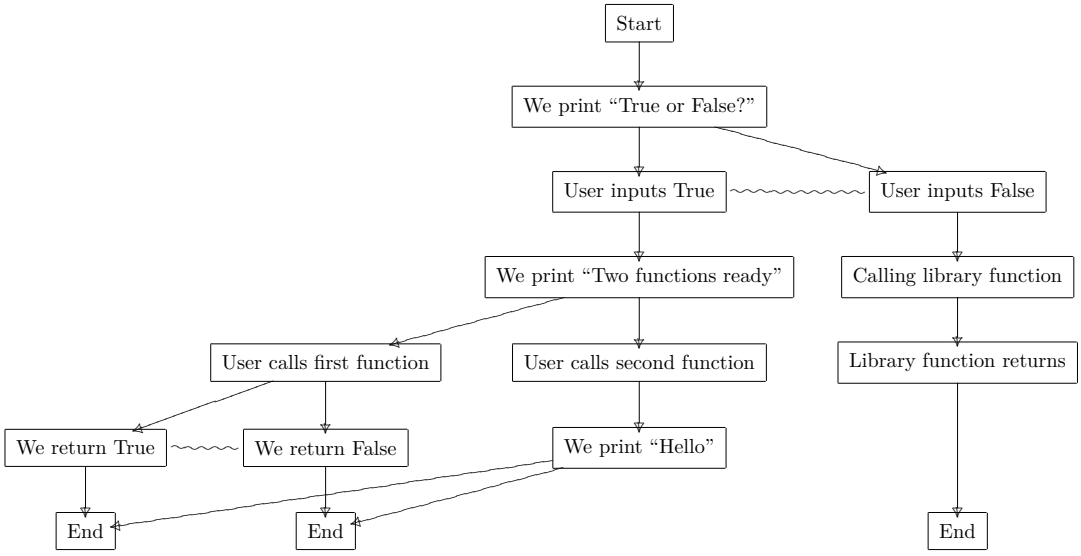


Figure 4.2: Example of Event Structure

"End". They have to be understood as three distinct events End_0, End_1 and End_2 where we kept implicit the indeces for simplicity of notation. In general, whenever we would want an event to be enabled in multiple different ways, we will split this event into copies, one for each of its possible causal histories.

Definition 4.2.1. *An event structure E is a triple $(|E|, \leq_E, \text{Con}_E)$ with:*

- *$|E|$ is a set of elements called* events.
- *$(|E|, \leq_E)$ is a partial order called* causality*, which has finite primes:*

$$\forall e \in |E|, \{e' \in |E| \mid e' \leq_E e\} \text{ is finite}$$

- *$\text{Con}_E \subseteq \mathcal{P}_{\text{fin}}(|E|)$ called* consistency *relation, which satisfies:*

$$\begin{array}{rcl} & & \varnothing \in \text{Con}_E \\ e \in |E| & \Longrightarrow & \{e\} \in \text{Con}_E \\ Y \subseteq X \in \text{Con}_E & \Longrightarrow & Y \in \text{Con}_E \\ e' \leq_E e \in X \in \text{Con}_E & \Longrightarrow & X \cup \{e'\} \in \text{Con}_E \end{array}$$

As in Section 2.2, we say that a subset of events $X \subseteq |E|$ is down-closed if for all $e' \leq_E e \in X$ we have $e' \in X$. We define the set of (finite) configurations of E as

$$\mathcal{C}(E) := \{x \in \text{Con}_E \mid x \text{ down-closed } \}$$

We say that a configuration is a prime if additionally it has a unique maximal element[1]. We use uppercase letters X, Y, Z for arbitrary sets of events, and reserve lowercase letters x, y, z for configurations. We remark that if $X \in \mathrm{Con}_E$, then there exists $x \in \mathcal{C}(E)$ such that $X \subseteq x$. Indeed, one can take $x = \bigcup_{e \in X} \{e' \in |E| \mid e' \leq_E e\}$.

For E, F event structures, we say that E is a substructure of F if $|E|$ is a down-closed subset of $|F|$, and if $\leq_E$ and Con_E are exactly the restriction of $\leq_F$ and Con_F to $|E|$. We list some standard notations, noting that we will keep the subscript E implicit when there is no ambiguity:

$[e]_E$	$:=$	$\{e' \in \lvert E\rvert \mid e' \leq_E e\} \in \mathcal{C}(E)$	prime configuration of e
$[e)_E$	$:=$	$\{e' \in \lvert E\rvert \mid e' <_E e\} \in \mathcal{C}(E)$	
$x \mathrel{-\!\!\subset}_E y$	:	$x, y \in \mathcal{C}(E)$ and $\exists e \in \lvert E\rvert, y = x \sqcup \{e\}$	one-step extension
$e \rightarrowtriangle_E e'$	:	$e <_E e'$ and $\nexists e'', e <_E e'' <_E e'$	immediate causality
$e \sim_E e'$	:	$\{e, e'\} \notin \mathrm{Con}_E$ and $\begin{cases} [e)_E \cup [e']_E \in \mathcal{C}(E) \text{ and} \\ [e]_E \cup [e')_E \in \mathcal{C}(E) \end{cases}$	minimal conflict

Proposition 4.2.2. *We say that an event structure E has binary conflict if it satisfies:*

$$\forall X \in \mathcal{P}_{\mathrm{fin}}(|E|), \forall e, e' \in X, \{e, e'\} \in \mathrm{Con}_E \implies X \in \mathrm{Con}_E$$

Or equivalently, for all $X, Y, Z \in \mathcal{P}_{\mathrm{fin}}(|E|)$, $X \cup Y, Y \cup Z, Z \cup X \in \mathrm{Con}_E \implies X \cup Y \cup Z \in \mathrm{Con}_E$. If E is an event structure with binary conflict, then E is entirely characterised by $(|E|, \rightarrowtriangle_E, \sim_E)$.

In all our diagrams, we consider only event structures with binary conflict. Hence, it is enough to only specify $\rightarrowtriangle$ and $\sim$ in diagrams. Another useful characterisation of event structures is through configurations.

Proposition 4.2.3. *An event structure E is entirely characterised by its set of (finite) configurations $\mathcal{C}(E)$.*

In fact, an event structure yields a transition system having as states the configurations, as initial state the empty configuration, as labels the events, and as transitions labelled by e the one-step extensions $x \mathrel{-\!\!\subset} x \sqcup \{e\}$.

We now define a category of event structures. Similarly to how event structures yield transition systems, total maps of event structures can be seen as functional simulations *i.e.*,

$$x \overset{e}{-\!\!-\!\!\subset} y \implies f(x) \overset{f(e)}{-\!\!-\!\!\subset} f(y)$$

Definition 4.2.4. *A partial map of event structures f from A to B is a partial function $f : |A| \rightharpoonup |B|$ which:*

[1] Equivalently, $x \in \mathcal{C}(E)$ is a prime if whenever $x \subseteq y \cup z$ for $y, z \in \mathcal{C}(E)$ we have $x \subseteq y$ or $x \subseteq z$.

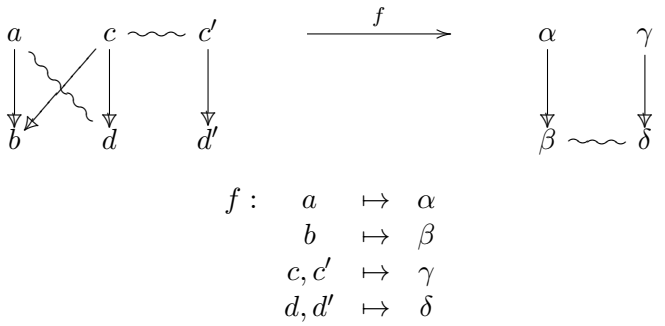


Figure 4.3: Example of map of event structures

Preserves Configurations $\forall x \in \mathcal{C}(A), f\,x \in \mathcal{C}(B)$.

Local Injectivity $\forall a, a' \in x \in \mathcal{C}(A), a, a' \in \mathrm{dom}(f)$ *and* $f(a) = f(a') \implies a = a'$.

Where $f\,X := \{f(e) \mid e \in X\}$. *The map is called total if the function* f *is total. Moreover, we say that this map is an inclusion if* A *is a substructure of* B *and* f *is the identity on events of* A.

Equivalently, f is a partial map of event structure if it is a partial function such that:

Preserves Consistency $\forall X \in \mathrm{Con}_A, f\,X \in \mathrm{Con}_B$.

Preserves Down-Closedness $\forall X \subseteq |A|$ down-closed , $f\,X$ is down-closed.

Local Injectivity $\forall a, a' \in X \in \mathrm{Con}_A, a, a' \in \mathrm{dom}(f)$ and $f(a) = f(a') \implies a = a'$.

See Fig. 4.3 for an example of total map of event structures. The configuration $\{a, b, c\}$ on the left hand side is sent by f to $\{\alpha, \beta, \gamma\}$ on the right hand side, which is also a configuration, and one can check similarly that every configuration on the left is sent to a configuration on the right. The map f is not injective, however it is still locally injective as even though $f(c) = f(c')$, c and c' never appear together in a configuration.

A notable property of maps of event structure is that they reflect conflict, and locally reflect causality. Formally:

Lemma 4.2.5. *If* f *is a partial map of event structures from* A *to* B, *then*

$$\forall a, a' \in \mathrm{dom}(f),\ \text{if } \{f(a), f(a')\} \notin \mathrm{Con}_B \text{ then } \{a, a'\} \notin \mathrm{Con}_A$$
$$\forall a, a' \in \mathrm{dom}(f),\ \text{if } \{a, a'\} \in \mathrm{Con}_A \text{ and } f(a) \leq_B f(a') \text{ then } a \leq_A a'$$

As a corollary if $f(a) \leq_B f(a')$ *and* $a \rightarrowtriangle_A a'$ *then* $f(a) \rightarrowtriangle_A f(a')$.

We write **ES** for the category of event structures and total maps of event structures. We will usually consider total maps, and will explicitly note when the maps used are partial, like in Definition 4.2.13. Isomorphisms in **ES**, are simply maps that "rename" the events of an event structure through a bijection, keeping the causality and the consistency unchanged. Similarly to how configurations characterise event structures, they also characterise isomorphisms.

Proposition 4.2.6. *For E and E' two event structures, if there exists a bijection $\iota : \mathcal{C}(E) \to \mathcal{C}(E')$ which preserves union and intersection, i.e., $\iota(x \cup y) = \iota(x) \cup \iota(y)$ and $\iota(x \cap y) = \iota(x) \cap \iota(y)$, then $f : E \to E'$ given by $f(e) = \max(\iota([e]))$ is defined and is an isomorphism of event structures.*

We note that preserving unions and intersections is equivalent to being an order-isomorphism for the inclusion: $\iota(x) \subseteq \iota(y) \iff x \subseteq y$.

In game semantics, we will use maps of event structures in the definition of strategies. If we assume that the game is an event structure E, then the strategy will be represented by another event structure S, often in examples with the same events as E, and a total map of event structures $\sigma : S \to E$, often in examples the identity function on events. More precisely, S will not be E whenever we needed to duplicate an event of E to authorize it in several causal histories. The definition of a map of event structures ensures that the strategy abides by the rules of the game:

- Preservation of down-closedness ensures that the strategy never plays a move not yet available.
- Preservation of consistency ensures that the strategy never plays a move no longer available.

As specified in the full definition in Section 4.4.2, strategies come with additional restrictions we cannot express yet because of the lack of polarities distinguishing Player moves from Opponent moves.

4.2.2 Parallel Composition and Coproduct

Definition 4.2.7. *For two event structures A and B, we define their parallel composition $A \parallel B$ and their coproduct $A \oplus B$ as follows:*

$$\begin{array}{rcccl} |A \parallel B| & := & |A| \uplus |B| & =: & |A \oplus B| \\ \leq_{A \parallel B} & := & \leq_{|A| \uplus |B|} & =: & \leq_{A \oplus B} \end{array}$$

$$\begin{array}{rcl} \mathrm{Con}_{A \parallel B} & := & \{X \uplus Y \mid X \in \mathrm{Con}_A, Y \in \mathrm{Con}_B\} \\ \mathrm{Con}_{A \oplus B} & := & \{X \uplus \varnothing \mid X \in \mathrm{Con}_A\} \cup \{\varnothing \uplus Y \mid Y \in \mathrm{Con}_B\} \end{array}$$

Both operations have the empty event structure $(\varnothing, \varnothing, \{\varnothing\})$ as their unit (up to isomorphism), which we write $\varnothing$. Both operations have infinite variants $\|_{i \in I} A_i$ and $\bigoplus_{i \in I} A_i$

for any set I, with $|\parallel_{i\in I} A_i| = |\bigoplus_{i\in I} A_i| = \{(i,a) \mid i \in I, a \in A_i\}|$ and the order and consistency are induced from the A_i as in the binary case.

We note that $\mathcal{C}(A \parallel B) = \{x \uplus y \mid x \in \mathcal{C}(A), y \in \mathcal{C}(B)\}$. We will usually write $x \parallel y$ instead of $x \uplus y$ when considering the configurations of $\mathcal{C}(A \parallel B)$. We will also write $a \in_{\parallel} |A|$ as a shorthand for $a = (0, a')$ with $a' \in |A|$, and $b \in_{\parallel} |B|$ as a shorthand for $b = (1, b')$ with $b' \in |B|$.

Similarly, since $\mathcal{C}(A \oplus B) = \{x \uplus \varnothing \mid x \in \mathcal{C}(A)\} \cup \{\varnothing \uplus y \mid y \in \mathcal{C}(B)\}$, we write $x \oplus \varnothing$ and $\varnothing \oplus y$ for the configurations of $A \oplus B$, and write $a \in_{\oplus} A$ or $b \in_{\oplus} B$ whenever $a = (0, a')$ with $a' \in |A|$ or $b = (1, b')$ with $b' \in |B|$.

Theorem 4.2.8. *The category* **ES** *forms an SMC for $\parallel$, and a cocartesian category for the coproduct $\oplus$. The bifunctor $(- \parallel -)$ and the copairing $[-;-]$ are given by:*

$$(f \parallel g) : \begin{cases} (0,a) \mapsto (0, f(a)) \\ (1,b) \mapsto (1, g(b)) \end{cases} \qquad [f;g] : \begin{cases} (0,a) \mapsto f(a) \\ (1,b) \mapsto g(b) \end{cases}$$

ES *also has arbitrary coproducts, with the copairing $[f_i \mid i \in I]$ given by:*

$$[f_i \mid i \in I] : (i,a) \mapsto f_i(a)$$

We note that this category is not a distributive SMC. Indeed, $A \parallel (B \oplus C)$ and $(A \parallel B) \oplus (A \parallel C)$ do not have the same number of events (unless A is empty). While the operation $\oplus$ will correspond to the type constructor $\oplus$, we did not provide in this section any construction corresponding to $\multimap$ or $\otimes$. We postpone them to Section 5.5, as their formal definition relies on event structures being of a particular shape.

4.2.3 Interactive Composition

Similarly to how a strategy on a game E will be represented by a map $\sigma : S \to E$, where S is represent the behaviour of the strategy, E represent the rules of the game, and σ guaranty that the strategy abides by the rules, a strategy from the game A to the game B will be represented by a map $\sigma : S \to A \parallel B$.

For example, looking at the leftmost column of Fig. 4.4, we have a map $g : G \to B \parallel C$, where B is the event structure representing the type $\mathbf{1}$, with only one event corresponding to the execution of the term $()$, and C is the event structure representing the type **bit**, with two events in conflict corresponding to the two boolean values. The event structure G represents a term that reads the context containing data of type $\mathbf{1}$, and then outputs the boolean $\mathtt{tt}$, in other words

$$x : \mathbf{1} \vdash_{\mathbb{L}} x; \mathtt{tt} : \mathbf{bit}$$

The map g specifies that the event $\mathtt{tt}$ of G indeed corresponds to the move $\mathtt{tt}$ of the game, and likewise for $\star$. Looking at the central column, we have a map $f : F \to A \parallel B$, with the

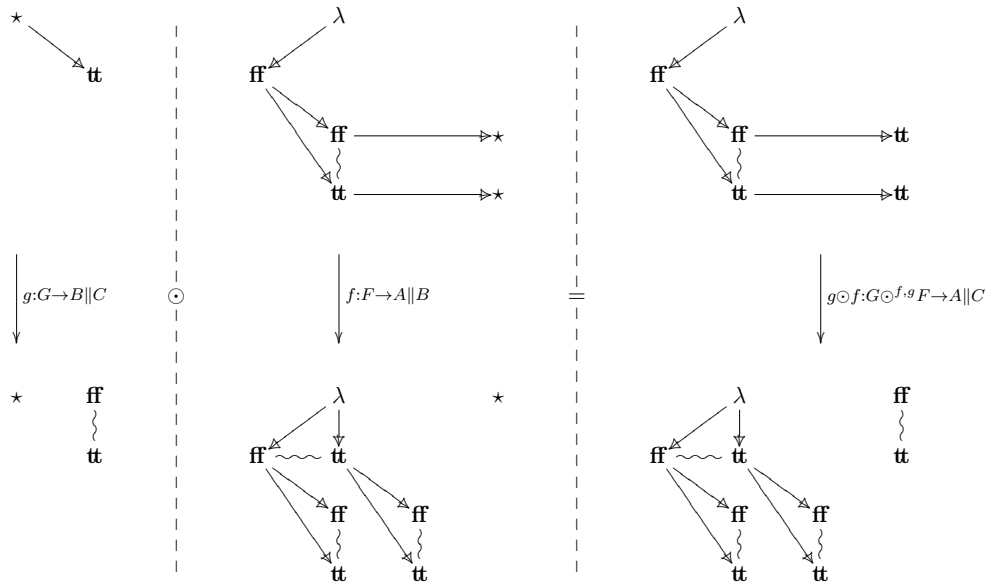


Figure 4.4: Example of Interactive Composition

same B, and A representing functions from booleans to booleans. The event structure A starts with an event λ which is an artefact of call-by-value semantics, one can understand it as representing "the function is defined, and ready to be called". This event λ is followed by two events in conflict representing the possible inputs, and then events representing the possible outputs. The event structure F represents a term that reads the function present in its context, inputs the value false, reads the output of the function, and then returns (), in other words

$$f_0 : \mathbf{bit} \multimap \mathbf{bit} \vdash_{\mathbb{L}} \mathbf{if}\ f_0\ \mathbf{ff}\ \mathbf{then}\ ()\ \mathbf{else}\ ()$$

Once again, the map f is here to specify which event of F corresponds to which move of the game. The goal of this subsection is to manage to compute the rightmost column of this figure, which corresponds to the composition of the two terms:

$$f_0 : \mathbf{bit} \multimap \mathbf{bit} \vdash_{\mathbb{L}} (x; \mathbf{tt})\{x \leftarrow \mathbf{if}\ f_0\ \mathbf{ff}\ \mathbf{then}\ ()\ \mathbf{else}\ ()\}$$

More precisely, in this subsection, we explain how one can make a map of event structures $f : F \to A \parallel B$ interact with another map $g : G \to B \parallel C$ in order to obtain a map $h : H \to A \parallel C$. This construction is called interactive composition, or "parallel interaction plus hiding". It is made in two steps: the interaction through pullbacks, and the hiding.

Pullbacks

We start with some categorical definitions. The notion of pullback is very similar to the notion of categorical product, in the sense that both of them are categorical limits.

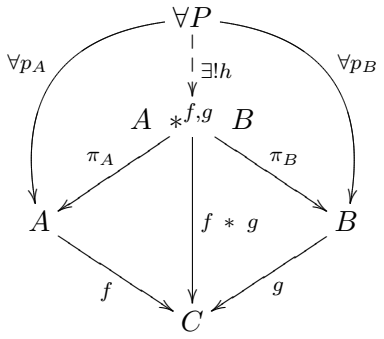
Definition 4.2.9. *In a category $\mathcal{C}$, a pre-pullback of a map $f \in \mathcal{C}(A, C)$ and $g \in \mathcal{C}(B, C)$ is a triple (P, p_A, p_B) with $p_A \in \mathcal{C}(P, A)$ and $p_B \in \mathcal{C}(P, B)$ such that*

$$f \circ p_A = g \circ p_B$$

A pullback is a pre-pullback $(A \ {}_*{}^{f,g} \ B, \pi_A, \pi_B)$ such that for every pre-pullback (P, p_A, p_B), there exists a unique map $h \in \mathcal{C}(P, A \ {}_*{}^{f,g} \ B)$ such that

$$\pi_A \circ h = p_A \qquad \pi_B \circ h = p_B$$

We write this morphism $f \ * \ g$, and sum up this property in the following diagram:



It follows that $A \ {}_*{}^{f,g} \ B$ is unique up to isomorphism if it exists.

We explain what we mean in this diagram in more detail: for all objects and morphisms inserted at the positions of labels starting with a $\forall$, if the diagram ignoring dashed arrows commutes, then there exists unique morphisms inserted at the position of labels starting with $\exists!$ such that the diagram including dashed arrows commutes.

Proposition 4.2.10. *The category* **ES** *has all pullbacks.*

We refer to [Win07, Win11] for a proof. Pullbacks in **ES** being syntactically complex to handle, the standard proof for existence of pullbacks in **ES** defines them in the category of certain families of configurations, called stable families, and recovers pullbacks in **ES** using an adjunction. The complexity of building pullbacks in **ES** will not hinder us, as we only use the universal property of pullbacks and never need the details of the syntactical definition.

We provide in Fig. 4.5 an example of pullback of **ES**. In this example, the event structure A is graphically a rectangle made of conflicts, the event structure B is a triangle made of causalities and conflicts, and the event structure $A \ {}_*{}^{f,g} \ B$ is a small "house" containing the conflicts and causalities of both A and B. This example highlights that as a first approximation, once can see a pullback as just "overlapping" the two event structures A and B to obtain a single event structure $A \ {}_*{}^{f,g} \ B$ which contains the causalities and

conflicts of both A and B. This intuition is incomplete, but holds whenever the maps f and g are injective.

In the example, the event structure C has only a single conflict between b and c, but this conflict is irrelevant, as we would have obtained the same $A \ *^{f,g} \ B$ without it. In general when computing the pullback $A \ *^{f,g} \ B$, we can disregard the causalities and conflicts of C and obtain the same result.

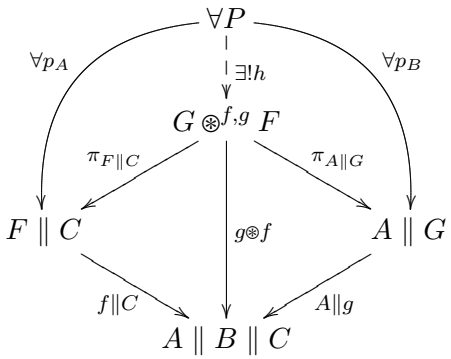
Interaction

We now use the notion of pullback to define the first step of "parallel interaction plus hiding". The interaction can be seen as a composition, but where we remember the intermediate values. It is achieved with pullbacks, the idea being to "make the first strategy play against the second one on their common game".

Definition 4.2.11 (Interaction)**.** *For two maps of event structures $f : F \to A \parallel B$ and $g : G \to B \parallel C$, we define the event structure $G \circledast^{f,g} F$ and the map $g \circledast f : G \circledast^{f,g} F \to A \parallel B \parallel C$ as:*

$$\begin{array}{rcl} G \circledast^{f,g} F & := & (F \parallel C) \ *^{f\parallel C, A\parallel g} \ (A \parallel G) \\ g \circledast f & := & (f \parallel C) \ * \ (A \parallel g) \end{array}$$

We sum it up in the following diagram:



We provide an example in Fig. 4.6. The right-most column is the result of the interaction. Looking at the event structure $G \circledast^{f,g} F$ at the top right of the figure, one can recognise a fragment corresponding to F, between the events λ and the two events $\star$. One can also recognise two fragments corresponding to G, starting with the event $\star$ and ending with $\mathbf{tt}$. The events $\star$ are at the junction between the fragments of F and the fragments of G. This is because $\star$ is part of the event structure B in $(A \parallel B \parallel C)$. Since F has two copies of $\star$, when joining G to F, we had to duplicate G: one copy for every $\star$.

We note that since $(- \parallel -)$ is not strictly associative, $(A \parallel B \parallel C)$ is only defined up to isomorphism, and we should post-compose $f \parallel C$ and $A \parallel g$ with adequate isomorphisms, this is in general left implicit. For simplicity, we assume that we choose $(A \parallel B \parallel C)$ such that $(A \parallel C)$ is a substructure of it.

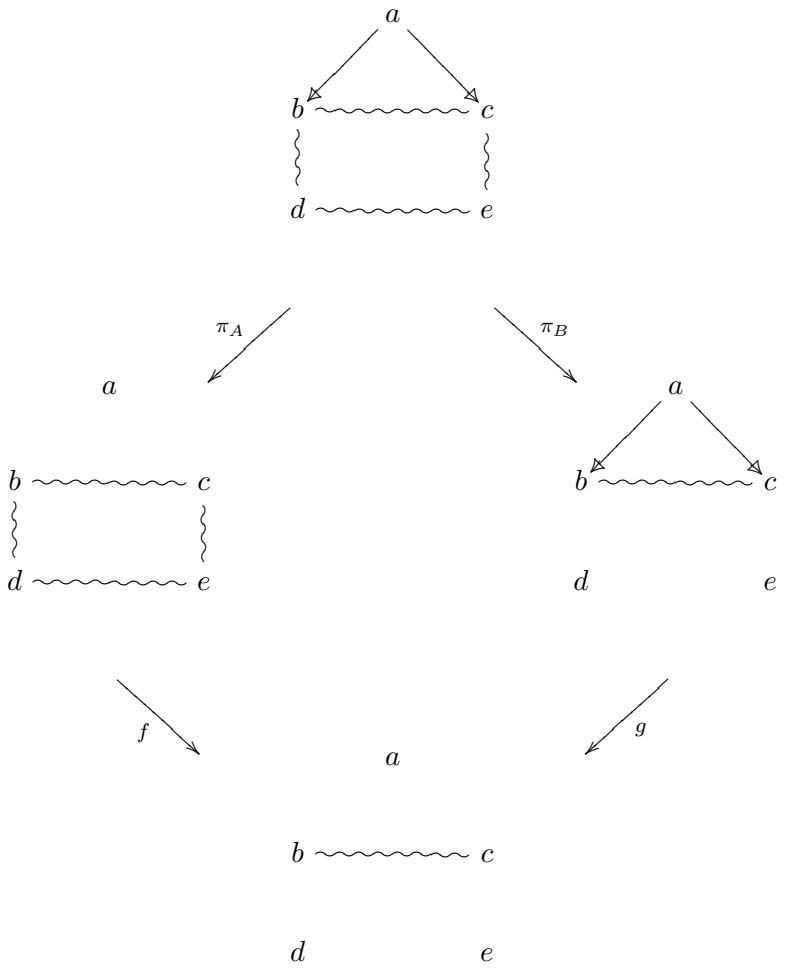


Figure 4.5: Example of a pullback of event structures

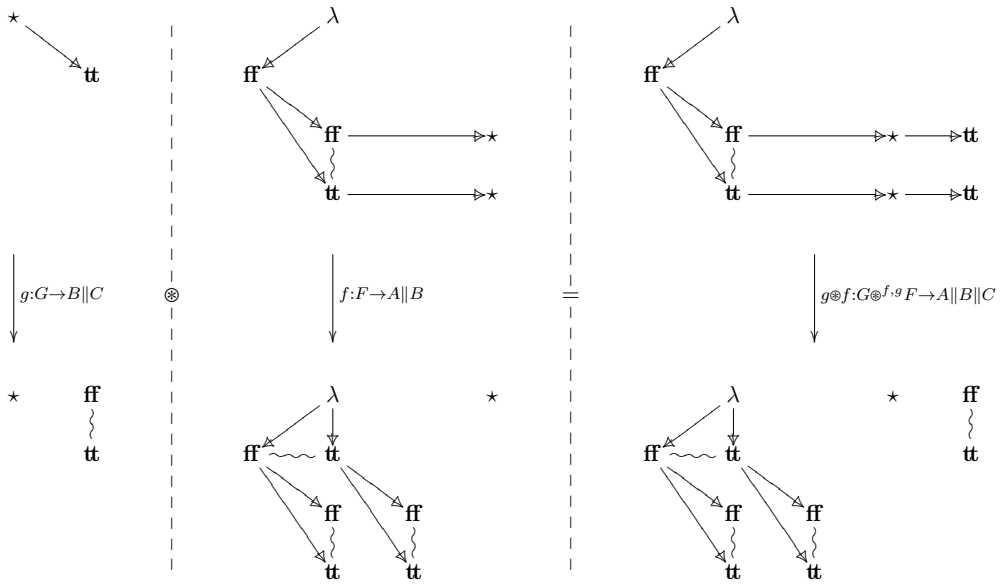


Figure 4.6: Example of Interaction

Proposition 4.2.12. *The interaction is associative up to isomorphism, i.e.,* $H \circledast^{g,h} (G \circledast^{f,g} F) \cong (H \circledast^{g,h} G) \circledast^{f,g} F$*, and the following diagram commutes:*

$$\begin{array}{ccc} H \circledast^{g,h} (G \circledast^{f,g} F) & \xleftarrow{\cong} & (H \circledast^{g,h} G) \circledast^{f,g} F \\ \Big\downarrow{\scriptstyle h \circledast (g \circledast f)} & & \Big\downarrow{\scriptstyle (h \circledast g) \circledast f} \\ (A \parallel B \parallel C) \parallel D & \xleftarrow{\cong} & A \parallel (B \parallel C \parallel D) \end{array}$$

The proof follows from the definition of the pullback: one just needs to remark that $H \circledast^{g,h} (G \circledast^{f,g} F)$ is a pre-pullback of $A \parallel (h \circledast g)$ and $f \parallel C \parallel D$, and that $(H \circledast^{g,h} G) \circledast^{f,g} F$ is a pre-pullback of $A \parallel B \parallel h$ and $(g \circledast f) \parallel D$.

Hiding

Another central operation for game semantics is hiding, which allows us to "forget" some events corresponding to intermediate computations.

Definition 4.2.13. *A partial map of event structures* $H : A \multimap B$ *is said to be a hiding if* $|B| \subseteq |A|$*, with* $\leq_B$ *and* Con_B *being exactly the restriction of* $\leq_A$ *and* Con_A *to* $|B|$*, and* H *is the identity on events.*

By construction, a hiding is intersection and union-preserving on configurations. We also note that whenever $|B|$ is a down-closed subset of $|A|$, the hiding is a left-inverse[2] of

[2] A left-inverse, or retraction, of a map $f \in \mathcal{C}(A, B)$ is a map $g \in \mathcal{C}(B, A)$ such that $g \circ f = \mathbf{id}_A$.

the substructure map from B to A.

Proposition 4.2.14. *For $f : A \to B$ a map and $H : B \rightharpoonup B'$ a hiding, there exists some unique event structure A', hiding $H' : A \rightharpoonup A'$ and map $f' : A' \to B'$ such that:*

$$H \circ f = f' \circ H'$$

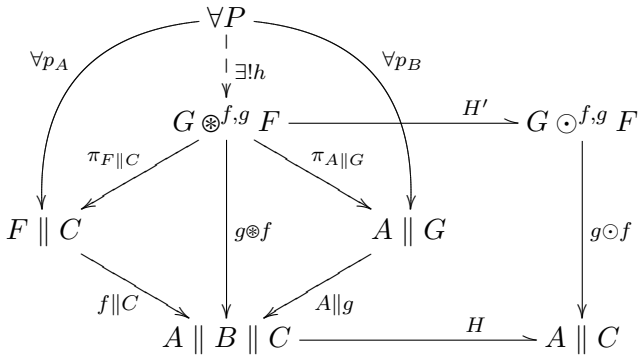
The proof is pretty straightforward, as we just take A' such that $|A'| = f^{-1}\,|B'|$.

Interactive Composition

We consider two maps of event structures $f : F \to A \parallel B$ and $g : G \to B \parallel C$. We have $G \circledast^{f,g} F$ and the map $g \circledast f : G \circledast^{f,g} F \to A \parallel B \parallel C$ as previously defined. Since $A \parallel C$ is a substructure of $A \parallel B \parallel C$, we can use Proposition 4.2.14 and obtain a uniquely defined $G \odot^{f,g} F$ and $g \odot f$ such that:

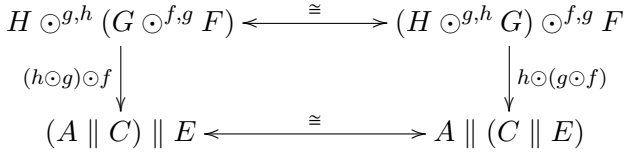
$$H \circ (g \circledast f) = (g \odot f) \circ H'$$

We sum it up in the following diagram:



In Fig. 4.7, we show that to obtain interactive composition described in Fig. 4.4 from the interaction described in Fig. 4.6, we simply remove from the diagrams all the events coming from the event structure B, *i.e.*, we remove all the events $\star$.

Proposition 4.2.15. *The interactive composition is associative up to isomorphism, i.e., $H \odot^{g,h} (G \odot^{f,g} F) \cong (H \odot^{g,h} G) \odot^{f,g} F$ and the following diagram commutes:*



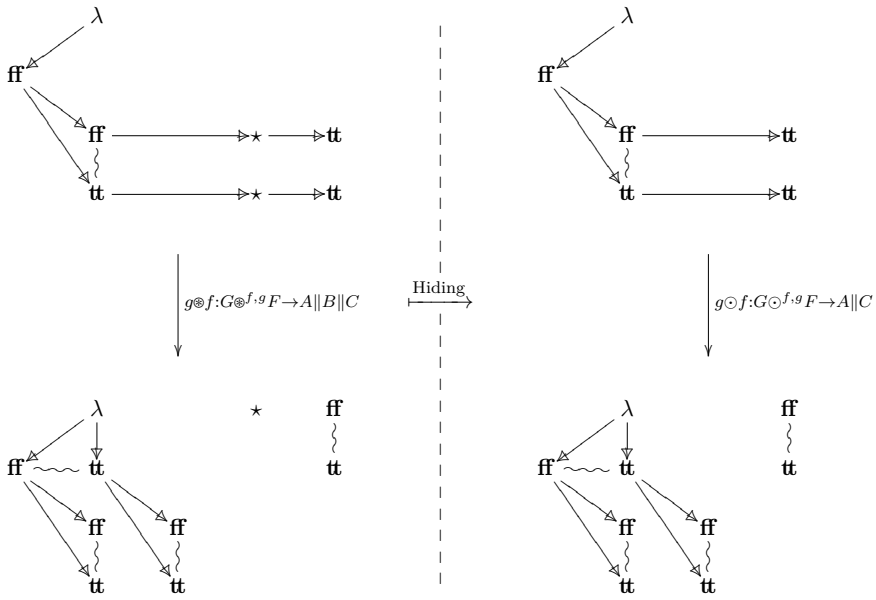


Figure 4.7: From Interaction to Interactive Composition

The proof follows from the associativity up to isomorphism of the interaction and Proposition 4.2.14. This interactive composition does not yet define a category, as we have no identity. Defining an identity for the interactive composition is non-trivial. In Section 4.4.2, after having added polarities to our event structures and some additional restrictions relying on those polarities, we will obtain a category with for composition the interactive composition.

4.3 Matching Pairs of Configurations

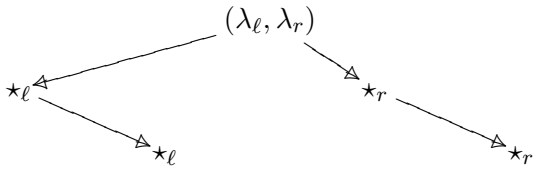
In this section, we focus on a tool that will ease the definitions and proof in the following chapter: the characterisation of configurations of the interaction and of the interactive composition.

4.3.1 Examples

In this section, we will use some guiding examples we describe here. In examples, we take for A, B and C the event structures corresponding to $\mathbf{1}$, $(\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{1})$ and $\mathbf{1}$ respectively. In other words, A and C are both the event structure with a single event $\star$,

and B is:

$$(\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{1})$$



The event structure B starts with an event $(\lambda_\ell, \lambda_r)$ to be understood as "the two functions are defined and ready to be called". The four other events correspond to calling those functions on input (), and those functions returning () as an output. We describe in Fig. 4.8 the maps $f : F \to A \parallel B$ and $g : G \to B \parallel C$. The map f represents the term that:

- Reads its context of type $\mathbf{1}$.
- Then defines two functions.
- Then whenever a function is called on input (), returns ().

The map g represents the term that:

- Reads its context of type $(\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{1})$.
- Then calls the first function of the context, and waits for it to return.
- Then calls the second function of the context, and waits for it to return.
- Then outputs ().

We describe in Fig. 4.9 the maps $f' : F' \to A \parallel B$ and $g' : G' \to B \parallel C$. Both maps do not correspond to terms of Λ, but could correspond to terms in a language with parallel threads and shared memory. The map f' represents the system that:

- Reads its context of type $\mathbf{1}$.
- Then defines two functions.
- Then whenever a function is called on input (), does nothing for now.
- Then whenever both functions have been called on input (), returns () for both functions.

This last item requires some synchronisation between two functions being called in parallel. This can be done within a language with shared memory, but not within Λ. The map g' represents the system that:

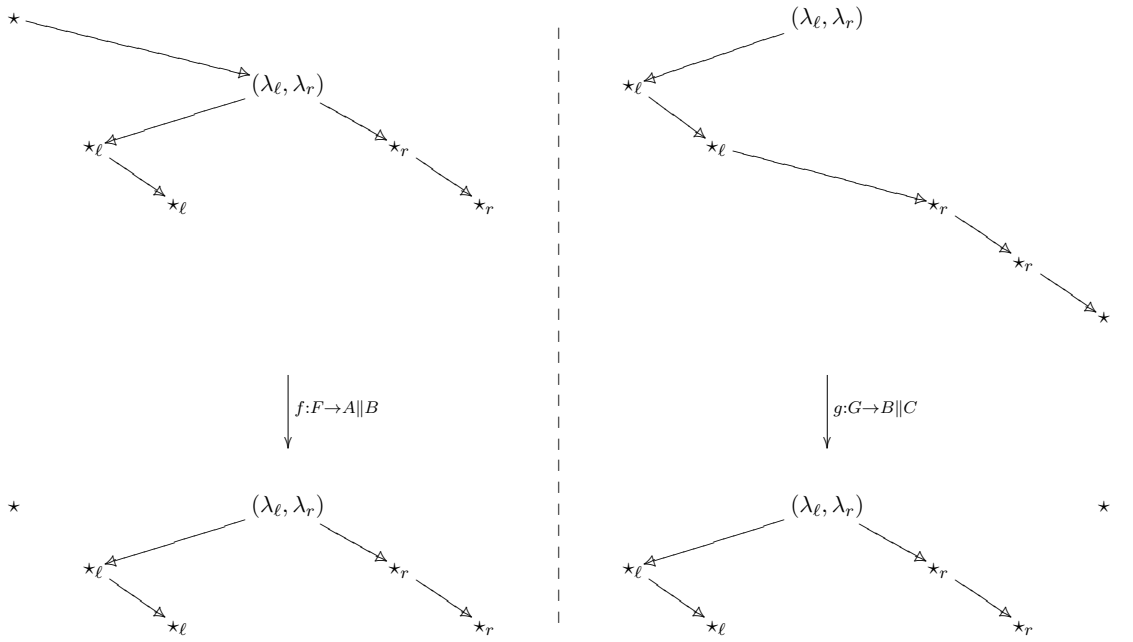


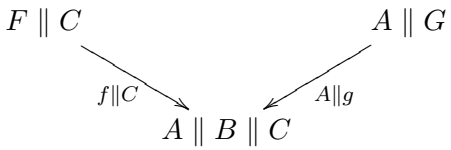
Figure 4.8: The maps of event structures f and g, with the event structure F and G at the top, and $A \parallel B$ and $B \parallel C$ at the bottom.

- Reads its context of type $(\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{1})$.
- Then calls in parallel both functions on input ().
- Whenever the first function returns, it does nothing.
- Whenever the second function returns, it outputs () (even if the first has not yet returned).

Similarly, this system cannot be represented by a term in Λ, as we cannot start a computation and continue without waiting for its result.

4.3.2 Definition of Matching Pairs

We consider $f : F \to A \parallel B$ and $g : G \to B \parallel C$ two maps of event structures. We have



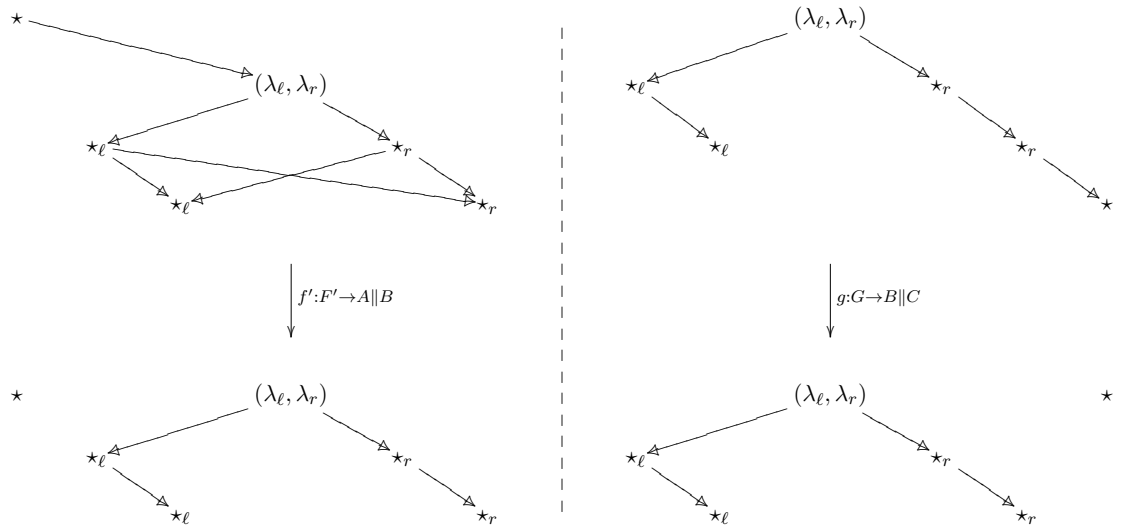


Figure 4.9: The maps of event structures f' and g', with the event structure F' and G' at the top, and $A \parallel B$ and $B \parallel C$ at the bottom.

For any $x \in \mathcal{C}(F)$, we write $f\,x = x_A \parallel x_B$. Similarly, for every $y \in \mathcal{C}(G)$, we write $g\,y = y_B \parallel y_C$.

Definition 4.3.1. *Two configurations $x \in \mathcal{C}(F)$ and $y \in \mathcal{C}(G)$ are said*

Matching *if $x_B = y_B$.*

Matching Compatible *if moreover the induced pre-order over $x_A \parallel x_B \parallel y_C$ is acyclic, i.e., an order. This pre-order is obtained as follows: we note that $(x \parallel y_C, \leq_{F \parallel C})$ and $(x_A \parallel y, \leq_{A \parallel G})$ are two posets, we consider their image by $f \parallel C$ and $A \parallel g$ respectively, and then the transitive closure of the union of both images. The transitive closure of the union of two posets might not be a poset.*

Minimal Matching Compatible *if moreover for every (x', y') matching compatible, with $x'_A = x_A$, $y'_C = y_C$, $x' \subseteq x$ and $y' \subseteq y$, we have $x' = x$ and $y' = y$.*

Under some reasonable conditions on f and g discussed in Section 6.2.1, matching pairs are always compatible, in other words the interaction does not create any deadlocks. As discussed in Lemma A.2.2, matching compatible pairs of configurations that satisfy the $\oplus$-covered condition will always be minimal.

Example 4.1 Simple Case

If we look at the configurations $|F| \in \mathcal{C}(F)$ and $|G| \in \mathcal{C}(G)$ containing all the events of F and G respectively, this pair of configurations is

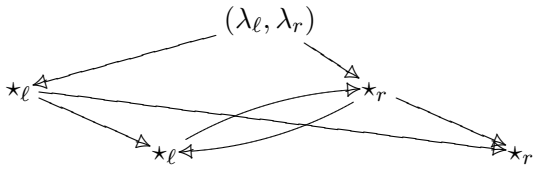
- Matching, as they project to the same configuration $|B|$ of B.
- Matching compatible, as there is no deadlock.
- Minimal matching compatible. Indeed, if we consider any pair of matching compatible configuration (x, y) such that $y_C = \{\star\}$ and $x_A = \{\star\}$, then looking at G we are forced to have $y = |G|$, so $y_B = |B|$. Since they are matching, it follows that $x_B = |B|$, so $x = |F|$. We then have $(x, y) = (|F|, |G|)$.

Example 4.2 Deadlock Case

If we look at the configurations $|F'| \in \mathcal{C}(F')$ and $|G| \in \mathcal{C}(G)$ containing all the events of F' and G respectively, this pair of configurations is

- Matching, as they project to the same configuration $|B|$ of B.
- **Not** matching compatible, as the induced pre-order on $|B|$ has a cycle:

$$(\mathbf{1} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{1})$$



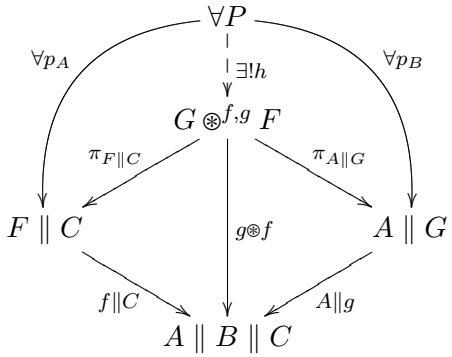
Example 4.3 Non-Minimal Case

If we look at the configurations $|F| \in \mathcal{C}(F)$ and $|G'| \in \mathcal{C}(G')$ containing all the events of F and G' respectively, this pair of configurations is

- Matching, as they project to the same configuration $|B|$ of B.
- Matching compatible, as there is no deadlock.
- **Not** minimal matching compatible, as $(\{\star, (\lambda_\ell, \lambda_r), \star_r, \star_r\}, \{(\lambda_\ell, \lambda_r), \star_r, \star_r, \star\})$ is a matching compatible pair which is smaller while having the same projections on A and C.

4.3.3 Matching Pairs in the Interaction

We keep the same notations as in previous subsections. We recall the diagram that defines $g \circledast f$:



We say that $z \in \mathcal{C}(G \circledast^{f,g} F)$ is the interaction of $x \in \mathcal{C}(F)$ and $y \in \mathcal{C}(G)$ if $\pi_{F\|C}\, z = x \parallel y_C$ and $\pi_{A\|G}\, z = x_A \parallel y$.

Lemma 4.3.2. *We have the following properties*

- *For every pair $(x, y) \in \mathcal{C}(F) \times \mathcal{C}(G)$, there exists* at most one $z \in \mathcal{C}(G \circledast^{f,g} F)$ *which is the interaction of x and y. When it exists, we write it $y \circledast x$.*

- *For every $z \in \mathcal{C}(G \circledast^{f,g} F)$, there exists* exactly one *pair (x, y) such that $z = y \circledast x$. Those configurations x and y are the projections of z.*

Proposition 4.3.3. *We consider $f : F \to A \parallel B$ and $g : G \to B \parallel C$ two maps of event structures. We have a bijection between the set of configurations $z \in \mathcal{C}(G \circledast^{f,g} F)$ and the set of matching compatible pairs $(x, y) \in \mathcal{C}(F) \times \mathcal{C}(G)$. The bijection is given by $z = y \circledast x$.*

This proposition is at the core of the notion of matching compatible pairs, and allows us to work on the interaction of strategies, without having to explicitly use its definition through the pullback. We will implicitly use this proposition when ranging over the configurations of $G \circledast^{f,g} F$ saying for example "for $y \circledast x \in \mathcal{C}(G \circledast^{f,g} F)$, ...".

Lemma 4.3.4. *The operation $\circledast$ is union-preserving and intersection-preserving, i.e., if $(x, y), (x', y')$ are matching compatible, and $x \cup x'$ and $y \cup y'$ are configurations, then $(x \cup x', y \cup y')$ is matching compatible and $(y \cup y') \circledast (x \cup x') = (y \circledast x) \cup (y' \circledast x')$; and a similar property for the intersection.*

Example 4.4 Interaction without Deadlocks

The Figs. 4.10 and 4.12 describe the interaction of f with g and f with g'. As no deadlock is involved, all the matching pairs are compatible, so the configurations of $\mathcal{C}(G \circledast^{f,g} F)$ and $\mathcal{C}(G' \circledast^{f,g'} F)$ are in one-to-one correspondence with the pairs of matching configurations of $\mathcal{C}(G) \times \mathcal{C}(F)$ and $\mathcal{C}(G') \times \mathcal{C}(F)$ respectively. For example in Fig. 4.10, the configuration $\{\star, (\lambda_\ell, \lambda_r), \star_l\} \in \mathcal{C}(G \circledast^{f,g} F)$ comes from the matching pair $(\{\star, (\lambda_\ell, \lambda_r), \star_l\}, \{(\lambda_\ell, \lambda_r), \star_l\}) \in \mathcal{C}(F) \times \mathcal{C}(G)$.

Example 4.5 Interaction for the Deadlock Case

The Fig. 4.11 describes the interaction of f' and g. This time, not all pairs of matching configurations are compatible, which is why the interaction is so small. For example the matching pair $(\{\star, (\lambda_\ell, \lambda_r), \star_l, \star_r\}, \{(\lambda_\ell, \lambda_r), \star_l, \star_r\}) \in \mathcal{C}(F') \times \mathcal{C}(G)$ is not compatible as we have $\star_\ell \geq \star_r$ on one side and $\star_\ell \leq \star_r$ on the other.

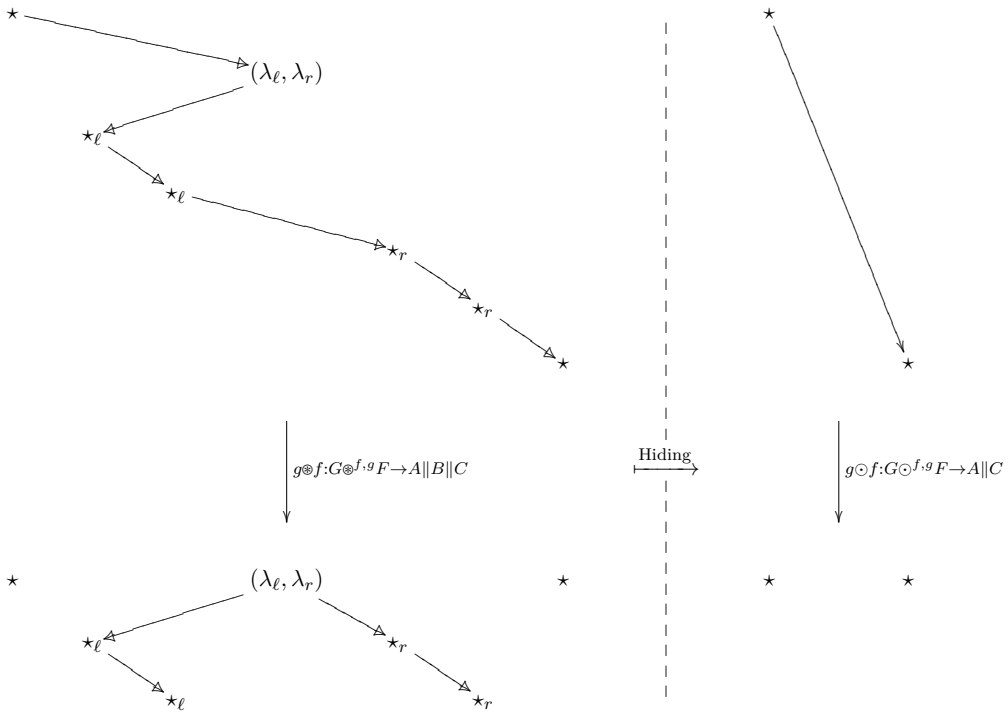


Figure 4.10: The interaction and interactive composition of f and g

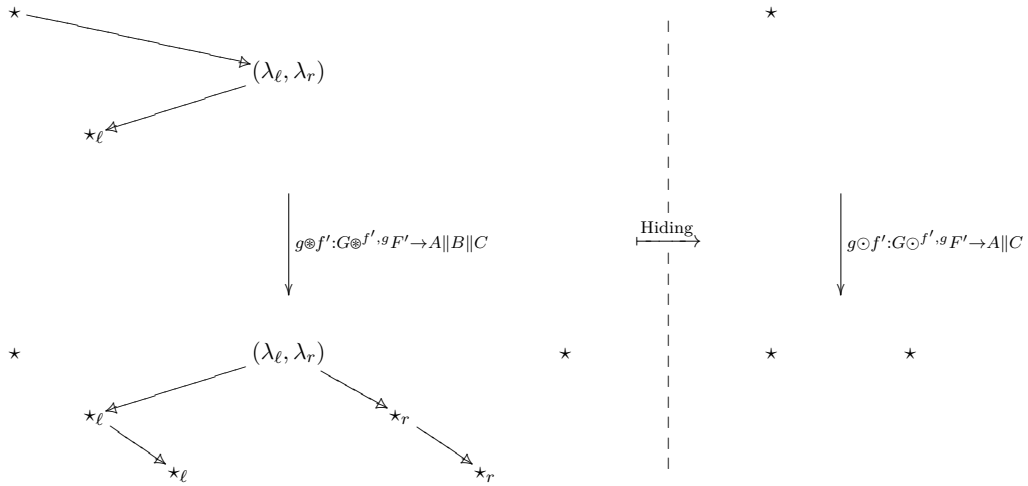


Figure 4.11: The interaction and interactive composition of f' and g

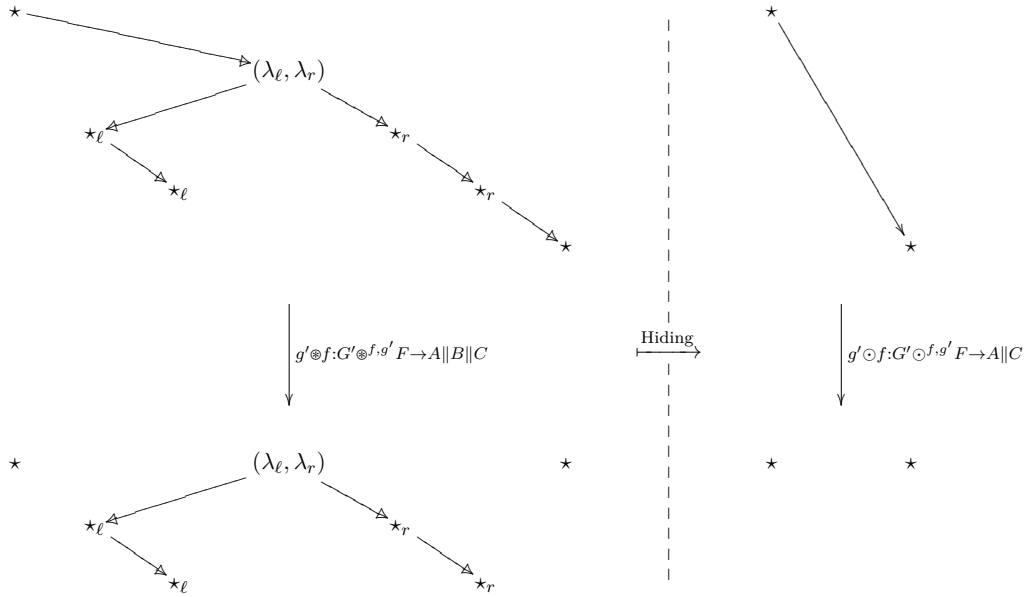
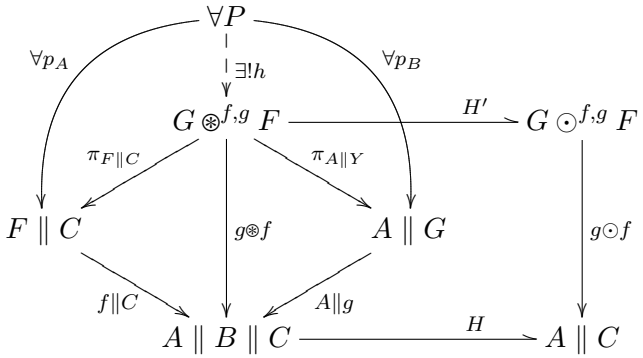


Figure 4.12: The interaction and interactive composition of f and g'

4.3.4 Interactive Composition

We keep the same notations as in the previous subsection. We recall the diagram that defines $g \odot f$:



Since H' is a hiding map, it is injective, and the events of $G \odot^{f,g} F$ can be seen as events of $G \circledast F$. For $z \in \mathcal{C}(G \odot^{f,g} F)$, we write $[z]_\circledast$ for the down-closure of the pre-image of z, *i.e.*, $[H'^{-1}(z)]_{G \circledast^{f,g} F}$. Since H' reflects consistency, we have $[z]_\circledast \in \mathcal{C}(G \circledast^{f,g} F)$. We note that $[z]_\circledast$ is the smallest configuration z' of $G \circledast^{f,g} F$ such that $H'(z') = z$.

We say that $z \in \mathcal{C}(G \odot^{f,g} F)$ is the interactive composition of $x \in \mathcal{C}(F)$ and $y \in \mathcal{C}(G)$ if $[z]_\circledast = y \circledast x$.

Lemma 4.3.5. *We have the following properties*

- *For every pair $(x, y) \in \mathcal{C}(F) \times \mathcal{C}(G)$, there exists* at most one *$z \in \mathcal{C}(G \odot^{f,g} F)$ which is the interactive composition of x and y. When it exists, we write it $y \odot x$.*
- *For every $z \in \mathcal{C}(G \odot^{f,g} F)$, there exists* exactly one *pair (x, y) such that $z = y \odot x$.*

Proposition 4.3.6. *We consider $f : F \to A \parallel B$ and $g : G \to B \parallel C$ two maps of event structures. We have a bijection between the set of configurations $z \in \mathcal{C}(G \odot^{f,g} F)$ and the set of minimal matching compatible pairs $(x, y) \in \mathcal{C}(F) \times \mathcal{C}(G)$. The bijection is given by $z = y \odot x$.*

As for the corresponding proposition in the case of the interaction, we will implicitly use this proposition when ranging over the configurations of $G \odot^{f,g} F$ saying for example "for $y \odot x \in \mathcal{C}(G \odot^{f,g} F)$, ...".

Lemma 4.3.7. *The operation $\odot$ is union-preserving and intersection-preserving, i.e., if $(x, y), (x', y')$ are minimal matching compatible, and $x \cup x'$ and $y \cup y'$ are configurations, then $(x \cup x', y \cup y')$ is minimal matching compatible and $(y \cup y') \odot (x \cup x') = (y \odot x) \cup (y' \odot x')$; and a similar property for the intersection.*

Example 4.6 Interactive Composition

The Figs. 4.10 to 4.12 describe the interactive composition of f with g, f' with g and f with g' respectively.

- Looking at $g \odot f$, the pairs of minimal matching configurations are $(\varnothing, \varnothing)$, $(\{\star\}, \varnothing)$ and $(|F|, |G|)$. They correspond to the configurations $\varnothing, \{\star\}, \{\star, \star\} \in \mathcal{C}(G \odot^{f,g} F)$ respectively.
- Looking at $g \odot f'$, because of the deadlocks, $(|F'|, |G|)$ is not a matching compatible pair, so the only minimal matching configurations are $(\varnothing, \varnothing)$ and $(\{\star\}, \varnothing)$, corresponding to $\varnothing, \{\star\} \in \mathcal{C}(G \odot^{f',g} F')$ respectively.
- Looking at $g' \odot f$, while there is no deadlock, $(|F|, |G'|)$ is not a minimal matching compatible pair as it fails at minimality. The minimal matching configurations are $(\varnothing, \varnothing)$,$(\{\star\}, \varnothing)$ and $(\{\star, (\lambda_\ell, \lambda_r), \star_r, \star_r\}, \{(\lambda_\ell, \lambda_r), \star_r, \star_r, \star\})$, corresponding to $\varnothing, \{\star\}, \{\star, \star\} \in \mathcal{C}(G' \odot^{f,g'} F)$ respectively.

4.4 Games and Strategies

Usually in game semantics, a game consists of a set of available moves and a set of rules that Player and Opponent must abide to. A strategy for Player is described by the set of possible plays (*i.e.*, sequences of moves) if Player chooses to follows this strategy. Obviously, the strategy must abide to the game, meaning that all the plays of the strategy must respect the rules of the game.

In previous sections, we introduced event structures. They are used to represent both games and strategies. Indeed, when representing games, events of the event structures stand for the available moves, and order and consistency for the rules of the game. Hence, configurations represent "legal states of the game". When representing strategies, configurations of the strategy loosely correspond to "plays" of the strategy. The event structure of the strategy and the event structure of the game are related by a map of event structure, which formalises "the strategy abides by the rules of the game". Indeed, maps of event structure send configurations to configurations, so assuming the maps goes from a strategy to a game, it will send "plays of the strategy" to "legal states of the game".

However, with only the definitions of previous sections, there is no difference between Player moves and Opponent moves. In this section, we add polarities to event structures to represent Player and Opponent, and build a category in which strategies only describe the behaviour of Player, *i.e.*, a Player strategy cannot prevent opponent to play a move which is allowed by the game.

4.4.1 Event Structures with Polarities

Events of our event structures will correspond to moves of a game. As noted earlier, we use two-player games, between Player and Opponent, represented by polarities $\oplus$ and $\ominus$ respectively.

Definition 4.4.1. *An event structure with polarities, or esp, is an event structure E together with a polarity function $p_E : |E| \to \{\ominus, \oplus\}$. We say that an esp E is a substructure of an esp E' if they are substructures as event structures and the inclusion preserves polarities. Maps of esps are maps of event structures preserving polarities. We write* **ESP** *for the category.*

We extend $\parallel$ and $\oplus$ inheriting polarities. In fact, **ESP** is a cocartesian SMC. We extend all the notations from event structures, and add the following ones for $p \in \{\ominus, \oplus\}$:

$$\begin{array}{rcccc} e^p & : & e \in |E| & \text{and} & p_E(e) = p \\ x \subseteq^p y & : & x \subseteq y & \text{and} & \forall e \in y \backslash x, p_E(e) = p \\ x \mathrel{-\!\!\subset}^p y & : & x \mathrel{-\!\!\subset} y & \text{and} & \forall e \in y \backslash x, p_E(e) = p \end{array}$$

The notion of hiding also extends to esps.

Definition 4.4.2. *For E an esp, we define the negation $E^\perp$ as the esp with the same underlying event structure as E, but opposite polarities. It extends to an endofunctor on* **ESP**.

Definition 4.4.3. *We consider two maps of esps $f : F \to A^\perp \parallel B$ and $g : G \to B^\perp \parallel C$. We define the esp $G \odot^{f,g} F$, as the event structure $G \odot^{f,g} F$ together with the polarity*

$$p_{G \odot^{f,g} F}(e) = p_{A^\perp \parallel C}((g \odot f)(e))$$

and we remark that the map of event structures $g \odot f$ is a map of esps, from $G \odot^{f,g} F$ to $A^\perp \parallel C$. This extends interactive composition to esps.

We note that we cannot properly express the interaction $G \circledast^{f,g} F$ within esps, as the events at the middle are neither positive nor negative. It is possible to add a third polarity, neutral, as for example in [CHLW14, CCHW18], but we do not need to do so in this thesis.

4.4.2 The Category of Concurrent Games and Strategies

Definition 4.4.4. *A game is an esp A which is*

Alternating *if whenever $a \rightarrowtriangle_A b$, the events a and b have different polarities.*

Race-Free *if whenever $x \mathrel{-\!\!\subset}^+ y$ and $x \mathrel{-\!\!\subset}^- z$, then $y \cup z \in \mathcal{C}(A)$. In the binary conflict case, it is equivalent to: whenever $a \sim_A b$, the events a and b have the same polarity.*

While our games are not strictly "turn-based", as parallelism might allow a player to make multiple moves before its opponent plays any, those two properties allow us to keep intuitions from turn-based games. Those conditions do not appear in the most general definitions of concurrent games and strategies (see [CCRW17, Win11]), but will be required for the quantum game model to work so we introduce them immediately. The race freeness properties can be reformulated to:

$$x \sqsubseteq y \implies x \cup y \in \mathcal{C}(A)$$

where $\sqsubseteq := \ ^{-}\supseteq \circ \subseteq^{+}$. This partial order $\sqsubseteq$ on configurations in called the Scott order (see [Win13]). It is known to appear in a lot of contexts in concurrent game semantics, and is reminiscent of the pointwise order on functions in domain theory: increasing in the order means decreasing the information on inputs (Opponent moves) and increasing the information on outputs (Player moves).

Definition 4.4.5. *A pre-strategy (σ, S) from a game A to a game B is an esp S and a map of esps $\sigma : S \to A^{\perp} \parallel B$. Two pre-strategies σ, σ' from A to B are said isomorphic, and we write $\sigma \cong \sigma'$, if there exists an isomorphism of esps $\iota : S \to S'$ such that $\sigma = \sigma' \circ \iota$.*

When we say that "σ is a pre-strategy", we instead mean "(σ, S) is a pre-strategy", keeping the esp S implicit. We take the convention that the esp of a pre-strategy σ will be named S. Similarly a pre-strategy named $\tau, \sigma', \tau', \sigma_n$ or τ_n will have an esp named respectively T, S', T', S_n or T_n (for $n \in \mathbb{N}$).

Definition 4.4.6. *The copy-cat pre-strategy $\mathfrak{c}_A : \mathfrak{C}_A \to A^{\perp} \parallel A$ is defined as:*

- $|\mathfrak{C}_A| = |A^{\perp} \parallel A|$, $p_{\mathfrak{C}_A} = p_{A^{\perp} \parallel A}$ *and $\mathfrak{c}_A$ is the identity on events.*
- $(i,a) <_{\mathfrak{C}_A} (j,b)$ *when* $a <_A b$ *or* $\begin{cases} a = b \\ p_{\mathfrak{C}_A}(i,a) = \ominus \\ p_{\mathfrak{C}_A}(j,b) = \oplus \end{cases}$
- $X \parallel Y \in \mathrm{Con}_{\mathfrak{C}_A}$ *when* $X \cup Y \in \mathrm{Con}_A$.

We refer to Fig. 4.13 for an example of a copy-cat strategy. The definition of copy cat we gave here is a simplification of the definition given in [CCRW17]. This simplification is only well-behaved because of the alternating and race-free properties on games[3].

It follows from the definition that $x \parallel y \in \mathcal{C}(\mathfrak{C}_A)$ if and only if $x, y \in \mathcal{C}(A)$, $y \backslash x$ contains only negative events and $x \backslash y$ contains only positive events. This exactly corresponds to $x \sqsupseteq y$. We note that $\mathfrak{c}_A \odot \mathfrak{c}_A \cong \mathfrak{c}_A$. We do not always have $\mathfrak{c}_A \odot \sigma \cong \sigma \cong \sigma \odot \mathfrak{c}_A$ for σ a pre-strategy. However, we will prove in Theorem 4.4.9 that it is true for a restricted class of pre-strategies, respecting properties called receptivity and courtesy.

[3] More precisely, in the non-race-free case $\mathfrak{C}_A$ does not always satisfy the axioms of event structures, and in the non-alternating case this definition would not give rise to a courteous pre-strategy. See below for the definition of courtesy.

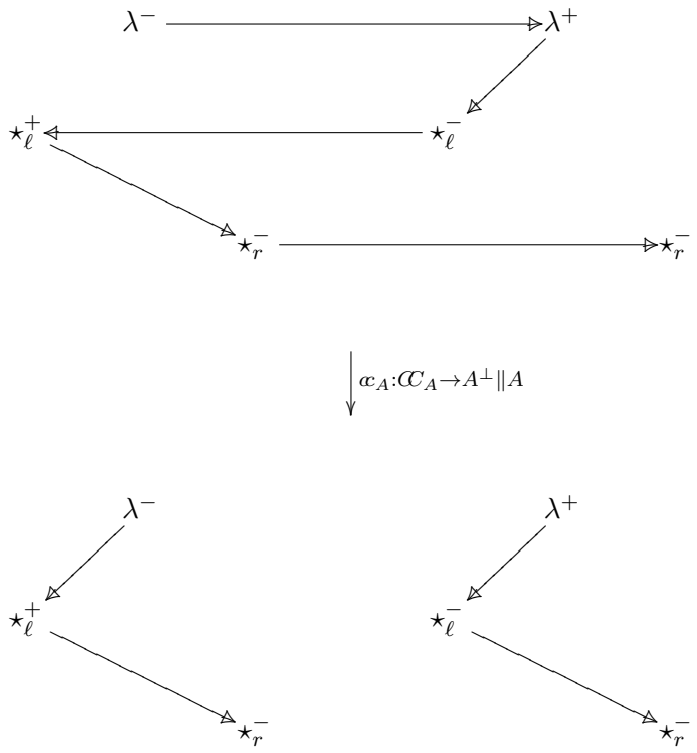


Figure 4.13: Example of the copy-cat pre-strategy ϖ_A with $A = \mathbf{1} \multimap \mathbf{1}$.

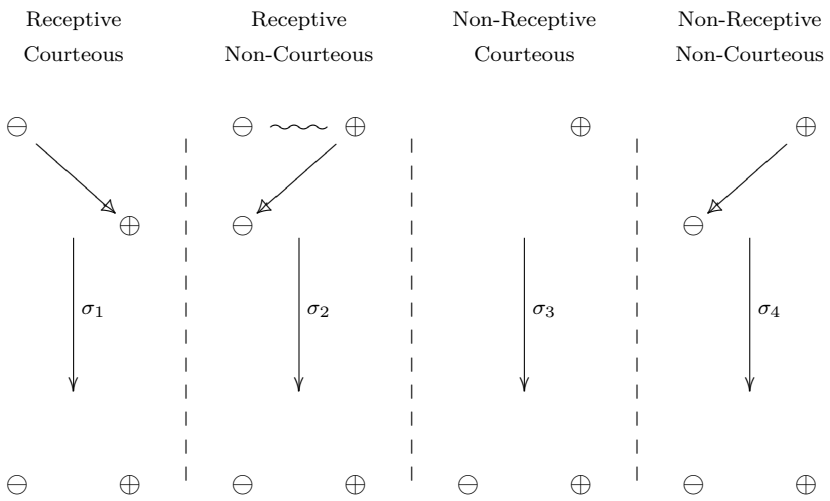


Figure 4.14: Examples of (non)-receptivity and (non)-courtesy.

Definition 4.4.7. *A pre-strategy σ from A to B is said to be*

Receptive *if whenever $x \in \mathcal{C}(S)$ and $\sigma\, x \mathrel{-\!\!\subset^-} (\sigma\, x \sqcup \{e^-\})$, there exists a unique $x \mathrel{-\!\!\subset^-} (x \sqcup \{s^-\})$ such that $\sigma(s) = e$.*

Courteous *if whenever $s \rightarrowtriangle_S t$, if s is positive or t is negative, we have $\sigma(s) \rightarrowtriangle_{A^\perp \| B} \sigma(t)$*

Both conditions aim at the same goal: ensuring that the strategy only describe Player's behaviour, without putting any restriction on Opponent's behaviour that was not already in the game.

In Fig. 4.14, the pre-strategies σ_3 and σ_4 are non-receptive since we cannot play any negative move in the pre-strategy from the empty configuration, while we can play a negative move from the empty configuration in the game. The pre-strategies σ_2 and σ_4 are non-courteous since we have an immediate causality $\oplus \rightarrowtriangle \ominus$ which does not come from the game, and courtesy only allows immediate causalities $\ominus \rightarrowtriangle \oplus$ as in σ_1.

Proposition 4.4.8. *For every game A, æ_A is receptive and courteous. Additionally, the interactive composition of two receptive (resp. courteous) pre-strategies is receptive (resp. courteous).*

We refer to [CCRW17] for a proof. When considering the interactive composition $\tau \odot \sigma$, we will use $T \odot S$ as a shorthand for $T \odot^{\sigma,\tau} S$. We similarly write $T \circledast S$ for $T \circledast^{\sigma,\tau} S$.

Theorem 4.4.9. *For every pre-strategy σ from A to B, σ is receptive and courteous if and only if $\text{æ}_B \odot \sigma \cong \sigma \cong \sigma \odot \text{æ}_A$.*

This theorem is well-known and its proof detailed in [CCRW17].

Definition 4.4.10. *A strategy σ from a game A to a game B is a receptive and courteous pre-strategy. We write it $\sigma : A \nrightarrow B$*

Lemma 4.4.11. *If $\sigma : S \to A^\perp \parallel B$ is a strategy, then S is alternating and race-free.*

Proof. Indeed, assume $s \rightarrowtriangle_S s'$, then by courtesy we have s negative and s' positive, or $\sigma(s) \rightarrowtriangle_{A^\perp \parallel B} \sigma(s')$, and since A and B are alternating, this means that s and s' have opposite polarities. So S is alternating.

Similarly, assume $x \mathrel{-\!\!\subset^+} x \cup \{s^+\}$ and $x \mathrel{-\!\!\subset^-} x \cup \{t^-\}$, with $x \in \mathcal{C}(S)$. Using race-freeness of the game, $\sigma(x) \cup \{\sigma(s), \sigma(t)\}$ is a configuration. Using receptivity, there exists $x \cup \{s^+\} \mathrel{-\!\!\subset^-} x \cup \{s^+, t'^-\}$ such that $\sigma(t') = \sigma(t)$. Using courtesy, since we do not have $\sigma(s) \rightarrowtriangle \sigma(t')$, it means we do not have $s \rightarrowtriangle_S t'$, so $x \mathrel{-\!\!\subset^-} x \cup \{t'^-\}$. Using the uniqueness part of receptivity, we obtain $t = t'$. It follows that $x \cup \{s^+, t^-\} \in \mathcal{C}(S)$. So S is race-free. □

Games and strategies are not per se a category, as the associativity and identity laws only hold up to isomorphism. In [CCRW17], they are proven to form a bicategory. However, the bicategorical structure is not a focus of this thesis, and for syntactical simplicity we choose not to work at the bicategorical level. The natural alternative would be to quotient by isomorphism, and to consider the category of games and equivalence classes of strategies. This approach works for every use of the category but one: the recursion. When trying to model recursion in Section 9.3.3, we will temporarily need to work with concrete strategies. This is because the substructure order defined in Definition 4.4.1 is not preserved by isomorphism, and does not induce an order on the quotient category[4]. This is why we choose to use an ad hoc in-between: when we write "Game and strategies form a category up to isomorphism", we mean that games and strategies have the same *data* as a category, but the *axioms* of a category are only satisfied up to isomorphism. We hope the reader will excuse this slight abuse of terminology.

Proposition 4.4.12. *Games and strategies form a CpCC* $(\mathbf{Strat}, \parallel, \varnothing, (_)^\perp)$ *up to isomorphism.*

We refer to [CCRW17] for a proof. We can define a monoidal product $\parallel$ by simply taking the parallel composition of esps. For $\sigma : S \to A^\perp \parallel B$, and $\tau : T \to C^\perp \parallel D$, we can see them as maps of esps and consider $\sigma \parallel \tau : S \parallel T \to A^\perp \parallel B \parallel C^\perp \parallel D$. Using the associator and braiding of the SMC of esps and maps of esps, we can obtain a map of esps from $S \parallel T$ to $A^\perp \parallel C^\perp \parallel B \parallel D$, so a pre-strategy from $A \parallel C$ to $B \parallel D$. We easily check that receptivity and courtesy are preserved, so it is a strategy. For the unit and counit, we simply take the copy-cat map $\mathrel{Œ}C_A \to A^\perp \parallel A$, and see it either as a strategy from $\varnothing$ to $A^\perp \parallel A$ or from $A \parallel A^\perp$ to $\varnothing$. We note the ambiguity between $\sigma \parallel \tau$ seen as strategies and $\sigma \parallel \tau$ seen as maps of esps, and hope it does not confuse the reader in later proofs.

[4] The antisymmetry fails.

Chapter 5

Quantum Concurrent Games

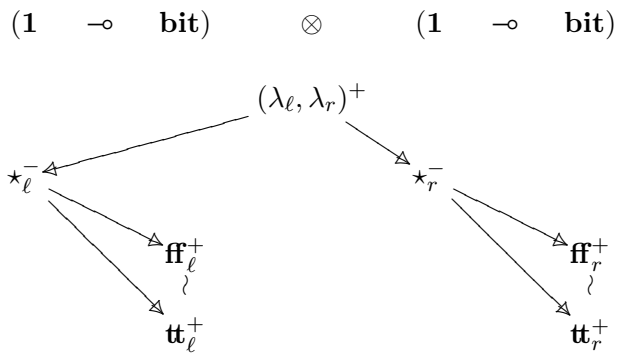
5.1 Guiding Example

This chapter focuses on defining the game model. The interpretation of QΛ in this model will be the the focus of the next chapter. While we will not properly give a semantics to terms of QΛ in this chapter, we will still have a guiding example coming from QΛ. We refer to Section 6.1 for an explanation on how the games and strategies of the examples are computed systematically from the types and terms.

We will consider three terms t_N, t_P, t_Q. The term t_Q is from AQΛ, the term t_P is from a probabilistic variant of AQΛ, and t_N is from a non-deterministic variant of AQΛ in which $+$ stands for a non-deterministic choice. We start by studying a non-deterministic term as probabilistic and quantum semantics can be seen as non-deterministic semantics weighted respectively by probabilities or quantum operators.

$$
\begin{array}{llll}
f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} & \vdash_{\mathbb{A}} & t_N : & (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\
 & & t_N := & (\lambda().f_0\ () + f_1\ ()) \otimes \lambda().\mathbf{tt} \\
f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} & \vdash_{\mathbb{A}} & t_P : & (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\
 & & t_P := & \left(\lambda().\tfrac{1}{3} f_0\ () + \tfrac{2}{3} f_1\ ()\right) \otimes \lambda().\mathbf{tt} \\
f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} & \vdash_{\mathbb{A}} & t_Q : & (\mathbf{qubit} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\
 & & t_Q := & (\lambda x.\mathbf{if}\ \mathbf{meas}\ x\ \mathbf{then}\ f_0\ ()\ \mathbf{else}\ f_1\ ()) \otimes \lambda().\mathbf{tt}
\end{array}
$$

The three terms call either f_0 or f_1 from the context depending on the result of a choice, a non-deterministic one for t_N, and probabilistic one for t_P, and a quantum one for t_Q. In this section, we will explain the semantics of t_N. We keep t_P and t_Q for the next two sections. First, the games. The return type of our term is $(\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$, which we represent in the following diagram:



This is an event structure with polarities, as defined in Section 4.2. Every event of this event structure can be seen as corresponding to a component of the type $(\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$. Accordingly, we write the type associated to the game at the top of the event, and organise the events so that each event appears below the corresponding part of the type. Reading from top to bottom, we have

- A positive event $(\lambda_\ell, \lambda_r)^+$, representing Player (*i.e.*, the program) saying "two functions are defined and ready to be called".

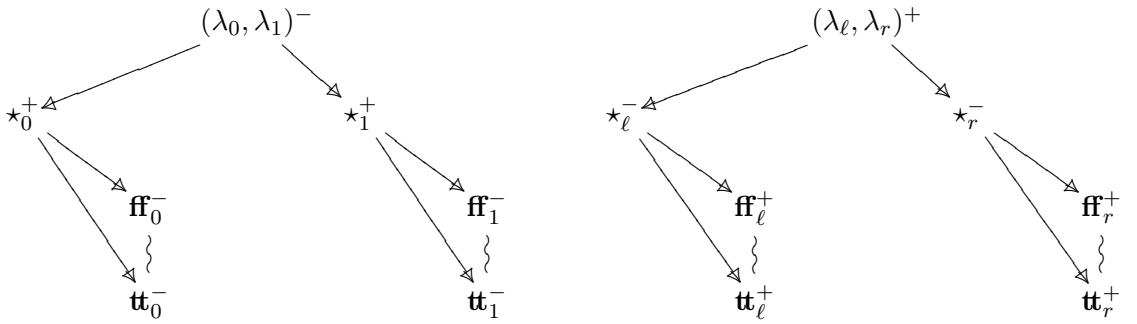
- Two negative events $\star_\ell$ and $\star_r$, representing Opponent (*i.e.*, the user) calling respectively the first function on input () and the second function on input (). Those two events causally depend on $(\lambda_\ell, \lambda_r)^+$ as the user cannot call a function that does not exist yet. Those two events have no conflict between them as it is possible for the user to call both.

- Two positive events at the left hand side $\mathbf{ff}_\ell^+$ and $\mathbf{tt}_\ell^+$ representing the two possible outputs that Player can give when the left hand side function is called. They causally depend on $\star_\ell^-$ since a function cannot output before being called. They are in conflict with each other as Player can only give one output when the function is called.

- Two positive events at the right hand side $\mathbf{ff}_r^+$ and $\mathbf{tt}_r^+$ representing the two possible outputs that Player can give when the right hand side function is called.

To represent the full typing context $f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$, we use the following game. Note that the polarities of the left hand side are reversed compared to the right hand side, as the functions in the context are given by Opponent and potentially called by Player.

$$\mathbf{1} \multimap \mathbf{bit} \quad , \quad \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$



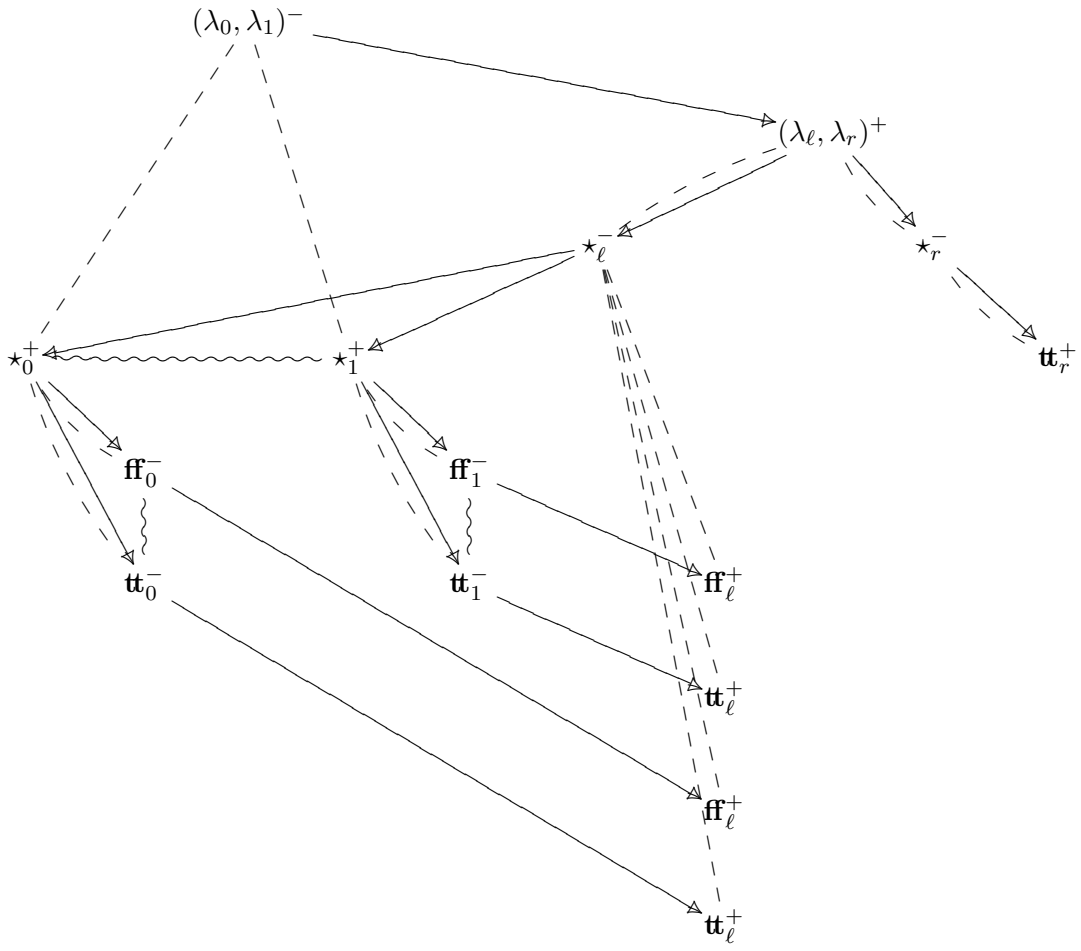
The term t_N is represented by a strategy, which we describe through the event structure represented in Fig. 5.1 (ignoring dashed lines), together with a correspondence between its events and the event of the game. In diagrams, this correspondence between events of the strategy and events of the game is implicitly given by (1) our naming convention of the events, (2) the presence of the typing context at the top of the diagram, with every event below the component of the type it corresponds to, and (3) the causal links of the game being reminded through dashed lines. From top to bottom, the strategy reads as follows:

- Opponent starts the computation with the event $(\lambda_0, \lambda_1)^-$ signalling that the functions f_0 and f_1 are ready to be used by Player.
- Player then answers with the event $(\lambda_\ell, \lambda_r)^+$ announcing he successfully managed to define two functions, and that they are ready to be called.
- Opponent can call any of the two functions with the events $\star_\ell^-$ and $\star_r^-$, or both.
- If Opponent calls the second function, then Player answer "true" through the event $\mathbf{tt}_r^+$.
- If Opponent calls the first function, then Player non-deterministically chooses between calling f_0 with $\star_0^+$ or f_1 with $\star_1^+$, but does not call both.

→ If Player calls f_0, then Opponent answers a boolean through either $\mathbf{ff}_0^-$ or $\mathbf{tt}_0^-$, which Player forwards as the answer to Opponent's call to the first function, represented by either $\mathbf{ff}_\ell^+$ or $\mathbf{tt}_\ell^+$.

→ Similarly, if Player calls f_1 instead, then Opponent answers a boolean through either $\mathbf{ff}_1^-$ or $\mathbf{tt}_1^-$, which Player forwards as the answer to Opponent's call to the first function, represented by either the second copy of $\mathbf{ff}_\ell^+$ or the second copy of $\mathbf{tt}_\ell^+$.

$$\mathbf{1} \multimap \mathbf{bit} \quad , \quad \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$



Term represented:

$$f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} \quad \vdash_{\mathbb{A}} \quad \begin{array}{ll} t_N : & (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\ t_N := & (\lambda().f_0\ () + f_1\ ()) \otimes \lambda().\mathbf{tt} \end{array}$$

Figure 5.1: Example of non-deterministic strategy

5.2 Probabilistic Concurrent Strategies

Quantum strategies arise as a generalisation of probabilistic strategies as developed in [Win14]. We give a quick overview of probabilistic strategies here.

5.2.1 The Guiding Example

We now consider the term t_P which we recall here:

$$\begin{array}{lllll} f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} & \vdash_{\mathbb{A}} & t_P : & (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\ & & t_P := & \Big(\lambda().\tfrac{1}{3} f_0\ () + \tfrac{2}{3} f_1\ ()\Big) \otimes \lambda().\mathbf{tt} \end{array}$$

Its typing context is the same as t_N, hence the game representing the type will be the same. In fact, t_N and t_P behave almost exactly the same, the only difference being that the choice between calling f_0 and calling f_1 now has probabilities associated: a third for f_0 and two thirds for f_1. We describe in Fig. 5.2 the game semantics of t_P, and one can remark that the event structure is exactly identical to the one of t_N in Fig. 5.1. The only difference between their game semantics is the addition of the valuation v which associates to every configuration of the strategy a probability in $[0, 1]$, which has to be understood as follows:

On the configuration $\{(\lambda_0, \lambda_1)^-, (\lambda_\ell, \lambda_r)^+, \star_\ell^-, \star_0^+\}$, the valuation is $1/3$, meaning that if we ignore the unknown likelihood of Opponent eventually playing the moves $(\lambda_0, \lambda_1)^-$ and $\star_\ell^-$, the probability of eventually reaching a configuration greater or equal to $\{(\lambda_0, \lambda_1)^-, (\lambda_\ell, \lambda_r)^+, \star_\ell^-, \star_0^+\}$ is $1/3$.

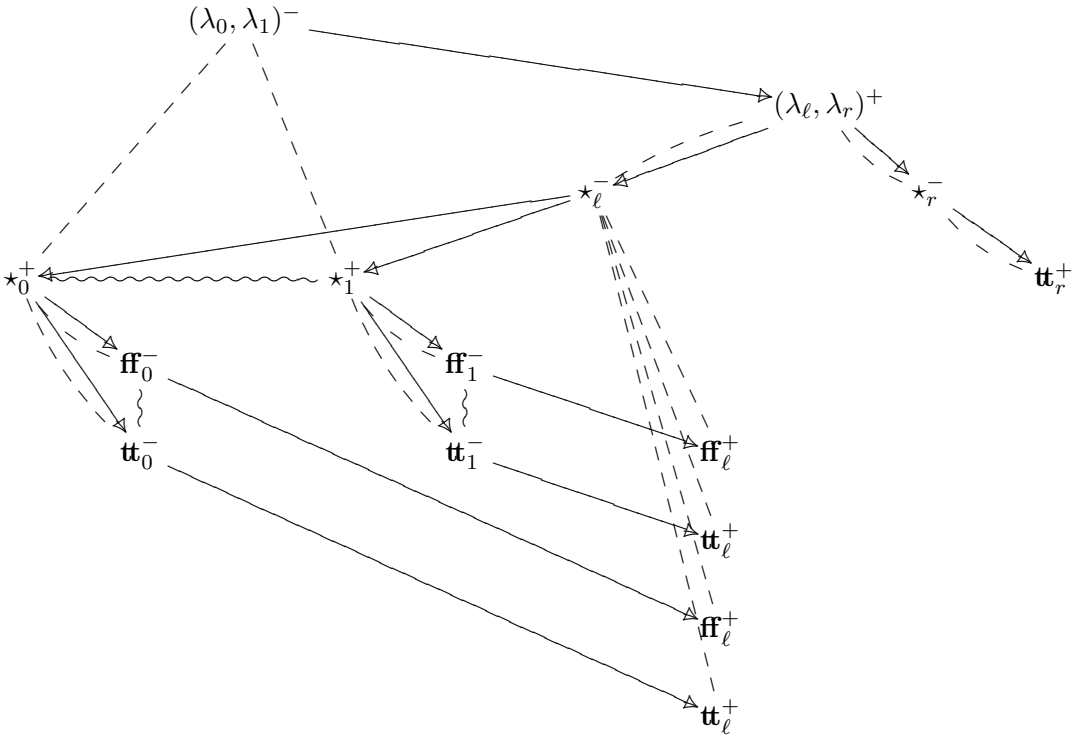
This probabilistic valuation will be expected to satisfy a collection of axioms ensuring that the strategy indeed corresponds to a probabilistic system, rather than having some arbitrary weight in $[0, 1]$ given to every configuration without any coherence requirement.

5.2.2 Defining Probabilistic Strategies

To give an intuition about the restrictions probabilistic strategies should satisfy, we start with a simpler case: we consider a finite event structure E, and want to add probabilities to it.

A first approach is to consider a probability measure μ_E over $\mathcal{C}(E)$. In this approach, if E describes a system and all its possible executions, then $\mu_E(\{x\})$ is to be interpreted as the probability the configuration x being the final state of an execution. More generally, $\mu_E(S)$ is the probability of the final state of an execution to be any $x \in S$. Since E is finite, a remarkable property is that μ_E is entirely characterised by the $v_E(x) = \mu_E(\{y \mid x \subseteq y\})$ for $x \in \mathcal{C}(E)$. We call $v_E(x)$ the probabilistic valuation of x, it corresponds to the probability of reaching at least the configuration x. This notion of valuation obtained from a probability distribution on configurations of an event structure is extended to the infinite case and studied in more detail in [Win14].

$$\mathbf{1} \multimap \mathbf{bit} \quad , \quad \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$



Probabilistic valuation:

$$v(x) = \begin{cases} 1/3 & \text{if } \star_0^+ \in x \\ 2/3 & \text{if } \star_1^+ \in x \\ 1 & \text{otherwise} \end{cases}$$

Term represented:

$$f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} t_P : (\mathbf{1} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$
$$t_P := \left(\lambda().\tfrac{1}{3} f_0\ () + \tfrac{2}{3} f_1\ ()\right) \otimes \lambda().\mathbf{tt}$$

Figure 5.2: Example of probabilistic strategy

E	μ_E	v_E
$a \sim\sim b$	$\{\varnothing\} \mapsto 0$ $\{\{a\}\} \mapsto 1/3$ $\{\{b\}\} \mapsto 2/3$	$\varnothing \mapsto 1$ $\{a\} \mapsto 1/3$ $\{b\} \mapsto 2/3$

Configurations: $\mathcal{C}(E) = \{\varnothing, \{a\}, \{b\}\}$
Property satisfied: $v_E(\varnothing) - v_E(\{a\}) - v_E(\{b\}) \geq 0$

E	μ_E	v_E
$a \quad b$	$\{\varnothing\} \mapsto 0$ $\{\{a\}\} \mapsto 1/4$ $\{\{b\}\} \mapsto 1/2$ $\{\{a,b\}\} \mapsto 1/4$	$\varnothing \mapsto 1$ $\{a\} \mapsto 1/2$ $\{b\} \mapsto 3/4$ $\{a,b\} \mapsto 1/4$

Configurations: $\mathcal{C}(E) = \{\varnothing, \{a\}, \{b\}, \{a,b\}\}$
Property satisfied: $v_E(\varnothing) - v_E(\{a\}) - v_E(\{b\}) + v_E(\{a,b\}) \geq 0$

Figure 5.3: Examples of event structure with probabilities

A second approach is, instead of considering a probability measure and then computing the valuation of every configuration, to directly axiomatise this probabilistic valuation. Such a valuation should satisfy the normalisation property $v_E(\varnothing) = 1$, as the probability of reaching at least the empty configuration is 1, and an inclusion-exclusion principle illustrated in Fig. 5.3 and formalised as:

$$x \subseteq y_1, \ldots, y_n \implies v_E(x) - \sum_{\substack{\varnothing \neq I \subseteq \{1,\ldots,n\} \\ \bigcup_{i\in I} y_i \in \mathcal{C}(E)}} (-1)^{|I|-1} v_E\left(\bigcup_{i\in I} y_i\right) \geq 0$$

When adding polarities for strategies, Opponent moves must be treated differently. Indeed, the strategy should only describe the behaviour of Player, and should not put any constraint on Opponent. To express this, it is easier to rely on the formalism of probabilistic valuations than on the one of probability measures, hence the following definition.

Definition 5.2.1. *A probabilistic strategy σ from a game A to a game B is a strategy $\sigma : A \nrightarrow B$ together with a probabilistic valuation $v_\sigma : \mathcal{C}(S) \to [0,1]$ satisfying:*

Normalisation $v_\sigma(\varnothing) = 1$

Obliviousness $x \subseteq^- y \implies v_\sigma(y) = v_\sigma(x)$

Drop condition $x \subseteq^+ y_1, \dots, y_n \implies d(x; y_1, \dots, y_n) \geq 0$, *where*

$$y_I := \bigcup_{i \in I} y_i \qquad d(x; y_1, \dots, y_n) := v_\sigma(x) - \sum_{\substack{\varnothing \neq I \subseteq \{1,\dots,n\} \\ y_I \in \mathcal{C}(S)}} (-1)^{|I|-1} v_\sigma(y_I)$$

We recall that S denotes the esp such that $\sigma : S \to A^\perp \parr B$. This definition shares a lot of similarity with [DH02], which defines probabilistic strategies in the context of sequential game semantics. In fact, if we restrict ourselves to only sequential strategies[1], we recover the same definition of probabilistic strategies.

Definition 5.2.2. *The probabilistic valuation of copy-cat is constant equal to* 1*. The interactive composition $\tau \odot \sigma$ is given by the interactive composition of strategies and*

$$v_{\tau \odot \sigma}(y \odot x) := v_\tau(y) \times v_\sigma(x)$$

Similarly,

$$v_{\sigma \parr \tau}(x \parr y) := v_\sigma(x) \times v_\tau(y)$$

We recall that all the configurations of $T \odot S$ can be written as $y \odot x$, as per Proposition 4.3.6. We note that proving that the interactive composition of two probabilistic strategies satisfies the drop composition is non-trivial, and uses the race-freeness of games. See [Win11] for a proof, or Proposition 5.3.11 for a proof in the more general case of quantum strategies.

Proposition 5.2.3. *Games and probabilistic strategies, up to isomorphism, form a CpCC* $(\mathbf{PStrat}, \parr, \varnothing, (_)^\perp)$.

5.3 Quantum Concurrent Games

5.3.1 Definition of Quantum Games

We now want to generalise probabilistic strategies to quantum strategies. We note that [Win14] defines both the notion of probabilistic strategies detailed in Section 5.2, and a notion of quantum strategies. However, while our notion of quantum strategies is an extension of his notion of probabilistic strategies, we choose a completely different representation of quantum operations, which we believe to be more suited for game semantics of the quantum λ-calculus. We will replace the probabilistic valuation $v_\sigma : \mathcal{C}(S) \to [0, 1]$ by a quantum valuation $\mathcal{Q}_\sigma : \mathcal{C}(S) \to \mathbf{CPM}(_, _)$. However, we need to determine what Hilbert spaces to put instead of $_$. Our goal is quantum game semantics, where the game

[1] A strategy σ is sequential if S has only configurations of the form $s_0 \rightarrowtriangle_S s_1 \rightarrowtriangle_S \dots \rightarrowtriangle_S s_n$ with s_0 minimal. In other words, σ is sequential if S is tree-like and its branches are in conflict with each other.

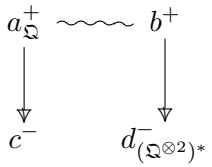


Figure 5.4: A Quantum Game

will represent the type of a term, and the strategy will represent the term itself. In QΛ, it is the type of a term that specifies the number of qubits it takes as inputs or outputs. As such, it should be the game of a strategy that specifies the Hilbert spaces used as domain and codomain for the quantum valuation.

Definition 5.3.1. *A quantum game is a game A together with a space annotation $\mathcal{H}_A$ which associates to every event a (finite dimensional) Hilbert space. We write for $x \in \mathcal{C}(A)$:*

$$\mathcal{H}_A(x) := \bigotimes_{e \in x} \mathcal{H}(e)$$

We then take $\mathcal{H}_{A^\perp}(e) := \mathcal{H}_A(e)^*$, $\mathcal{H}_{A\|B}(0,a) = \mathcal{H}_{A\oplus B}(0,a) = \mathcal{H}_A(a)$ and $\mathcal{H}_{A\|B}(1,b) = \mathcal{H}_{A\oplus B}(1,b) = \mathcal{H}_B(b)$.

We note that for the definition of $\mathcal{H}_A(x)$ to be rigorous, we need to specify an order in which we go through the $e \in x$. However, we choose to leave them implicit to aid readability. We leave the order in which we tensor the Hilbert spaces implicit, and leave the associators, braiding and unitor isomorphisms implicit too. The coherence theorem of SMCs ensures that the different available ways to insert such isomorphisms lead to the same result. Additionally, up to the natural isomorphism between H and H^{**}, we have $(A^\perp)^\perp = A$. As such, one can consider $(_)^\perp$ to be an involution.

In diagrams, we write e^p_H for an event e of polarity p and Hilbert space H. Events which appears in diagrams without space annotations are implicitly annotated by $\mathbf{1} = \mathbb{C}$ if they are positive, and $\mathbf{1}^*$ if they are negative[2]. For example, in Fig. 5.4, we have

$$\mathcal{H}_A(a) = \mathfrak{Q} = \mathbb{C}^2 \quad \mathcal{H}_A(b) = \mathbf{1} = \mathbb{C} \quad \mathcal{H}_A(c) = \mathbf{1}^* = \mathbb{C}^* \quad \mathcal{H}_A(d) = (\mathfrak{Q}^{\otimes 2})^* = (\mathbb{C}^4)^*$$

Those Hilbert spaces on events describe the amount of data received by the environment, in the case of negative events, or sent to the environment, in the case of positive events. Those exchanges of information interact naturally with the Scott order (see Section 4.4.2), which can be understood as an information order: if $x \sqsupseteq y$ for the Scott order,

[2]Since $\mathbf{1}$ and $\mathbf{1}^*$ are unitarily isomorphic, we will often identify the two. In particular, we will sometimes consider that $\mathrm{lu}_A \in \mathbf{CPM}(\mathbf{1}^* \otimes A, A)$ and $\mathrm{ru}_A \in \mathbf{CPM}(A \otimes \mathbf{1}^*, A)$.

i.e., there exists z such that $x \,{}^{+}\!\supseteq z \subseteq^{-} y$, then y is doing "less work" (less positive events) with "more resources" (more negative events). In the case of data management, it is always possible to do less work with more resources: discard the additional work you did, and discard the additional resources given to you. Formally, this means that whenever $x \sqsupseteq y$, we should expect a canonical morphism from $\mathcal{H}(x)$ to $\mathcal{H}(y)$.

Definition 5.3.2. *We write* $\mathrm{Scott}(A)$ *for the category with objects configurations of* A*, and maps given by the Scott (partial) order* $\sqsubseteq$*. We can lift* $\mathcal{H}_A$ *into a contravariant functor from* $\mathrm{Scott}(A)$ *to* **CPM** *as follows:*

$$\mathcal{H}_A(x) := \bigotimes_{e \in x} \mathcal{H}(e)$$

$$\mathcal{H}_A(x \subseteq^{+} y) := \mathbf{id}^{\mathbf{CPM}}_{\mathcal{H}_A(x)} \otimes \mathbf{Tr}_{\mathcal{H}_A(y \setminus x)} \in \mathbf{CPM}(\mathcal{H}_A(y), \mathcal{H}_A(x))$$

$$\mathcal{H}_A(x \,{}^{-}\!\supseteq y) := \mathbf{id}^{\mathbf{CPM}}_{\mathcal{H}_A(y)} \otimes \mathbf{Tr}^{\dagger}_{\mathcal{H}_B(x \setminus y)} \in \mathbf{CPM}(\mathcal{H}_A(y), \mathcal{H}_A(x))$$

We recall that the morphisms $\mathbf{Tr}^{\dagger}_H \in \mathbf{CPM}(\mathbf{1}, H)$ and $\mathbf{Tr}_H \in \mathbf{CPM}(H, \mathbf{1})$ are defined as follows:

$$\mathbf{Tr}_H : \quad M \quad \mapsto \quad \sum_{i=1}^{\dim H} m_{i,i} \qquad \mathbf{Tr}^{\dagger}_H : \quad z \quad \mapsto \quad z \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

The main use of this functor is in upcoming Definition 5.3.3 to have a canonical way to coerce two functions, one in $\mathbf{CPM}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$ and one in $\mathbf{CPM}(\mathcal{H}_A(y_A), \mathcal{H}_B(y_B))$, onto a common space $\mathbf{CPM}(H, K)$ so that they can be compared to each other.

5.3.2 Back to the Guiding Example

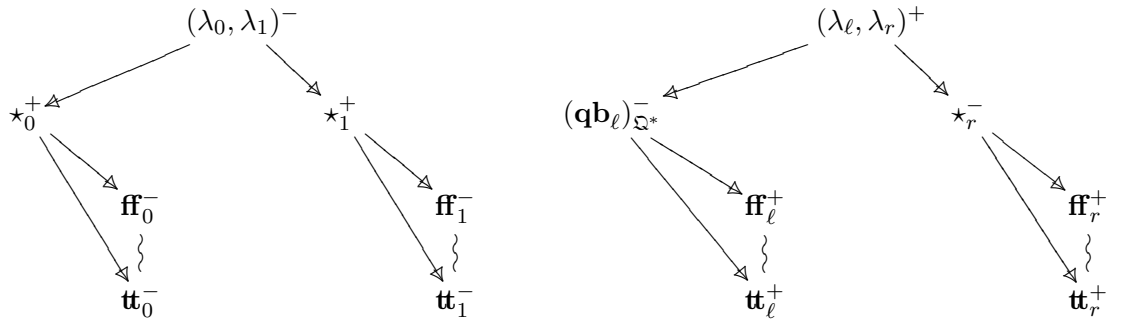
We now consider the term t_Q which we recall here:

$$\begin{array}{lllll} f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} & \vdash_{\mathbb{A}} & t_Q : & (\mathbf{qubit} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\ & & t_Q := & (\lambda x.\mathbf{if}\ \mathbf{meas}\ x\ \mathbf{then}\ f_0\ ()\ \mathbf{else}\ f_1\ ()) \otimes \lambda().\mathbf{tt} \end{array}$$

As its type is slightly different from the one of t_N and t_P, the game differs too and is represented below: the negative event $\star^{-}_{\ell}$ is now replaced by $(\mathbf{qb}_{\ell})^{-}_{\mathbb{Q}^*}$ which comes annotated

by the space $\mathfrak{Q}^*$, as Opponent has to provide one qubit in order to call the function.

$$\mathbf{1} \multimap \mathbf{bit} \quad , \quad \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{qubit} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$



We describe the game semantics of t_Q in Fig. 5.5. Note that the difference with the semantics of t_P is small: they have exactly the same event structure apart from the renaming of $\star_\ell^-$ into $(\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}$, and the probabilistic valuation v is replaced by a quantum valuation $\mathcal{Q}$ which provides a **CPM** operator for every configuration instead of a probability:

- For any configuration x which does not contain $(\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}$, all the quantum spaces are trivial so $\mathcal{Q}(x) = p\mathbf{id_1}$ with p being the probability that would be associated to this configuration by a probabilistic semantics, in our case 1.

- For any configuration x containing $(\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}$ and $\star_0^+$, we are in an execution where the measurement returned true. The **CPM** operator corresponding to "measuring true" is $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto d$ which is in $\mathbf{CPM}(\mathfrak{Q}, \mathbf{1})$. However, we are not measuring a qubit of the context, we are measuring a qubit obtained from a λ-abstraction. Each layer of λ-abstraction involves a use of the currying adjunction of the compact closure of **CPM**. More formally, we will have $\mathcal{Q}(x) \in \mathbf{CPM}(H_{\text{context}}, H_{\text{term}})$ with H_{context} being the tensor of the Hilbert spaces of the events of x that are at the left of the $\vdash_{\mathbb{A}}$, and H_{term} being the tensor of the Hilbert spaces of the events of x that are at the right of $\vdash_{\mathbb{A}}$. So here $\mathcal{Q}(x) \in \mathbf{CPM}(\mathbf{1}, \mathfrak{Q}^*)$. Applying the compact closure on $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto d$ to obtain a morphism of $\mathbf{CPM}(\mathbf{1}, \mathfrak{Q}^*)$ gives $z \mapsto \begin{pmatrix} 0 & 0 \\ 0 & z \end{pmatrix}$.

- For any configuration x containing $(\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}$ and $\star_1^+$, we are in the execution where the measurement returned false. The **CPM** operator corresponding to "measuring false" is $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a$ which is in $\mathbf{CPM}(\mathfrak{Q}, \mathbf{1})$. Using the compact closure as previously, we obtain $z \mapsto \begin{pmatrix} z & 0 \\ 0 & 0 \end{pmatrix}$.

- For any configuration x containing $(\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}$ but not yet $\star_0^+$ or $\star_1^+$, we are in an execution where the user gave a qubit as an input but we did not use it yet. Within our model, an unused qubit is represented in the same way as a qubit measured with the result of the measurement ignored. Before compact closure, the valuation would be $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a+d$, so after compact closure the quantum valuation is $z \mapsto \begin{pmatrix} z & 0 \\ 0 & z \end{pmatrix}$.

5.3.3 Definition of Quantum Strategies

When trying to generalise the obliviousness and drop conditions of probabilistic strategies to quantum strategies, the main issue is that we have to sum and compare **CPM** maps that do not have the same domain and codomain. The room for manoeuvre is quite limited, as we must ensure that the condition remains preserved under interactive composition of strategies. The contravariant functor $\mathcal{H}$ is a more elegant formulation of the solution present in our paper [CdV20]. When generalising the conditions on probabilistic valuations v_σ to conditions on quantum valuations $\mathcal{Q}_\sigma$, instead of comparing and summing real numbers, we now need to compare and sum morphisms of **CPM**; those morphisms might not have the same domain and codomain, so we use the functor $\mathcal{H}$ as a canonical way to coerce them into having the same domain and codomain.

Definition 5.3.3. *A quantum strategy from a quantum game A to a quantum game B is a strategy $\sigma : A \nrightarrow B$ and a quantum valuation $\mathcal{Q}_\sigma$ on configurations $x \in \mathcal{C}(S)$ such that:*

$$\sigma\, x = x^A \parallel x^B \implies \mathcal{Q}_\sigma(x) \in \mathbf{CPM}(\mathcal{H}_A(x^A), \mathcal{H}_B(x^B))$$

Normalisation $\mathcal{Q}_\sigma(\varnothing) = \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$

Obliviousness $x \subseteq^- x' \implies \mathcal{Q}_\sigma(x') = \mathcal{H}_B(x'^B \ {}^-\!\supseteq x^B) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(x^A \subseteq^+ x'^A)$

Drop condition $x \subseteq^+ x_1, \ldots, x_n \implies d_\sigma(x; x_1, \ldots, x_n)$ *is defined in* **CPM**, *where*

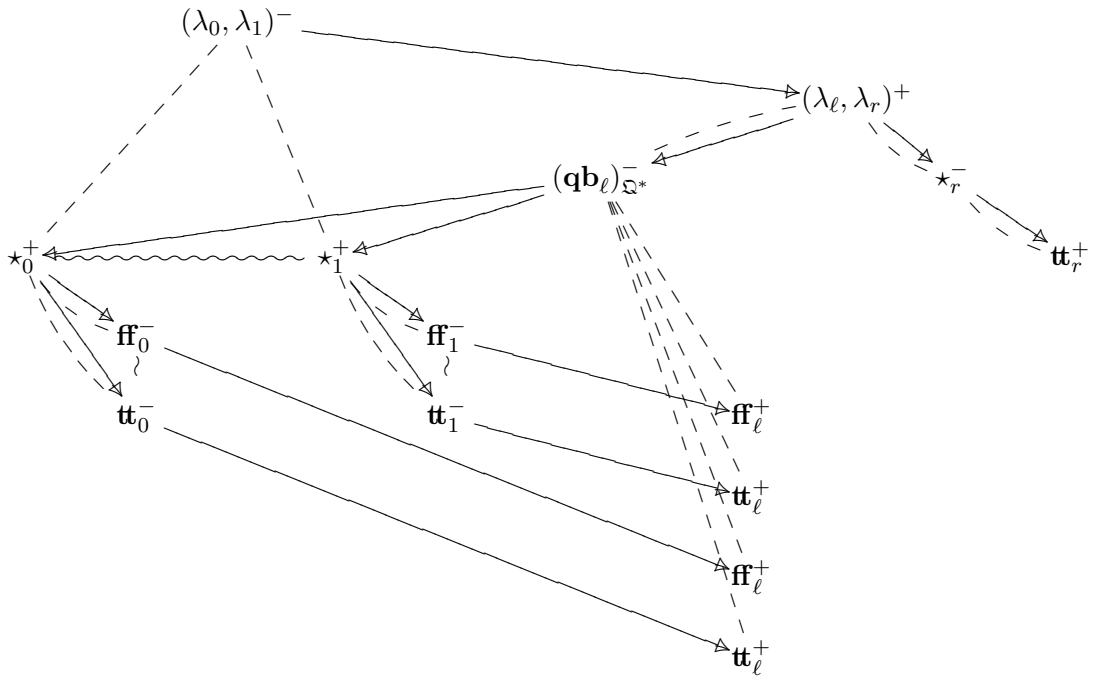
$$d_\sigma(x; x_1, \ldots, x_n) := \mathcal{Q}_\sigma(x) - \sum_{\substack{\varnothing \neq I \subseteq \{1,\ldots,n\} \\ x_I \in \mathcal{C}(S)}} (-1)^{|I|-1} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)$$

and $x_I := \bigcup_{i \in I} x_i$.

We recall that S denotes the esp such that $\sigma : S \to A^\perp \parallel B$. We find it important to note the domain of the quantum valuation is $\mathcal{H}_A(x^A)$ and not $\mathcal{H}_{A^\perp}(x^A)$, which will allow for a smooth composition of quantum valuations. Going back to Fig. 5.5, if we look at the drop condition applied on $x = \{(\lambda_0, \lambda_1)^-, (\lambda_\ell, \lambda_r)^+, (\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}\}$, $x_1 = x \cup \{\star_0^+\}$ and $x_2 = x \cup \{\star_1^+\}$, we obtain

$$d(x; x_1, x_2) = \mathcal{Q}(x) - \mathcal{Q}(x_1) - \mathcal{Q}(x_2) \in \mathbf{CPM}(\mathfrak{Q}, \mathbf{1})$$

$$\mathbf{1} \multimap \mathbf{bit} \quad , \quad \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} (\mathbf{qubit} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit})$$



Quantum valuation:

$$\mathcal{Q}(x) = \begin{cases} \mathbf{id_1} & \text{if } (\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*} \notin x \\ z \mapsto \begin{pmatrix} z & 0 \\ 0 & 0 \end{pmatrix} & \text{if } (\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}, \star_0^+ \in x \\ z \mapsto \begin{pmatrix} 0 & 0 \\ 0 & z \end{pmatrix} & \text{if } (\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*}, \star_1^+ \in x \\ z \mapsto \begin{pmatrix} z & 0 \\ 0 & z \end{pmatrix} & \text{if } (\mathbf{qb}_\ell)^-_{\mathfrak{Q}^*} \in x \not\ni \star_0^+, \star_1^+ \end{cases}$$

Term represented:

$$f_0 : \mathbf{1} \multimap \mathbf{bit}, f_1 : \mathbf{1} \multimap \mathbf{bit} \vdash_{\mathbb{A}} \begin{array}{ll} t_Q : & (\mathbf{qubit} \multimap \mathbf{bit}) \otimes (\mathbf{1} \multimap \mathbf{bit}) \\ t_Q := & (\lambda x.\mathbf{if\ meas}\ x\ \mathbf{then}\ f_0\ ()\ \mathbf{else}\ f_1\ ()) \otimes \lambda().\mathbf{tt} \end{array}$$

Figure 5.5: Example of quantum strategy

which is correct as $\mathcal{Q}(x) - \mathcal{Q}(x_1) - \mathcal{Q}(x_2) = 0 \in \mathbf{CPM}(\mathfrak{Q}, \mathbf{1})$. We now consider another example. We represent in Fig. 5.6 the term

$$x_0 : \mathbf{qubit}, x_1 : \mathbf{qubit} \vdash_{\mathbb{A}} s : (\mathbf{1} \multimap (\mathbf{1} \oplus \mathbf{qubit})) \otimes (\mathbf{1} \multimap \mathbf{qubit})$$
$$s := (\lambda().\mathbf{if}\ \mathbf{meas}\ x_0\ \mathbf{then}\ \mathbf{inj}_\ell\ ()\ \mathbf{else}\ \mathbf{inj}_r\ (\mathbf{new}\ \mathbf{ff})) \otimes \lambda().x_1$$

The event structure for the strategy is quite straightforward, the only subtlety being that there is a conflict between $\star_\ell^+$ and $(\mathbf{qb}_\ell)^+_{\mathfrak{Q}}$, representing that the left hand side function can output either a value of type $\mathbf{1}$, or a value of type $\mathbf{qubit}$, but not both.

The quantum valuation requires a little more explanation. For $M \in \mathrm{Pos}(\mathfrak{Q}^{\otimes 2})$, we write m_{ij} for its coefficient on the i-th row and the j-th column. If M represents the two qubits given in the context, then the diagonal coefficients m_{11}, m_{22}, m_{33} and m_{44} represent the probability of obtaining when measuring them $(\mathbf{ff}, \mathbf{ff})$, $(\mathbf{tt}, \mathbf{ff})$, $(\mathbf{ff}, \mathbf{tt})$ and $(\mathbf{tt}, \mathbf{tt})$ respectively. In the line $\mathcal{Q}(x)$, we have not yet measured any of the two inputs, so the four are still possible. In the line $\mathcal{Q}(x \cup [\star_\ell^+])$, we have measured the first qubit and obtained true, this is why the coefficients m_{11} and m_{33} have been eliminated. Conversely, in the line $\mathcal{Q}(x \cup [(\mathbf{qb}_\ell)^+_{\mathfrak{Q}}])$, we have measured the first qubit and obtained false, which is why the coefficients m_{22} and m_{44} have been eliminated. In the last three lines, we actually output the second qubit instead of discarding it, hence the appearance of some of the diagonal coefficients of M.

We consider $x = \{(\mathbf{qb}, \mathbf{qb})^-, (\lambda_\ell, \lambda_r)^+, \star_\ell^-, \star_r^-\}$, $x_1 = x \cup \{\star_\ell^+\}$, $x_2 = x \cup \{(\mathbf{qb}_\ell)^+_{\mathfrak{Q}}\}$ and $x_3 = x \cup \{(\mathbf{qb}_r)^+_{\mathfrak{Q}}\}$ The drop associated to them is:

$$\begin{array}{rcl} d(x; x_1, x_2, x_3) & = & \mathcal{Q}(x) \\ & & -\mathcal{Q}(x_1) - \mathbf{Tr}_{\mathfrak{Q}} \circ \mathcal{Q}(x_2) - \mathbf{Tr}_{\mathfrak{Q}} \circ \mathcal{Q}(x_3) \\ & & +\mathbf{Tr}_{\mathfrak{Q}} \circ \mathcal{Q}(x_1 \cup x_3) + \mathbf{Tr}_{\mathfrak{Q}^{\otimes 2}} \circ \mathcal{Q}(x_2 \cup x_3) \end{array}$$

If we apply this equation to $M \in \mathrm{Pos}(\mathfrak{Q}^{\otimes 2})$, we obtain:

$$\begin{array}{rcl} d(x; x_1, x_2, x_3)(M) & = & (m_{11} + m_{22} + m_{33} + m_{44}) \\ & & -(m_{22} + m_{44}) - (m_{11} + m_{33}) - (m_{11} + m_{22} + m_{33} + m_{44}) \\ & & +(m_{22} + m_{44}) + (m_{11} + m_{33}) \\ & = & 0 \end{array}$$

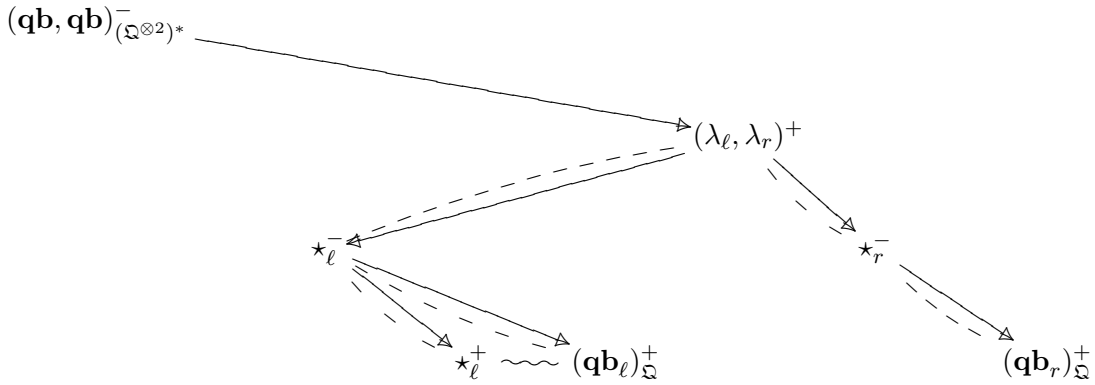
We again obtained 0, but this is not always the case. For example, if we consider $d(x; x_1)$ we would obtain:

$$d(x; x_1)(M) = (m_{11} + m_{22} + m_{33} + m_{44}) - (m_{22} + m_{44}) = m_{11} + m_{33} \geq 0$$

Non-zero drops happen when the extensions $x \subseteq^+ x_1, \ldots, x_n$ do not capture all the possible ways x could be extended, or when the term represented by the strategy has a probability to diverge and never return.

As the definition of quantum strategies is the core of this thesis, we now detail the proof of preservation of the different properties of quantum strategies under interactive composition. The proof given here is very similar to the proof present in [Win11] in the probabilistic case.

$$(\mathbf{qubit}, \mathbf{qubit}) \vdash_{\mathbb{A}} (\mathbf{qubit} \multimap (\mathbf{1} \oplus \mathbf{qubit})) \otimes (\mathbf{qubit} \multimap \mathbf{qubit})$$



Quantum valuation:

$$\forall \varnothing \neq x \subseteq \{(\mathbf{qb}, \mathbf{qb})^-, (\lambda_\ell, \lambda_r)^+, \star_\ell^-, \star_r^-\}$$

$$\begin{array}{lcl}
\mathcal{Q}(\varnothing) & = & \mathbf{id_1} \\
\mathcal{Q}(x) & = & M \mapsto m_{11} + m_{22} + m_{33} + m_{44} \\
\mathcal{Q}(x \cup [\star_\ell^+]) & = & M \mapsto m_{22} + m_{44} \\
\mathcal{Q}(x \cup [(\mathbf{qb}_\ell)^+_{\mathbb{Q}}]) & = & M \mapsto \begin{pmatrix} m_{11} + m_{33} & 0 \\ 0 & 0 \end{pmatrix} \\
\mathcal{Q}(x \cup [(\mathbf{qb}_r)^+_{\mathbb{Q}}]) & = & M \mapsto \begin{pmatrix} m_{11} + m_{22} & m_{31} + m_{42} \\ m_{13} + m_{24} & m_{33} + m_{44} \end{pmatrix} \\
\mathcal{Q}(x \cup [\star_\ell^+] \cup [(\mathbf{qb}_r)^+_{\mathbb{Q}}]) & = & M \mapsto \begin{pmatrix} m_{22} & m_{42} \\ m_{24} & m_{44} \end{pmatrix} \\
\mathcal{Q}(x \cup [(\mathbf{qb}_\ell)^+_{\mathbb{Q}}] \cup [(\mathbf{qb}_r)^+_{\mathbb{Q}}]) & = & M \mapsto \begin{pmatrix} m_{11} & m_{31} & 0 & 0 \\ m_{13} & m_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
\end{array}$$

Term represented:

$x_0 : \mathbf{qubit}, x_1 : \mathbf{qubit} \vdash_{\mathbb{A}} s : (\mathbf{1} \multimap (\mathbf{1} \oplus \mathbf{qubit})) \otimes (\mathbf{1} \multimap \mathbf{qubit})$
$s := (\lambda().\mathbf{if\ meas}\ x_0\ \mathbf{then\ inj}_\ell\ ()\ \mathbf{else\ inj}_r\ (\mathbf{new\ ff})) \otimes \lambda().x_1$

Figure 5.6: Another example of quantum strategy

5.3.4 Properties of the Drop Function

Before tackling interactive composition of quantum strategies, we first need to state various properties about the drop condition. In this Section 5.3.4, we assume $\sigma : A \nrightarrow B$ is a strategy together with a quantum valuation $\mathcal{Q}_\sigma$ which is normalised and oblivious, but does **not** necessarily satisfy the drop condition. For any configuration $x \in \mathcal{C}(S)$, we write x^A and x^B the two configurations of A and B such that $\sigma \cdot x = x^A \parallel x^B$. For $x \subseteq^+ x_1, \ldots, x_n$ some configurations, we define as in Definition 5.3.3:

$$x_\varnothing := x \qquad\qquad x_I := \bigcup_{i \in I} x_i \quad (\forall \varnothing \neq I \subseteq \{1, \ldots, n\})$$

$$d_\sigma(x; x_1, \ldots, x_n) := \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ x_I \in \mathcal{C}(S)}} (-1)^{|I|} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)$$

The drop d_σ is defined here as a linear map, as it might not be completely positive. Proving that it is defined in **CPM** is equivalent to proving that it is greater than 0 for the Loewner order $\sqsubseteq$. We list a sequence of properties of this drop. Their proofs are trivial algebraic transformations. One can refer to [Win11] for similar proofs in the probabilistic case.

Lemma 5.3.4. *For f a permutation of $\{1, \ldots, n\}$, we have*

$$d_\sigma(x; x_1, \ldots, x_n) = d_\sigma(x; x_{f(1)}, \ldots, x_{f(n)})$$

Moreover, if $x_n \supseteq x_{n-1}$ then $d_\sigma(x; x_1, \ldots, x_n) = d_\sigma(x; x_1, \ldots, x_{n-1})$. This allows to unambiguously write $d_\sigma(x; \{x_i \mid 1 \leq i \leq n\})$.

Proof. To prove the first property, we simply use a property of commutation of the sum of linear maps: we reorder the sum $\sum_{\substack{I \subseteq \{1,\ldots,n\} \\ x_I \in \mathcal{C}(S)}}$ into $\sum_{\substack{f(I) \subseteq \{1,\ldots,n\} \\ x_{f(I)} \in \mathcal{C}(S)}}$. To prove the second part, we use commutation of the sum in order to split it depending on whether $n \notin I$, $n \in I$ but $n-1 \notin I$, or $n, n-1 \in I$. We note that $x_{J \cup \{n\}} = x_{J \cup \{n-1,n\}}$ for any $J \subseteq \{1, \ldots, n-2\}$.

$$\begin{aligned} d_\sigma(x; x_1, \ldots, x_n) &= d_\sigma(x; x_1, \ldots, x_{n-1}) \\ &+ \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ n-1 \notin I \ni n \\ x_I \in \mathcal{C}(S)}} (-1)^{|I|} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A) \\ &+ \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ n-1 \in I \ni n \\ x_I \in \mathcal{C}(S)}} (-1)^{|I|} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A) \end{aligned}$$

We can now change the sums so that they range over $J \subseteq \{1, \ldots, n-2\}$ instead.

$$\begin{aligned} d_\sigma(x; x_1, \ldots, x_n) &= d_\sigma(x; x_1, \ldots, x_{n-1}) \\ &+ \sum_{\substack{J \subseteq \{1,\ldots,n-2\} \\ I = J \cup \{n\} \\ x_I \in \mathcal{C}(S)}} (-1)^{|J|+1} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ ^- \!\!\supseteq x^A) \\ &+ \sum_{\substack{J \subseteq \{1,\ldots,n-2\} \\ I = J \cup \{n-1,n\} \\ x_I \in \mathcal{C}(S)}} (-1)^{|J|+2} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ ^- \!\!\supseteq x^A) \\ &= d_\sigma(x; x_1, \ldots, x_{n-1}) \end{aligned}$$

Even though both sums do not use exactly use the same index I, $I = J \cup \{n\}$ for the first and $I = J \cup \{n-1, n\}$ for the second), we know that $x_{J \cup \{n\}} = x_{J \cup \{n-1,n\}}$ so every term of one sum is cancelled by the corresponding term of the other sum. □

Lemma 5.3.5. *If $n > 0$ then*

$$\begin{aligned} d_\sigma(x; x_1, \ldots, x_n) &= d_\sigma(x; x_1, \ldots, x_{n-1}) - \Bigg(\mathcal{H}_B(x^B \subseteq^+ x_n^B) \\ &\quad \circ d_\sigma \left(x_n; \left\{ x_i \cup x_n \;\middle|\; \begin{array}{c} 1 \leq i \leq n-1 \\ x_i \cup x_n \in \mathcal{C}(S) \end{array} \right\} \right) \circ \mathcal{H}_A(x_n^A \ ^- \!\!\supseteq x^A) \Bigg) \end{aligned}$$

Proof. We consider $d_\sigma(x; x_1, \ldots, x_n)$ and split the sum in two, one containing all the terms of $d_\sigma(x; x_1, \ldots, x_{n-1})$, and the other containing the remaining ones. We note that the remaining ones are all pre-composed with

$$\mathcal{H}_A(x_I^A \ ^- \!\!\supseteq x^A) = \mathcal{H}_A(x_I^A \ ^- \!\!\supseteq x_n^A) \circ \mathcal{H}_A(x_n^A \ ^- \!\!\supseteq x^A)$$

and post-composed with

$$\mathcal{H}_B(x^B \subseteq^+ x_I^B) = \mathcal{H}_B(x^B \subseteq^+ x_n^B) \circ \mathcal{H}_B(x_n^B \subseteq^+ x_I^B)$$

So we can use linearity of the composition and factor $\mathcal{H}_A(x_n^A \ ^- \!\!\supseteq x^A)$ and $\mathcal{H}_B(x^B \subseteq^+ x_n^B)$. The sum at the middle is then exactly the expected drop. □

Lemma 5.3.6. *If there exists $1 \leq i \leq n$ such that $x_i = x$, then $d_\sigma(x; x_1, \ldots, x_n) = 0$*

Proof. Since we can reorder the x_k, we can consider without loss of generality that $i = n$. We use Lemma 5.3.5 and obtain:

$$\begin{array}{rcl} d_\sigma(x; x_1, \ldots, x_n) & = & d_\sigma(x; x_1, \ldots, x_{n-1}) \\ & - & \mathcal{H}_B(x^B \subseteq^+ x^B) \circ d_\sigma\left(x; \{x_i \mid 1 \leq i \leq n-1\}\right) \circ \mathcal{H}_A(x^A \supseteq^- x^A) \\ & = & 0 \end{array}$$

□

Lemma 5.3.7. *If $x \subseteq^+ x'_n \subseteq^+ x_n$, then*

$$\begin{array}{rcl} d_\sigma(x; x_1, \ldots, x_n) & = & d_\sigma(x; x_1, \ldots, x'_n) + \Big(\mathcal{H}_B(x^B \subseteq^+ x'^B_n) \\ & & \circ d_\sigma\left(x'_n; \left\{ x_i \cup x'_n \;\middle|\; \begin{array}{c} 1 \leq i \leq n \\ x_i \cup x'_n \in \mathcal{C}(S) \end{array} \right\}\right) \circ \mathcal{H}_A(x'^A_n \supseteq^- x^B) \Big) \end{array}$$

Proof. We use Lemma 5.3.4 and obtain that the drop $d_\sigma(x; x_1, \ldots, x'_n)$ is equal to the drop $d_\sigma(x; x_1, \ldots, x_n, x'_n)$. We then use Lemma 5.3.5 to decompose $d_\sigma(x; x_1, \ldots, x_n, x'_n)$ into $d_\sigma(x; x_1, \ldots, x_n)$ minus $d_\sigma(x'_n, \{x_i \cup x'_n\})$ pre and post-composed by some $\mathcal{H}(-)$. This is equivalent to the expected equation. □

If we iterate the use of Lemma 5.3.7, we obtain that $d_\sigma(x; x_1, \ldots, x_n)$ decomposes as a sum of drop of one-step extensions, composed with some morphisms obtained by $\mathcal{H}$. Formally, we obtain the following proposition.

Proposition 5.3.8. *If for all $y \mathrel{-\!\!\subset}^+ y_1, \ldots, y_m$, we have $d_\sigma(y; y_1, \ldots, y_m) \sqsupseteq 0$, then for all $x \subseteq^+ x_1, \ldots, x_n$, we have $d_\sigma(x; x_1, \ldots, x_n) \sqsupseteq 0$.*

5.3.5 The Drop Condition

We now take σ and τ two quantum strategies from A to B and B to C, so satisfying the obliviousness and drop condition, and look at $\tau \circledast \sigma$. We define the quantum valuation of the interaction as follows, using the fact that every configuration of $T \circledast S$ can be written as $y \circledast x$ with $y \in \mathcal{C}(T)$ and $x \in \mathcal{C}(S)$, as per Proposition 4.3.3.

$$\mathcal{Q}_{\tau \circledast \sigma}(y \circledast x) := \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x) \in \mathbf{CPM}(\mathcal{H}_A(x^A), \mathcal{H}_C(y^C))$$

We assign polarities on $T \circledast S$ as follows:

$$p_{T \circledast S}(e) := \begin{cases} p_{A^\perp}(a) & \text{if } (\tau \circledast \sigma)(e) = a \in A \\ \oplus & \text{if } (\tau \circledast \sigma)(e) = b \in B \\ p_C(c) & \text{if } (\tau \circledast \sigma)(e) = c \in C \end{cases}$$

As already mentioned in the previous chapter, to properly define the polarity of $T \circledast S$, one should add a third polarity, the neutral polarity 0, and take $p_{T\circledast S}(e) = 0$ whenever $(\tau \circledast \sigma)(e) = b \in B$. In the presence of a neutral polarity, the obliviousness condition still applies only to negative extensions, while the drop condition applies to all non-negative extensions. So within the context of this proof, neutral events and positive events are treated the same way.

We consider $z \mathrel{-\!\!\subset}^+ z_1, \ldots, z_n \in \mathcal{C}(T \circledast S)$, writing $z = y \circledast x$ and $z_i = y_i \circledast x_i$. Each of those one-step extensions is either positive in S or positive in T. Up to reordering, we have $0 \leq k \leq n$ such that

$$\forall 1 \leq i \leq k, x \mathrel{-\!\!\subset}^+ x_i \text{ and either } y \mathrel{-\!\!\subset}^- y_i \text{ or } y = y_i$$

$$\forall k < i \leq n, y \mathrel{-\!\!\subset}^+ y_i \text{ and either } x \mathrel{-\!\!\subset}^- x_i \text{ or } x = x_i$$

The following lemma is the core of the proof of preservation of the drop condition by interaction.

Lemma 5.3.9. *We have* $d_\tau(y; y_{k+1}, \ldots, y_n) \circ d_\sigma(x; x_1, \ldots, x_k) = d_{\tau\circledast\sigma}(z; z_1, \ldots, z_n)$

Proof. For $I \subseteq \{1, \ldots, k\}$ and $J \subseteq \{k+1, \ldots, n\}$ we have

$$z_{I\cup J} \in \mathcal{C}(T \circledast S) \iff x_I \in \mathcal{C}(S) \text{ and } y_J \in \mathcal{C}(T)$$

Indeed, $z_{I\cup J}$ is a configuration if and only if both $y_{I\cup J}$ and $x_{I\cup J}$ are configurations and $z_{I\cup J} = y_{I\cup J} \circledast x_{I\cup J}$. Using race-freeness of the games, $y^B_{I\cup J} = x^B_{I\cup J}$ are configurations if and only if y^B_J and x^B_I are configurations. Using receptivity, $y_{I\cup J}$ and $x_{I\cup J}$ are configurations if and only if y^B_J and x^B_I are configurations. For $I \subseteq \{1, \ldots, k\}$ and $J \subseteq \{k+1, \ldots, n\}$ we have

$$\begin{array}{ll} & \mathcal{Q}_\tau(y_J) \circ \mathcal{H}_B(y^B_J \mathrel{-\!\!\supseteq} y^B) \circ \mathcal{H}_B(x^B \subseteq^+ x^B_I) \circ \mathcal{Q}_\sigma(x_I) \\ & \text{(functoriality of } \mathcal{H}) \\ = & \mathcal{Q}_\tau(y_J) \circ \mathcal{H}_B(y^B_J \subseteq^+ y^B_J \cup x^B_I) \circ \mathcal{H}_B(y^B_J \cup x^B_I \mathrel{-\!\!\supseteq} x^B_I) \circ \mathcal{Q}_\sigma(x_I) \\ & \text{(obliviousness of } \mathcal{Q}_\tau \text{ and } \mathcal{Q}_\sigma) \\ = & \mathcal{Q}_\tau(y_{I\cup J}) \circ \mathcal{Q}_\sigma(x_{I\cup J}) \\ & \text{(definition of } \mathcal{Q}_{\tau\circledast\sigma}) \\ = & \mathcal{Q}_{\tau\circledast\sigma}(y_{I\cup J} \circledast x_{I\cup J}) \\ = & \mathcal{Q}_{\tau\circledast\sigma}(z_{I\cup J}) \end{array}$$

This means that

$$\begin{aligned}
&\left(\sum_{\substack{J\subseteq\{k+1,\ldots,n\}\\ y_J\in\mathcal{C}(T)}}(-1)^{|J|} \quad \mathcal{H}_C(y^C \subseteq^+ y_J^C)\circ\mathcal{Q}_\tau(y_J)\circ\mathcal{H}_B(y_J^B \ {}^-\!\supseteq y^B)\right)\\
&\circ\left(\sum_{\substack{I\subseteq\{1,\ldots,k\}\\ x_I\in\mathcal{C}(S)}}(-1)^{|I|} \quad \mathcal{H}_B(x^B \subseteq^+ x_I^B)\circ\mathcal{Q}_\sigma(x_I)\circ\mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)\right)\\
= &\left(\sum_{\substack{I\subseteq\{1,\ldots,k\}\\ x_I\in\mathcal{C}(S)}}\sum_{\substack{J\subseteq\{k+1,\ldots,n\}\\ y_J\in\mathcal{C}(T)}}(-1)^{|I|+|J|} \quad \begin{array}{l}\mathcal{H}_C(y^C \subseteq^+ y_J^C)\circ\mathcal{Q}_\tau(y_J)\circ\mathcal{H}_B(y_J^B \ {}^-\!\supseteq y^B)\\ \circ\mathcal{H}_B(x^B \subseteq^+ x_I^B)\circ\mathcal{Q}_\sigma(x_I)\circ\mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)\end{array}\right)\\
= &\left(\sum_{\substack{I\subseteq\{1,\ldots,k\}\\ J\subseteq\{k+1,\ldots,n\}\\ x_{I\cup J}\in\mathcal{C}(S)\\ y_{I\cup J}\in\mathcal{C}(T)}}(-1)^{|I|+|J|} \quad \mathcal{H}_C(y^C \subseteq^+ y_J^C)\circ\mathcal{Q}_{\tau\circledast\sigma}(z_{I\cup J})\circ\mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)\right)
\end{aligned}$$

Or, in other words:

$$d_\tau(y; y_{k+1},\ldots,y_n)\circ d_\sigma(x; x_1,\ldots,x_k) = d_{\tau\circledast\sigma}(z; z_1,\ldots,z_n) \qquad \square$$

Lemma 5.3.10. *If $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ are quantum strategies, and $\tau \odot \sigma$ with the quantum valuation $\mathcal{Q}_{\tau\odot\sigma}(y \odot x) = \mathcal{Q}_\tau(y)\circ\mathcal{Q}_\sigma(x)$ satisfies normalisation and obliviousness, then it satisfies the drop condition.*

Proof. Lemma 5.3.9, together with the fact that the composition of completely positive maps is completely positive, states that since σ and τ satisfy the drop condition, then $\tau \circledast \sigma$ satisfies the drop condition for one-step extensions. Using Proposition 5.3.8 it follows that $\tau \circledast \sigma$ satisfies the drop condition. We then note that whenever (x, y) are a minimal matching pair of configurations, then $\mathcal{Q}_{\tau\circledast\sigma}(y \circledast x) = \mathcal{Q}_{\tau\odot\sigma}(y \odot x)$. So for $(y \odot x) \subseteq^+ (y_1 \odot x_1), \ldots, (y_n \odot y_n)$ we have

$$d_{\tau\odot\sigma}(y \odot x; y_1 \odot x_1, \ldots, y_n \odot x_n) = d_{\tau\circledast\sigma}(y \circledast x; y_1 \circledast x_1, \ldots, y_n \circledast x_n)$$

(And since we put positive polarities on events at the middle of $T \circledast S$, all the $y \circledast x \subseteq y_i \circledast x_i$ are positive extensions). So $\tau \odot \sigma$ satisfies the drop condition too. $\square$

5.3.6 The Category of Quantum Games and Strategies

Proposition 5.3.11. *For $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ two quantum strategies, we define*

$$\mathcal{Q}_{\tau\odot\sigma}(y \odot x) := \mathcal{Q}_\tau(y)\circ\mathcal{Q}_\sigma(x)$$

The strategy $\tau \odot \sigma$ together with the valuation $\mathcal{Q}_{\tau\odot\sigma}(y \odot x)$ is a quantum strategy.

Proof. The drop condition follows from Lemma 5.3.10. The normalisation is trivially true. We need to prove obliviousness.

We consider $y \odot x \subseteq^- y' \odot x' \in \mathcal{C}(T \odot S)$, then using Lemma A.1.4, we obtain that $y \subseteq^- y'$ and $x \subseteq^- x'$, with all the extensions being on the $A^\perp$ and C side (and none on the B side). Using obliviousness, we have $\mathcal{Q}_\sigma(x') = \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(x_A \subseteq^+ x'_A)$ and $\mathcal{Q}_\tau(y') = \mathcal{H}_C(y'_C \supseteq^- y_C) \circ \mathcal{Q}_\tau(y)$. It follows that we have $\mathcal{Q}_{\tau\odot\sigma}(y' \odot x') = \mathcal{H}_C(y'_C \supseteq^- y_C) \circ \mathcal{Q}_{\tau\odot\sigma}(y \odot x) \circ \mathcal{H}_A(x_A \subseteq^+ x'_A)$. □

Theorem 5.3.12. *Quantum games and strategies, up to isomorphism, form a CpCC* $(\mathbf{QStrat}, \|, \varnothing, (_)^\perp)$.

The bifunctor $\|$ is given by $\mathcal{Q}_{\sigma\|\tau}(x \parallel y) := \mathcal{Q}_\sigma(x) \otimes \mathcal{Q}_\tau(y)$. This theorem simply follows from the fact that both **Strat** and **CPM** are CpCCs.

5.3.7 The Polarised Quantum Valuation

As **CPM** is a compact closed category, we can move Hilbert spaces from the domain to the codomain of the quantum valuation, or vice-versa. More explicitly:

Definition 5.3.13. *We take $\sigma : A \nrightarrow B$ a quantum strategy. For $x \in \mathcal{C}(S)$, we write $\sigma\, x = x^- \sqcup x^+$ with x^- containing only negative events, and x^+ only positive ones. We define $\mathcal{Q}^{-,+}_\sigma(x) \in \mathbf{CPM}(\mathcal{H}_{A\|B^\perp}(x^-), \mathcal{H}_{A^\perp\|B}(x^+))$ from $\mathcal{Q}_\sigma(x)$ using the compact closure of* **CPM**.

We put the emphasis on the fact that the domain of the valuation uses $\mathcal{H}_{A\|B^\perp}$ while the codomain uses $\mathcal{H}_{A^\perp\|B}$, which is its dual. Looking back at the example from Fig. 5.5, we obtain the following

	$(\mathbf{qb}_\ell)^-_{\mathbb{Q}^*} \notin x$	$(\mathbf{qb}_\ell)^-_{\mathbb{Q}^*}, \star_0^+ \in x$	$(\mathbf{qb}_\ell)^-_{\mathbb{Q}^*}, \star_1^+ \in x$	$(\mathbf{qb}_\ell)^-_{\mathbb{Q}^*} \notin x$
$\mathcal{Q}(x)$	$\mathbf{id_1}$	$z \mapsto \begin{pmatrix} z & 0 \\ 0 & 0 \end{pmatrix}$	$z \mapsto \begin{pmatrix} 0 & 0 \\ 0 & z \end{pmatrix}$	$z \mapsto \begin{pmatrix} z & 0 \\ 0 & z \end{pmatrix}$
$\mathcal{Q}^{-,+}(x)$	$\mathbf{id_1}$	$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a$	$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto d$	$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a + d$

When looking at $\mathcal{Q}^{-,+}$, we directly observe the **CPM** morphism corresponding to the operation behaviour, here a measurement. The fact that this measurement is under a λ-abstraction in the term t_Q is no longer observable in $\mathcal{Q}^{-,+}$. Though this information is not lost as it is contained in the fact that the event $(\mathbf{qb}_\ell)^-_{\mathbb{Q}^*}$ is at the right hand side of the game.

A similar definition $\mathcal{Q}^{X,Y}_\sigma$ could be made for any partition $\sigma\, x = X \sqcup Y$, but we focus on $\mathcal{Q}^{-,+}_\sigma$ as it has a very interesting property described in Theorem 5.3.15. We first note it satisfies properties of normalisation, obliviousness and drop condition similar to the quantum valuation $\mathcal{Q}_\sigma$.

Proposition 5.3.14. *For $\sigma : A \nrightarrow B$ a quantum strategy, $\mathcal{Q}_\sigma^{-,+}$ satisfies normalisation, obliviousness, and the drop condition. Formally, if for any $x \in \mathcal{C}(S)$ we write $\sigma\, x = x^- \sqcup x^+$, then we have*

Normalisation $\mathcal{Q}_\sigma^{-,+}(\varnothing) = \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$

Obliviousness $x \subseteq^- x' \implies \mathcal{Q}_\sigma^{-,+}(x') = \mathcal{Q}_\sigma^{-,+}(x) \circ \mathcal{H}_{A\|B^\perp}(x^- \subseteq^+ x'^-)$

Drop condition $x \subseteq^+ x_1, \ldots, x_n \implies d_\sigma^{-,+}(x; x_1, \ldots, x_n)$ *is defined in* **CPM**, *where*

$$d_\sigma^{-,+}(x; x_1, \ldots, x_n) := \mathcal{Q}_\sigma^{-,+}(x) - \sum_{\substack{\varnothing \neq I \subseteq \{1,\ldots,n\} \\ x_I \in \mathcal{C}(S)}} (-1)^{|I|-1} \mathcal{H}_{A^\perp\|B}(x^+ \subseteq^+ x_I^+) \circ \mathcal{Q}_\sigma^{-,+}(x_I)$$

and $x_I := \bigcup_{i \in I} x_i$.

Moreover, for $\sigma : A \nrightarrow B$ a strategy, and $\mathbb{Q}$ a function satisfying the above conditions, there exists a unique quantum strategy σ such that $\mathcal{Q}_\sigma^{-,+} = \mathbb{Q}$.

The proof is immediate from the compact closure of **CPM**. Looking back again to the example from Fig. 5.5 and the $\mathcal{Q}^{-,+}$ we described above, we note that $\mathcal{Q}^{-,+}(x)$ is always a superoperator in this example, in contrast with $\mathcal{Q}$ which might not be. We can prove this is always the case.

Theorem 5.3.15. *For $\sigma : A \nrightarrow B$ a quantum strategy, $\mathcal{Q}_\sigma^{-,+}(x)$ is always a superoperator, i.e., $\forall x \in \mathcal{C}(S), \mathcal{Q}_\sigma^{-,+}(x) \in \mathbf{CPM}_{\leq 1}(\mathcal{H}_{A\|B^\perp}(x^-), \mathcal{H}_{A^\perp\|B}(x^+))$.*

Proof. We have $\mathcal{Q}_\sigma^{-,+}(\varnothing) \in \mathbf{CPM}_{\leq 1}$. If $x \subseteq^- x'$ and $\mathcal{Q}_\sigma^{-,+}(x) \in \mathbf{CPM}_{\leq 1}$, using obliviousness we obtain

$$\mathcal{Q}_\sigma^{-,+}(x') = \mathcal{Q}_\sigma^{-,+}(x) \circ (\mathbf{id}_{\mathcal{H}_{A\|B^\perp}(x'^-)} \otimes \mathbf{Tr}_{\mathcal{H}_{A\|B^\perp}(x'^- \setminus x^-)}) \in \mathbf{CPM}_{\leq 1}$$

If $x \subseteq^+ x'$ and $\mathcal{Q}_\sigma^{-,+}(x) \in \mathbf{CPM}_{\leq 1}$, using the drop condition we obtain

$$\mathcal{Q}_\sigma^{-,+}(x) - (\mathbf{id}_{\mathcal{H}_{A^\perp\|B}(x'^+)} \otimes \mathbf{Tr}_{\mathcal{H}_{A^\perp\|B}(x'^+ \setminus x^+)}) \circ \mathcal{Q}_\sigma^{-,+}(x') \in \mathbf{CPM}$$

$$\implies \mathbf{Tr} \circ \mathcal{Q}_\sigma^{-,+}(x') \sqsubseteq \mathbf{Tr} \circ \mathcal{Q}_\sigma^{-,+}(x)$$

So $\mathcal{Q}_\sigma^{-,+}(x') \in \mathbf{CPM}_{\leq 1}$. □

As stated earlier, superoperators correspond to "physically realisable operations" obtained from creation, measurement and unitary primitives. Theorem 5.3.15 shows that our model of quantum computation only uses valuations that correspond to actually plausible quantum operations. This contrasts with the quantum relational model, where while we could express a similar notion of "valuation from negative to positive", this is not always a

superoperator, and it being a superoperator is not preserved under relational composition. In fact, we observe a similar behaviour with strategies: if we only required strategies to be such that $\mathcal{Q}^{-,+}$ was a superoperator (instead of normalisation, obliviousness and drop condition), this condition would not be preserved under interactive composition. What is preserved is the much stronger triple "normalisation, obliviousness and drop condition", which relies on the causal structure of the strategy, information absent from the relational model.

Lastly, we note that probabilistic strategies were a special case of quantum strategies:

Proposition 5.3.16. *We have a full and faithful functor from* **PStrat** *to* **QStrat** *which preserves all the structure.*

Proof. For A a game, we associate to each of its events the Hilbert space $\mathbf{1}$. We note that $(_)^\perp$ is preserved by the functor, but only up to the natural isomorphism between $\mathbf{1}$ and $\mathbf{1}^*$. For $\sigma : A \nrightarrow B$ a probabilistic strategy, we define $\mathcal{Q}^{-,+}_\sigma(x) := v_\sigma(x) \cdot \mathbf{id_1}$. This is faithful by definition. Fullness follows from Theorem 5.3.15 and the fact that $\mathbf{CPM}_{\leq \mathbf{1}}(\mathbf{1}, \mathbf{1}) \cong [0, 1]$. □

5.4 Payoff Games and Winning Strategies

In this section, we introduce a notion of payoff on games, and winner on strategies. They have two purposes

1. Defining the set of configurations of a strategy that are acceptable stopping points of a computation. In some ways, those correspond to "observable" configurations, and will be used to characterise the observational equivalence of QΛ in the proof of full abstraction. Those correspond to points of the web of the relational model.

2. Enforcing strict linearity when interpreting LQΛ.

We note that the first purpose can also be fulfilled through Question/Answer annotations on events and a well-bracketing restriction on strategies as in [Cas17], defining those acceptable stopping points as "every question has been answered". In this thesis, as the notion of payoff tackles both the problem of strict linearity and the problem of stopping points, we have no need for Questions/Answers annotations and well-bracketing. The notions of payoff we use here comes from [Mel05].

Definition 5.4.1. *A quantum payoff game is a quantum game A together with a payoff function $\kappa_A : \mathcal{C}(A) \to \{-1, 0, +1\}$. We define the payoff for the empty game $\varnothing$ as $\kappa_\varnothing(\varnothing) := 0$. A quantum payoff strategy is a quantum strategy between two quantum payoff games.*

Configurations of payoff $+1$ are said to be *winning* (for Player), configurations of payoff -1 are said *losing* (for Player), and those of payoff 0 are said *drawing*, and are considered "acceptable stopping point for a play". For example, the payoff game for $\mathbf{1} \multimap \mathbf{1}$ is:

$$\begin{array}{lccc} (\mathrm{LQ}\Lambda) & \mathbf{1} & \multimap & \mathbf{1} \\ & \lambda^+ & & \\ & \downarrow & & \\ & \star^- & & \\ & \searrow & & \\ & & \star^+ & \end{array} \qquad \begin{array}{l} \kappa(\varnothing) = -1 \\ \kappa(\{\lambda^+\}) = 1 \\ \kappa(\{\lambda^+, \star^-\}) = -1 \\ \kappa(\{\lambda^+, \star^-, \star^+\}) = 0 \end{array}$$

In this game, the payoff constrains Player and Opponent to continue playing until the end. This represents a strictly linear behaviour, as the function described by the game is forced (under penalty of losing) to be called by Opponent at some point. Conversely, the following game describes an affine function, *i.e.*, a function that can remain unused if Opponent wishes so:

$$\begin{array}{lccc} (\mathrm{AQ}\Lambda) & \mathbf{1} & \multimap & \mathbf{1} \\ & \lambda^+ & & \\ & \downarrow & & \\ & \star^- & & \\ & \searrow & & \\ & & \star^+ & \end{array} \qquad \begin{array}{l} \kappa(\varnothing) = -1 \\ \kappa(\{\lambda^+\}) = 0 \\ \kappa(\{\lambda^+, \star^-\}) = -1 \\ \kappa(\{\lambda^+, \star^-, \star^+\}) = 0 \end{array}$$

When considering the parallel composition $A \parallel B$ of games, a choice arises for the payoff: should a winning configuration be winning in both A and B, or in at least one of them? This choice splits $\parallel$ into two different monoidal structures.

Definition 5.4.2. *We define two bifunctors $\parr$ and $\boxtimes$ on quantum payoff games and strategies as follows.*

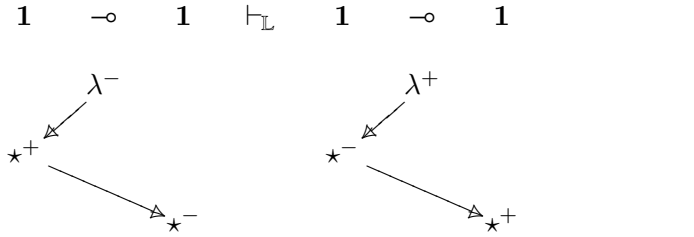
- *$A \parr B$ and $A \boxtimes B$ have $A \parallel B$ as their underlying quantum game.*
- *$\kappa_{A \parr B}(x \parallel y)$ and $\kappa_{A \boxtimes B}(x \parallel y)$ are computed from $\kappa_A(x)$ and $\kappa_B(y)$ as in Fig. 5.7.*
- *the quantum payoff strategies $\sigma \parr \tau$ and $\sigma \boxtimes \tau$ are simply obtained as the quantum strategy $\sigma \parallel \tau$ from* **QStrat**.

We extend $(_)^\perp$ to quantum payoff games with $\kappa_{A^\perp}(x) := -\kappa_A(x)$.

$\parr$	-1	0	+1
-1	-1	-1	+1
0	-1	0	+1
+1	+1	+1	+1

$\boxtimes$	-1	0	+1
-1	-1	-1	-1
0	-1	0	+1
+1	-1	+1	+1

Figure 5.7: Payoff functions for the parallel composition



$$\kappa(x) = \begin{cases} 1 & \text{if } x = y \parallel z \text{ with } y = \varnothing \text{ or } y = \{\lambda^-, \star^+\} \text{ or } z = \{\lambda^+\} \\ 0 & \text{if } x = \{\lambda^-, \star^+, \star^-\} \parallel \{\lambda^+, \star^-, \star^+\} \\ -1 & \text{otherwise} \end{cases}$$

Figure 5.8: Example of a payoff game.

We note that $(A \parr B)^\perp = A^\perp \boxtimes B^\perp$. We use the notation $\boxtimes$ instead of the usual $\otimes$, as we reserve $\otimes$ for the monoidal product corresponding to the $\otimes$ of the **QRel** language.

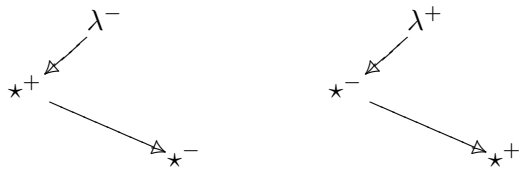
For example, the game representing $\mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} \mathbf{1} \multimap \mathbf{1}$, which we compute as $A^\perp \parr A$ with A the game for $\mathbf{1} \multimap \mathbf{1}$, is described in Fig. 5.8. Similarly, the game representing its affine counterpart $\mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{A}} \mathbf{1} \multimap \mathbf{1}$ is represented in Fig. 5.9.

Definition 5.4.3. *A winning quantum strategy σ from a quantum payoff game A to a quantum payoff game B is a quantum payoff strategy $\sigma : A \nrightarrow B$ such that for all $x \in \mathcal{C}(S)$ $\oplus$-covered, i.e., all its maximal events are positive, we have $\kappa_{A^\perp \parr B}(\sigma(x)) \geq 0$.*

Configurations that are $\oplus$-covered must be understood as configurations where Player saw and reacted to all Opponent moves. They are a rough equivalent in concurrent games of "after a Player move" from sequential games. For example, in Fig. 5.10, the two strategies represent terms that are typed in LQΛ, so they are winning[3]. Formally, on the left the configurations $\{\lambda^-, \star^+\}$ and $\{\lambda^-, \star^+, \star^-, \lambda^+\}$ have payoff 1 in the game, and the configurations $\{\lambda^-, \star^+, \star^-, \lambda^+, \star^-, \star^+\}$ have payoff zero, so all the $\oplus$-covered configurations have non-negative payoff. On the right, there is no $\oplus$-covered configuration so all the $\oplus$-covered configurations have non-negative payoff. As another example, in Fig. 5.11, the strategies correspond to terms that are not well-typed in LQΛ. In both strategies, the configuration

[3]In fact, we will only use winning strategies to interpret terms of LQΛ and AQΛ.

$$\mathbf{1} \quad \multimap \quad \mathbf{1} \quad \vdash_{\mathbb{A}} \quad \mathbf{1} \quad \multimap \quad \mathbf{1}$$



$$\kappa(x) = \begin{cases} 1 & \text{if } x = y \parallel z \text{ with } y = \varnothing \text{ or } y = \{\lambda^-, \star^+\} \\ 0 & \text{if } x = y \parallel z \text{ with } \begin{cases} y = \{\lambda^-\} \text{ or } y = \{\lambda^-, \star^+, \star^-\} \\ \text{and} \\ z = \{\lambda^+\} \text{ or } z = \{\lambda^+, \star^-, \star^+\} \end{cases} \\ -1 & \text{otherwise} \end{cases}$$

Figure 5.9: Example of payoff game.

$\{\lambda^-, \lambda^+\}$ is $\oplus$-covered and has payoff -1. However, if instead we considered them as terms of AQΛ, *i.e.*, considered the strategies τ_0 and τ_1 on the game for $\mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{A}} \mathbf{1} \multimap \mathbf{1}$ instead of $\mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} \mathbf{1} \multimap \mathbf{1}$, then both strategies would be winning as $\{\lambda^-, \lambda^+\}$ would now have payoff 0. This is one of the few differences when interpreting AQΛ and LQΛ: the game for $A \multimap B$ does not have the same payoff on the singleton configuration $\{\lambda^+\}$, as in AQΛ Opponent can choose not to call the the function and stop at $\{\lambda^+\}$ on a tie (payoff 0) while in LQΛ Opponent must call the function or lose (payoff 1).

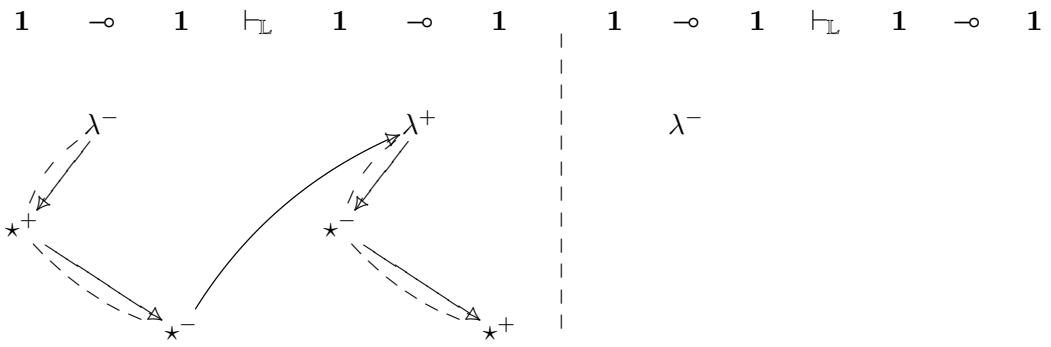
Quantum payoff games and winning strategies, up to isomorphism, form a category. Indeed, as shown below copy-cat satisfies the winning condition, and this winning condition is preserved under interactive composition.

Lemma 5.4.4. *The copy-cat strategy is winning.*

Proof. We consider $\text{ax}_A : A \twoheadrightarrow A$ with A a quantum payoff game. We consider $x \in \text{CC}_A$, and we recall that we have $x = y \parallel z$. Using Lemma A.2.3, we know that if x is $\oplus$-covered then $y = z$. In particular, we have $\kappa_{A^\perp}(y) = -\kappa_A(z)$, meaning they are either both 0, or -1 and $+1$, and in both cases their $\parr$ is non-negative. It follows that if x is $\oplus$-covered, then $\kappa_{A^\perp \parr A}(\sigma\, x) \geq 0$. □

To show that the winning condition is preserved under interactive composition, we start by recalling that Lemma A.2.1 proves that $\oplus$-coveredness is well-behaving with the interactive composition: a configuration $y \odot x$ is $\oplus$-covered if and only if both y and x are $\oplus$-covered.

Lemma 5.4.5. *The interactive composition of two winning quantum strategies is winning.*



Quantum valuations

$$\mathcal{Q}_{\sigma_0}(x) = \mathbf{id_1} \qquad \mathcal{Q}_{\sigma_1}(y) = \mathbf{id_1}$$

Terms represented

$$f : \mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} f\ (); \lambda().() : \mathbf{1} \multimap \mathbf{1} \qquad f : \mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} \bot : \mathbf{1} \multimap \mathbf{1}$$

Figure 5.10: Two winning strategies σ_0 (left) and σ_1 (right)

Proof. We consider $\sigma : A \twoheadrightarrow B$ and $\tau : B \twoheadrightarrow C$. We take $y \odot x \in \mathcal{C}(T \odot S)$ $\oplus$-covered. By Lemma A.2.1, we have x and y $\oplus$-covered too. It means that $\kappa_{A^\perp \parr B}(\sigma\, x) \geq 0$ and $\kappa_{B^\perp \parr C}(\tau\, y) \geq 0$. We write $\sigma\, x = x_A \parallel x_B$ and $\tau\, y = y_B \parallel y_C$. Looking at the definition of $\parr$, and noting that $\kappa_{B^\perp}(y_B) = -\kappa_B(x_B)$, we remark the following:

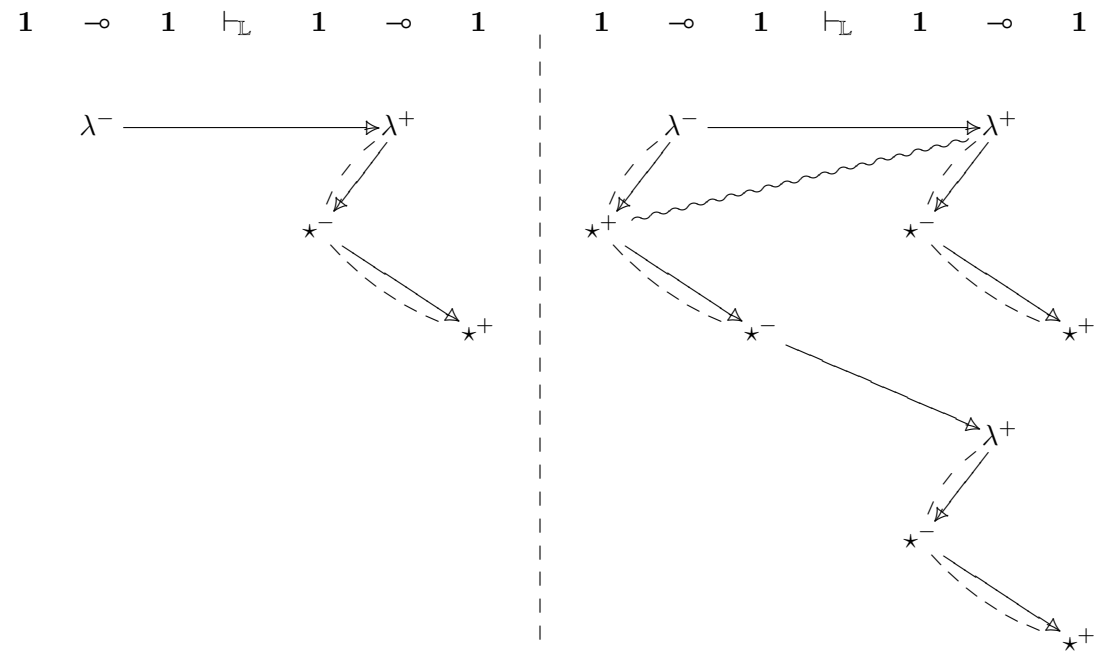
$$\begin{aligned}
\kappa_{A^\perp}(x_A) < 0 &\implies \kappa_B(x_B) > 0 \\
&\implies \kappa_{B^\perp}(y_B) < 0 \\
&\implies \kappa_C(y_C) > 0 \\
&\implies \kappa_{A^\perp \parr C}(x_A \parallel y_C) > 0
\end{aligned}$$

$$\begin{aligned}
\kappa_{A^\perp}(x_A) = 0 &\implies \kappa_B(x_B) \geq 0 \qquad (\text{and } \kappa_{A^\perp}(x_A) = 0) \\
&\implies \kappa_{B^\perp}(y_B) \leq 0 \qquad (\text{and } \kappa_{A^\perp}(x_A) = 0) \\
&\implies \kappa_C(y_C) \geq 0 \qquad (\text{and } \kappa_{A^\perp}(x_A) = 0) \\
&\implies \kappa_{A^\perp \parr C}(x_A \parallel y_C) \geq 0
\end{aligned}$$

$$\kappa_{A^\perp}(x_A) > 0 \implies \kappa_{A^\perp \parr C}(x_A \parallel x_C) > 0$$

So we always have $\kappa_{A^\perp \parr C}(x_A \parallel x_C) \geq 0$. □

Proposition 5.4.6. *Quantum payoff games and winning quantum strategies, up to iso-*



Quantum valuations

$$\mathcal{Q}_{\tau_0}(x) = \mathbf{id_1} \qquad \mathcal{Q}_{\tau_1}(y) = \begin{cases} \mathbf{id_1} & \text{whenever } |y| \leq 1 \\ \frac{1}{2}\mathbf{id_1} & \text{otherwise} \end{cases}$$

Terms represented

$$f : \mathbf{1} \multimap \mathbf{1} \not\vdash_{\mathbb{L}} \lambda().() : \mathbf{1} \multimap \mathbf{1} \qquad f : \mathbf{1} \multimap \mathbf{1} \not\vdash_{\mathbb{L}} \left(\frac{1}{2}f() + \frac{1}{2}()\right); \lambda().() : \mathbf{1} \multimap \mathbf{1}$$

Figure 5.11: Two non-winning strategies τ_0 (left) and τ_1 (right)

morphism, form two SMCs $(\mathbf{QCG}, \parr, \varnothing)$ *and* $(\mathbf{QCG}, \boxtimes, \varnothing)$. *In fact,* $(\mathbf{QCG}, \|, \boxtimes, \varnothing, (_)^\perp)$ *forms a linearly distributive category with negation, so a* $\star$*-autonomous category in light of their equivalence.*

5.5 The Freyd Category of Quantum Arenas and Strategies

5.5.1 The Category QA

In order to build a semantics for QΛ, we first prove that we have a denotational model for Λ. For that, we just need to show that quantum games and strategies form a non-trivial distributive CFC with a bottom. We place ourselves in a subcategory of games and strategies, where we choose to only consider games that are well-formed, which we call arenas.

Definition 5.5.1. *A quantum arena* A *is a quantum payoff game which is:*

Positive *All its minimal events are positive.*

Well-opened *All its minimal events are in pairwise conflict.*

Forest-like *Whenever* $a \leq_A b \geq_A a'$, *we have* $a \leq_A a'$ *or* $a \geq_A a'$.

Initially Losing $\kappa_A(\varnothing) = -1$

It is said to be affine if moreover for every minimal event e, *the payoff of* $\{e\}$ *is* 0. *A quantum strategy* $\sigma : A \nrightarrow B$ *between quantum arenas is said to be*

Negative *if the minimal events of* S *are negative. We note that they are necessarily sent to* $A^\perp$ *per* σ.

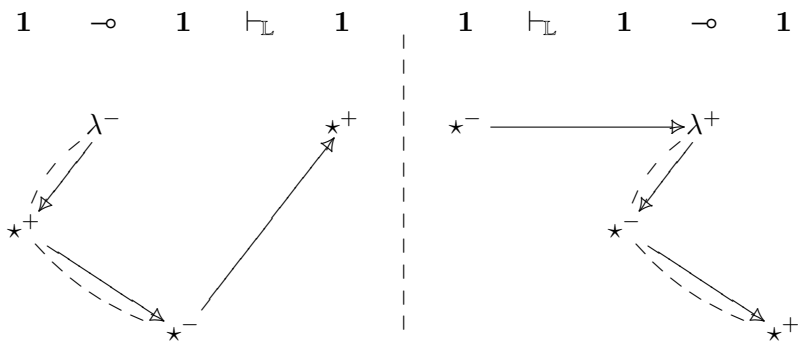
Thunkable *if it is negative and for every minimal event* m^- *of* S, *there exists a unique event* m^+, *called runner-up, such that* $m^- \rightarrowtriangle_S r^+$. *Moreover, it satisfies:*

$$\sigma(r) \in B \text{ and } d_\sigma(\{m\}; \{m, r\}) = 0$$

Visible *if it satisfies Definition 6.2.2.*

Thunkable strategies are to be thought of as "answering immediately", hence the runner-up events being on the B side, "answering only one thing", hence the uniqueness, and "answering with probability one[4]", hence the last condition. Negative strategies will form the computation category of our Freyd category, while thunkable ones will form the value category. In Fig. 5.12, we show the strategy σ_0 for the term $f()$ and the strategy τ_0

[4]The formal equivalent of "the configuration x has probability one" is "$\mathcal{Q}^{-,+}(x)$ is trace preserving", *i.e.*, $\mathbf{Tr} \circ \mathcal{Q}^{-,+}(x) = \mathbf{Tr}$.



Quantum valuation:

$$\mathcal{Q}_{\sigma_0}(x) = \mathbf{id_1} \qquad \mathcal{Q}_{\tau_0}(y) = \mathbf{id_1}$$

Term represented:

$$f : \mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} f\ () : \mathbf{1} \qquad \vdash_{\mathbb{L}} \lambda x.x : (\mathbf{1} \multimap \mathbf{1})$$

Figure 5.12: A non-thunkable strategy σ_0 (left) and a thunkable strategy τ_0 (right)

for the term $\lambda x.x$. The first term is not a value, and its strategy is not thunkable. The second term is a value, and its strategy is thunkable.

Visible strategies are strategies that respect a certain notion of scope, and at a certain point of the computation described by the strategy, Player can only use moves of the games that are within the scope. In concurrent game semantics, non-visible strategies usually correspond to strategies using shared memory as in [CCHW18]. We postpone the definition of visibility to Section 6.2.1, as this property is only required to ensure the absence of deadlocks, which we use in the proof of full abstraction.

Definition 5.5.2. *We write* $\mathbf{QA}$ *for the category of quantum arenas and visible winning negative quantum strategies. We write* $\mathbf{QA}_t$ *for its subcategory restricted to thunkable strategies. We write* $\mathbf{QA}^a$ *and* $\mathbf{QA}^a_t$ *for the full subcategories restricted to affine arenas.*

We note that quantum arenas decompose in the following way:

- Minimal events are positive and in conflict.
- Each minimal event a_i is "followed" by a negative forest-like quantum payoff game A_i, *i.e.*, the set of events $\{e > a_i \mid e \in |A|\}$ forms a negative forest-like quantum payoff game A_i.

As such, we write the decomposition of quantum arenas $A = \bigoplus_{i \in I} \downarrow_{(a_i : H_i)} A_i$, where a_i^+ are the minimal events, $H_i = \mathcal{H}_A(a_i)$, and A_i the negative games. The operation $\downarrow$ is

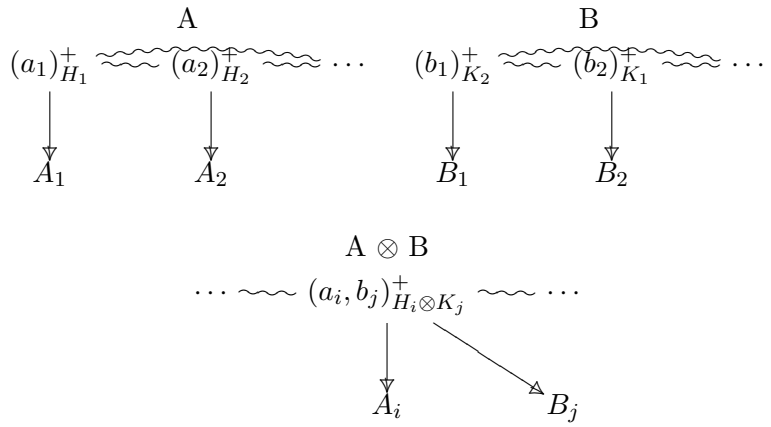


Figure 5.13: The tensor $\otimes$ of two arenas

called positive shift, and corresponds to "adding a minimal positive events before every other events". The payoff of the empty configuration is always -1, and the payoff of other configurations $\{a_i\} \sqcup x$ is given by $\kappa_{A_i}(x)$.

5.5.2 The Premonoidal Tensor

We now define the operation $\otimes$. This operation is similar to $\boxtimes$, except that instead of putting in parallel the two arenas, it synchronises the minimal events together. This construction is familiar from call-by-value games [HY97], and matches the fact that a value on $A \otimes B$ is a pair of values.

Definition 5.5.3. *We define the quantum arena* $\mathbf{1}$ *as* $\downarrow_{(\star:\mathbf{1})} \varnothing$*. For* A *and* B *two quantum arenas, we define* $A \otimes B$ *as in Fig. 5.13. Formally:*

$$\begin{array}{lcl} A & = & \bigoplus_{i \in I} \downarrow_{(a_i : H_i)} A_i \\ B & = & \bigoplus_{j \in J} \downarrow_{(b_j : K_j)} B_j \\ A \otimes B & := & \bigoplus_{(i,j) \in I \times J} \downarrow_{((a_i, b_j) : (H_i \otimes K_j))} (A_i \boxtimes B_j) \end{array}$$

We want to extend this operation $\otimes$ to strategies, so as to obtain premonoidal product of an SFC. For $\sigma : A \nrightarrow B$ and $\sigma' : A' \nrightarrow B'$, we want to define a strategy from $A \otimes A'$ to $B \otimes B'$. Following the premonoidal structure (Section 1.2.1), we must provide functors $C \otimes _$ and $_ \otimes C$ for any object C, then

- we can use the left-then-right tensor $\sigma \otimes^{\ell} \sigma' := (\sigma' \otimes B) \odot (A \otimes \sigma)$, or
- we can use the right-then-left tensor $\sigma \otimes^{r} \sigma' := (B' \otimes \sigma) \odot (\sigma' \otimes A)$, or

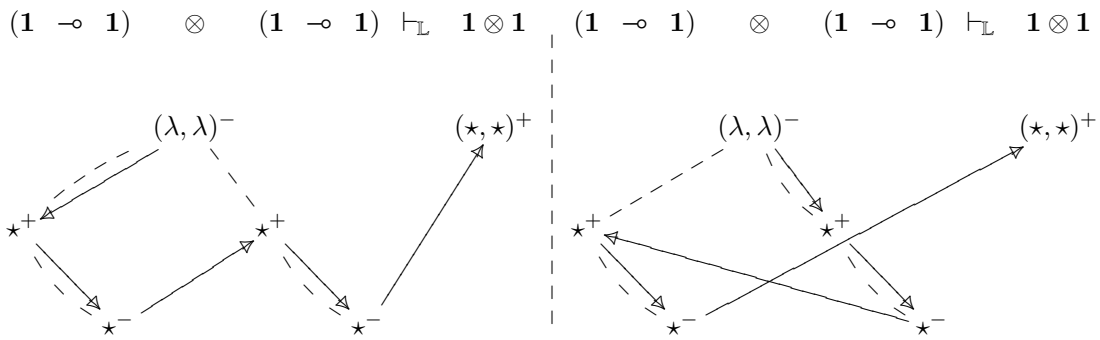


Figure 5.14: The strategies $\sigma_0 \otimes^\ell \sigma_0$ (left) and $\sigma_0 \otimes^r \sigma_0$ (right)

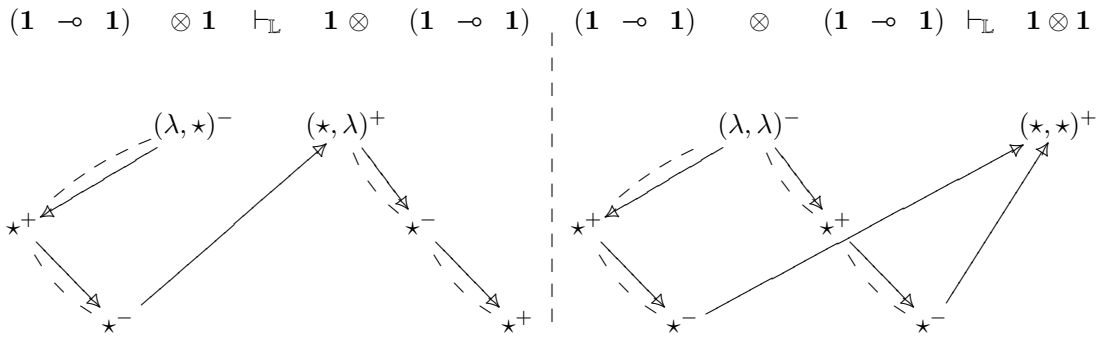


Figure 5.15: The strategies $\sigma_0 \otimes \tau_0$ (left) and $\sigma_0 \otimes^p \sigma_0$ (right)

- if any of the two σ and σ' comes from the value category, *i.e.*, is thunkable, then both $\otimes^\ell$ and $\otimes^r$ coincide and we can write $\sigma \otimes \sigma'$ unambiguously.

In Fig. 5.14, we give an example of a case where $\otimes^\ell$ and $\otimes^r$ differ: both are tensors of σ_0 from Fig. 5.12 with itself, and the first one starts by exploring the left hand side of the game while the second one starts by exploring the right hand side of the game. To illustrate the third item, we show on the left hand side of Fig. 5.15 the tensor of σ_0 and τ_0. Notice that the second move of τ_0, which is the first move playing on the game for $\mathbf{1}$, is postponed so that it synchronised with the fourth move of σ_0, which is the first move playing on the game for $\mathbf{1} \multimap \mathbf{1}$. Since τ_0 is thunkable, there is no ambiguity between which of σ_0 and τ_0 gets to act first and explore its side of the game, as τ_0 has "nothing to do" before the event $(\star, \lambda)^+$.

To define the SFC, instead of defining first the functors $C \otimes _$ and $_ \otimes C$ as expected, we will use a structure stronger[5] than the notion of premonoidal category: the notion

[5]Stronger in the sense that the parallel tensor generates a premonoidal category, but not all premonoidal categories have a parallel tensor.

of parallel tensor. Indeed, while premonoidal products come with a notion of evaluation order left-then-right $\otimes^\ell$ and right-then-left $\otimes^r$, in our concrete case, taking advantage of parallelism in our strategies, we have a third evaluation order $\otimes^p$ which is symmetric:

$$(\sigma \otimes^p \sigma') \circ \mathrm{br}_{A',A} \cong \mathrm{br}_{B',B} \circ (\sigma' \otimes^p \sigma)$$

and while it is not a bifunctor (as illustrated below), it still satisfies a property that we call semi-bifunctoriality. This property implies that $\tau \otimes^p _$ is functorial whenever τ is thunkable, meaning that $C \otimes _ := \mathbf{id}_C \otimes^p _$ and its symmetric $_ \otimes C$ are two functors, hence generates a premonoidal category. Moreover, this semi-bifunctoriality implies that all the thunkable morphisms are in the centre of the premonoid, hence generates an SFC. To our knowledge, this property does not appear in the literature. The property of symmetry means that $\sigma \otimes^p \tau$ "executes" both σ and τ in parallel. While we do not study parallel evaluation orders in this thesis, we conjecture that this parallel tensor can be used to build a model of QΛ where the reductions are not constrained to be left-then-right or right-then-left.

The idea behind $\otimes^p$ is to consider $\sigma \boxtimes \sigma'$, and then to synchronise the minimal events of $A \boxtimes A'$ in order to obtain $A \otimes A$, and the minimal events of $B \boxtimes B'$ in order to obtain $B \otimes B'$. On the right hand side of Fig. 5.15 we describe this third way to tensor σ_0 with itself which is neither $\sigma_0 \otimes^\ell \sigma_0$ nor $\sigma_0 \otimes^r \sigma_0$, as it runs both copies of σ_0 in parallel. We note that $\otimes^p$ is clearly not bifunctorial even in this simple example, as

$$\sigma_0 \otimes^\ell \sigma_0 := (\alpha_{\mathbf{1}} \otimes^p \sigma_0) \circ (\sigma_0 \otimes^p \alpha_{\mathbf{1} \multimap \mathbf{1}}) \neq \sigma_0 \otimes^p \sigma_0 \neq (\sigma_0 \otimes^c c_{\mathbf{1}}) \circ (\alpha_{\mathbf{1} \multimap \mathbf{1}} \otimes^p \sigma_0) =: \sigma_0 \otimes^r \sigma_0$$

Even though $\otimes^p$ is not bifunctorial, it still satisfies the bifunctoriality equations in some circumstances, like the following (recalling that τ_0 is thunkable):

$$(\tau_0 \otimes \tau_0) \odot (\sigma_0 \otimes^p \sigma_0) \cong (\tau_0 \odot \sigma_0) \otimes^p (\tau_0 \odot \sigma_0)$$

This is what we call the semi-bifunctoriality. We illustrate it in Fig. 5.16. From a term perspective, this means that we can identify the term $(f(); \lambda x.x \otimes f(); \lambda x.x)$ with the term $(f() \otimes f()); (\lambda x.x \otimes \lambda x.x)$.

To formally define this $\otimes^p$, we introduce some terminology: for $\sigma : A \nrightarrow B$ a negative quantum strategy, a configuration $x \in \mathcal{C}(S)$ is either

Empty if it is $\varnothing$.

Pre-Value if it is non-empty and it does not contain any $s \in x$ such that $\sigma(s) \in_\parallel \min(B)$ (*i.e.*, $\sigma(S) \in \varnothing \parallel \min(B)$).

Post-Value if it does contain a $s \in x$ such that $\sigma(s) \in_\parallel \min(B)$. This event is called the value-event of x.

If additionally $\tau : C \nrightarrow D$ is a negative quantum strategy, we say that $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(T)$ are synchronised if they are either both empty, both pre-value, or both post-value.

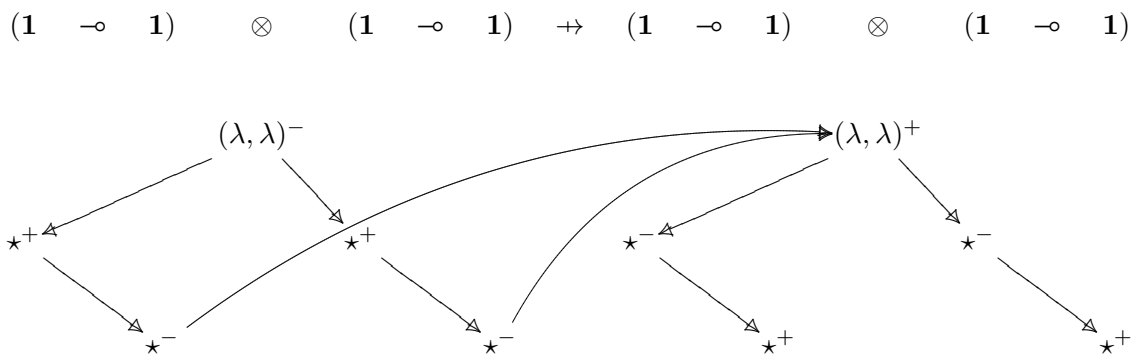


Figure 5.16: The strategy $(\tau_0 \otimes \tau_0) \odot (\sigma_0 \otimes^p \sigma_0) \cong (\tau_0 \odot \sigma_0) \otimes^p (\tau_0 \odot \sigma_0)$

Proposition 5.5.4 (Parallel Tensor of Strategies)**.** *For $\sigma : A \twoheadrightarrow B$ and $\sigma' : A' \twoheadrightarrow B'$ two negative quantum strategies, there exists a necessary unique (up to isomorphism) negative quantum strategy $\sigma \otimes^p \sigma'$ such that its configurations $z \in \mathcal{C}(S \otimes^p S')$ correspond to pairs written $x \otimes^p x'$ of synchronised configurations $x \in \mathcal{C}(S)$ and $x' \in \mathcal{C}(S')$, and its quantum valuation is:*

$$\mathcal{Q}_{\sigma \otimes^p \sigma'}(x \otimes x') = \mathcal{Q}_\sigma(x) \otimes \mathcal{Q}_{\sigma'}(x')$$

Proof. We first note than in a game $A \otimes A'$, every event e has a unique minimal event (m, m') smaller than it. Additionally, every event e of $A \otimes A'$ canonically corresponds to an event of A or A', or both if it is minimal. We generalise the notation (m, m') to events that are not minimal: for e an event greater than a minimal event $(m, m') \in |A \otimes A'|$, we write $e = (m, a')$ if e corresponds to $a' \neq m'$ in A', and we write $e = (a, m')$ if e corresponds to $a \neq m$ in A. This builds a bijection between $|A \otimes A'|$ and $\{(a, a') \in |A| \times |A'| \mid \text{either } a \text{ or } a' \text{ is minimal}\}$.

We define $\sigma \otimes^p \sigma'$. We say that an event $e \in |A \otimes A'|$ is pre-value or post-value whenever the configuration $[e]$ is pre-value or post-value.

- $|S \otimes^p S'| = \left\{ (s, s') \in |S| \times |S'| \;\middle|\; \begin{array}{lll} s \text{ pre-value}, & s' \in \min(S') & \text{or} \\ s \in \min(S), & s' \text{ pre-value} & \text{or} \\ s \text{ post-value}, & \sigma'(s') \in \min(B') & \text{or} \\ \sigma(s) \in \min(B), & s' \text{ post-value} & \end{array} \right\}$
- $\sigma \otimes^p \sigma' : (s, s') \in S \otimes^p S' \mapsto (\sigma(s), \sigma'(s'))$ either in $(A \otimes A')^\perp$ or in $(B \otimes B')$
- $(s, s') \leq_{S \otimes^p S'} (t, t') : s \leq_S t$ and $s' \leq_{S'} t'$
- $\mathrm{Con}_{S \otimes^p T} = \{Z \mid \pi_\ell(Z) \in \mathrm{Con}_S, \pi_r(Z) \in \mathrm{Con}_T\}$

- $\mathcal{Q}_{\sigma \otimes^p \sigma'}(z) = \mathcal{Q}_\sigma(\pi_\ell(z)) \otimes \mathcal{Q}_{\sigma'}(\pi_r(z))$

The fact that configurations of $\mathcal{C}(S \otimes^p S')$ correspond to synchronised pairs is a direct verification. □

This operation satisfies the property of semi-bifunctoriality, which we detail in the following proposition, and prove in Appendix B.1.

Proposition 5.5.5 (Semi-Bifunctoriality of the Parallel Tensor)**.** *For $\sigma : A \nrightarrow B$, $\sigma' : A' \nrightarrow B'$, $\tau : B \nrightarrow C$ and $\tau' : B' \nrightarrow C'$ two negative quantum strategies, we have the semi-bifunctoriality of $\otimes^p$, i.e., up to isomorphism*

$$(\tau \otimes^p \tau') \odot (\sigma \otimes^p \sigma') \cong (\tau \odot \sigma) \otimes^p (\tau' \odot \sigma')$$

whenever τ and τ' are thunkable, or σ and σ' are thunkable.

We recall that $\otimes^p$ defined here is **not** a bifunctor.

Proposition 5.5.6. *Quantum arenas and negative visible winning quantum strategies (up to isomorphism), form an SFC $(\mathbf{QA}, \mathbf{QA}_t, \mathbf{id}, \otimes, \mathbf{1})$, with for value category the category of quantum arenas and thunkable quantum strategies (up to isomorphism), and the Freyd inclusion being the identity functor. If we restrict to affine objects, $(\mathbf{QA}^a, \mathbf{QA}_t^a, \mathbf{id}, \otimes, \mathbf{1})$ is an affine SFC, meaning that $\mathbf{1}$ is a terminal object of $\mathbf{QA}_t^a$.*

Proof. First, we prove that $(\mathbf{QA}, \otimes, \mathbf{1})$ is an SPC. We define $A \otimes \sigma$ as $\mathrm{cc}_A \otimes^p \sigma$, and similarly for $\sigma \otimes A$. The braiding, associator, and unitors are easily built from the braiding, associator and unitor of $(\mathbf{QCG}, \boxtimes, \varnothing)$, and the coherence diagrams deduced from the ones of **QCG**. The semi-bifunctoriality ensures that thunkable maps are in the centre of the SPC.

As for the affineness, we take a thunkable strategy $\sigma : A \nrightarrow \mathbf{1}$. Minimal events of S are negative. Using thunkability, runner-up events of S are positive and necessarily sent to the unique event of $\mathbf{1}$ per σ. Using courtesy, second runner-up events of S, *i.e.*, events that have only two events lower, must be negative. However, there is no negative event accessible after the runner-up events. It follows that configurations of S cannot have more than two events. Using thunkability again, it follows that σ is isomorphic to $\mathbf{destr}_A : D_A \to A^\perp \parallel \mathbf{1}$:

- $|D_A| = \{a^- \mid a \in \min(A)\} \sqcup \{\star_a^+ \mid a \in \min(A)\}$
- $\mathcal{C}(D_A) = \{\varnothing\} \sqcup \{\{a\} \mid a \in \min(A)\} \sqcup \{\{a, \star_a\} \mid a \in \min(A)\}$
- $\mathcal{Q}_{\mathbf{destr}_A}(x) = \mathbf{Tr}_{\mathcal{H}_A(x_A)}$ □

When A is affine, this strategy is winning.

5.5.3 The Freyd Closure

We now want to define the Freyd closure. We first define the arena $A \multimap B$. As a first approximation, a strategy on $A \multimap B$ should be the same as a strategy from A to B, *i.e.*, a strategy on $A^\perp \parr B$. However, since we only consider negative strategies, there is an implicit causal link from the minimal events of A to the minimal events of B. The decomposition $A = \bigoplus_{i\in I} \downarrow_{(a_i:H_i)} A_i$ can be dualised into $A^\perp = \bigoplus_{i\in I} \uparrow_{(a_i:H_i^*)} A_i^\perp$, where $\uparrow$ represents the negative shift, *i.e.*, adding a minimal negative event before than every other event. This idea of $A^\perp \parr B$ with a causal link from the minimal events of A to the minimal events of B can be formalised as:

$$L := \bigoplus_{i\in I} \uparrow_{(a_i:H_i^*)} (A_i^\perp \parr B)$$

Unfortunately, L is not an arena, as its minimal events are negative. The solution is to add a minimal event λ^+. As explained in multiple prior examples, this minimal event λ^+ can be understood as "the function is defined and ready to be called" (which is an observable event in call-by-value languages), the event a_i^- can be understood as "the user calls the function on input a_i", the following events of $A_i^\perp$ correspond to functions given by the user that can be called by the program, and the events of B describe the outputs of the function.

Definition 5.5.7. *For A and B two quantum arenas, with $A = \bigoplus_{i\in I} \downarrow_{(a_i:H_i)} A_i$. We define the quantum arena $A \multimap B$ as in Fig. 5.17. Formally:*

$$A \multimap B := \downarrow_{(\lambda:\mathbf{1})} \bigoplus_{i\in I} \uparrow_{(a_i:H_i^*)} (A_i^\perp \parr B) \text{ with } \begin{array}{lcl} \kappa_{A\multimap B}(\varnothing) & = & 0 \\ \kappa_{A\multimap B}(\{\lambda\}) & = & 1 \\ \kappa_{A\multimap B}(\{\lambda, a_i\} \sqcup X) & = & \kappa_{A^\perp \parr B}(\{a_i\} \sqcup X) \end{array}$$

We then define $A \multimap\!\!\bullet\, B$ as $A \multimap B$ except that $\kappa_{A\multimap\bullet B}(\{\lambda\}) = 0$ instead of 1. The arena $A \multimap\!\!\bullet\, B$ is affine, while $A \multimap B$ is not.

Lemma 5.5.8. *There is a union and intersection preserving bijection between the non-empty configurations of $x \in \mathcal{C}(A \multimap B)$ and the pairs of configurations $(x_A, x_B) \in \mathcal{C}(A) \times \mathcal{C}(B)$ such that $x_A = \varnothing \Rightarrow x_B = \varnothing$. We write $x = x_A \multimap x_B$.*

Proof. For $x_A \in \mathcal{C}(A)$ non-empty and $x_B \in \mathcal{C}(B)$, we have a unique minimal event $a_i \in x_A$. We take $x = \{\lambda\} \uplus \{a_i\} \uplus (x_A \backslash \{a_i\} \parallel x_B) \in \mathcal{C}(A \multimap B)$. This forms a union and intersection preserving partial injection, which we extend into a bijection with $(\varnothing, \varnothing) \mapsto \{\lambda\}$. □

The difference between $A \multimap B$ and $A \multimap\!\!\bullet\, B$ is the payoff of $\{\lambda\}$. For the former, the payoff is 1, hence the configuration is "winning", so Opponent will not want to stop there and will call the function whenever possible, hence strict linearity will be respected. For the

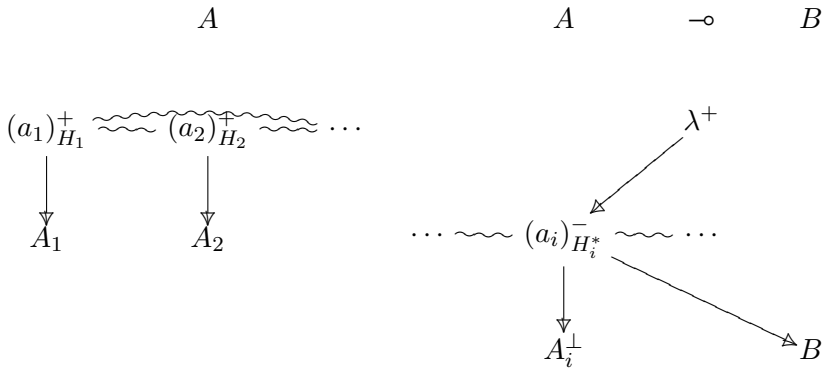


Figure 5.17: The linear arrow ⊸ of two arenas

latter, the payoff is 0, hence the configuration is a "tie", so both Player and Opponent might choose to stop here (and the function is never called), or continue and call the function, hence describing an affine behaviour. When interpreting LQΛ, we will use $A \multimap B$ to represent functions, while we will use $A \multimap\!\!\bullet\, B$ instead in AQΛ. We note that in [CdVW19], we did not use any payoff function to interpret AQΛ, as we did not build a fully abstract model.

The core property of ⊸ is the currying adjunction $\mathbf{QA}(A \otimes B, C) \cong \mathbf{QA}_t(A, B \multimap C)$. We describe in Figs. 5.18 and 5.19 the games for $(A \otimes B) \twoheadrightarrow C$ and $A \twoheadrightarrow (B \multimap C)$, and give an example of currying in Fig. 5.20. In this example, the top strategy represents

$$f_0 : \mathbf{qubit} \multimap \mathbf{1}, f_1 : \mathbf{1} \multimap \mathbf{qubit} \vdash_{\mathbb{L}} f_0\ (f_1\ ()) : \mathbf{1}$$

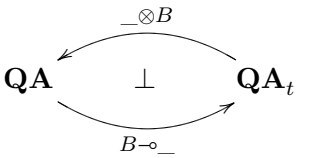
and the bottom strategy represents

$$f_0 : \mathbf{qubit} \multimap \mathbf{1} \vdash_{\mathbb{L}} \lambda f_1^{\mathbf{1} \multimap \mathbf{qubit}}.f_0\ (f_1\ ()) : \mathbf{1}$$

The main difference between the two is that the event $(\lambda_0, \lambda_1)^-$ is split into three events λ_0^-, λ^+ and λ_1^-.

Proposition 5.5.9. $(\mathbf{QA}, \mathbf{QA}_t, \mathbf{id}, \otimes, \multimap, \mathbf{1})$ *is a CFC. Similarly,* $(\mathbf{QA}^a, \mathbf{QA}_t^a, \mathbf{id}, \otimes, \multimap\!\!\bullet, \mathbf{1})$ *is an affine CFC.*

Proof. We want to prove that there is an adjunction



$\mathbf{QA}(A \otimes B, C) \cong \mathbf{QA}_t(A, B \multimap C)$

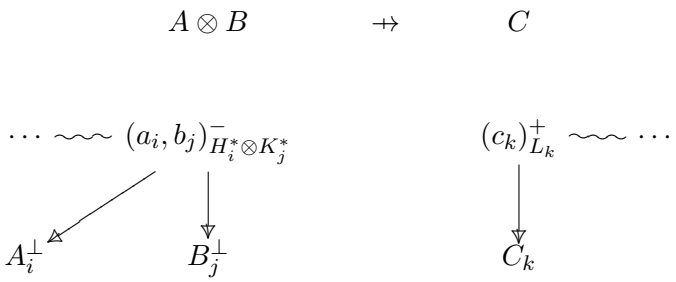


Figure 5.18: Game for $(A \otimes B) \twoheadrightarrow C$.

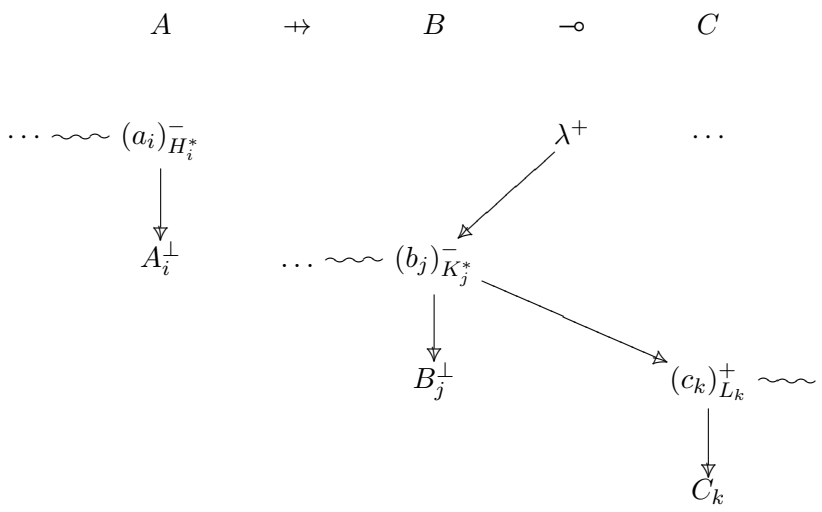
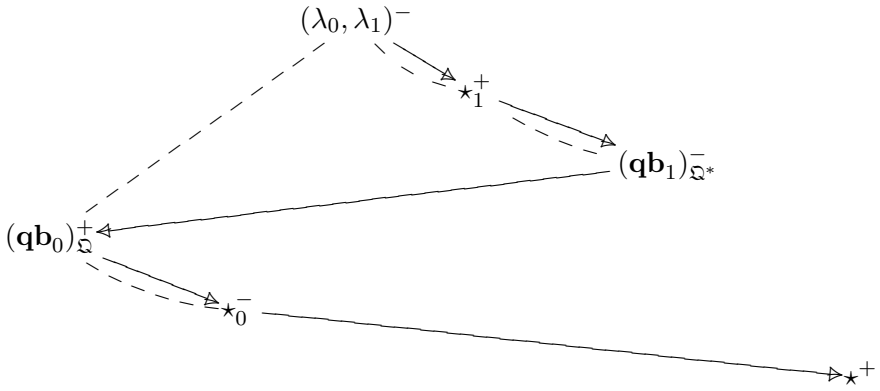
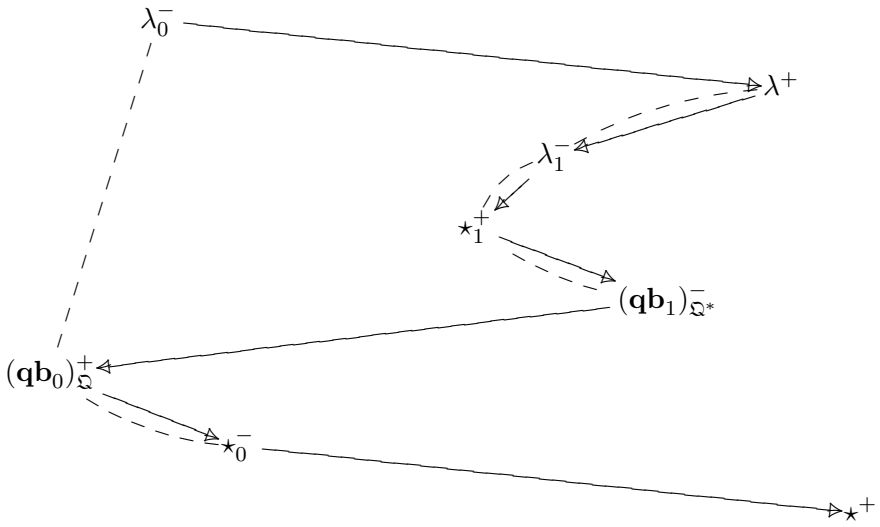


Figure 5.19: Game for $A \twoheadrightarrow (B \multimap C)$.

$$(\mathbf{qubit} \multimap \mathbf{1}) \otimes (\mathbf{1} \multimap \mathbf{qubit}) \vdash_{\mathbb{L}} \mathbf{1}$$



$$\mathbf{qubit} \multimap \mathbf{1} \vdash_{\mathbb{L}} (\mathbf{1} \multimap \mathbf{qubit}) \multimap \mathbf{1}$$



Quantum valuation for both strategies:

$$\mathcal{Q}^{-,+}(x) \begin{cases} \mathbf{id_1} & \text{if } x \subset [(\mathbf{qb}_r)^-_{\mathfrak{Q}^*}] \\ \mathbf{Tr}_{\mathfrak{Q}} & \text{if } x = [(\mathbf{qb}_r)^-_{\mathfrak{Q}^*}] \\ \mathbf{id}_{\mathfrak{Q}} & \text{if } x \supset [(\mathbf{qb}_r)^-_{\mathfrak{Q}^*}] \end{cases}$$

Figure 5.20: Example of currying.

Ignoring quantum valuations and payoff, we remark that for $\sigma \in \mathbf{QA}(A \otimes B, C)$, S starts by events of the form (a_i, b_j) with $a_i \in \min(A)$ and $b_j \in \min(B)$. It follows that:

$$\mathbf{QA}(A \otimes B, C) \cong \prod_{i \in I} \prod_{j \in J} \mathbf{QCG}(A_i \boxtimes B_j, C)$$

Similarly, for $\tau \in \mathbf{QA}_t(A, B \multimap C)$, T starts by sequences of events $a_i \twoheadrightarrow \lambda \twoheadrightarrow b_j$ with $a_i \in \min(A)$ and $b_j \in \min(B)$. It follows that:

$$\mathbf{QA}_t(A, B \multimap C) \cong \prod_{i \in I} \prod_{j \in J} \mathbf{QCG}(A_i, B_j^{\perp} \parr C)$$

Since $\mathbf{CG}$ is $\star$-autonomous, we obtain that ignoring quantum valuations and payoff:

$$\mathbf{QA}(A \otimes B, C) \cong \mathbf{QA}_t(A, B \multimap C)$$

We immediately note that this bijection sends winning strategies to winning strategies. In fact, ignoring quantum valuation it is a bijection between $\mathbf{QA}(A \otimes B, C)$ and $\mathbf{QA}_t(A, B \multimap C)$. We write $\Lambda(-)$ for this bijection between strategies. By construction, we have a bijection λ between the non-empty configurations of σ and the configurations of cardinal at least three of $\Lambda(\sigma)$, such that

$$\sigma(x) = (x_A \otimes x_B) \parallel x_C \iff \Lambda(\sigma)(\lambda(x)) = x_A \parallel (x_B \multimap x_C)$$

The normalisation condition ensures that the quantum valuation on the empty configuration is $\mathbf{id_1}$. Similarly, obliviousness ensures that the quantum valuation on singleton configurations is the trace, and thunkability ensures that there is still a unique possibility for the quantum valuation of configuration of size two. So because of normalisation, obliviousness and thunkability, the quantum valuation of a strategy of $\mathbf{QA}_t(A, B \multimap C)$ is uniquely determined by its value on configurations of cardinal at least three. By taking

$$\mathcal{Q}_\sigma(x) = \mathcal{Q}_{\Lambda(\sigma)}(\lambda(x))$$

we obtain, without ignoring anything this time, the bijection

$$\mathbf{QA}(A \otimes B, C) \cong \mathbf{QA}_t(A, B \multimap C)$$

To prove that this bijection defines an adjunction, It suffices check the two following equations with $\mathrm{eval}_{B,C} := \Lambda^{-1}(\mathbf{id}_{B \multimap C})$:

$$\begin{array}{rrcl} \forall \sigma \in \mathbf{QA}(A \otimes B, C), & \mathrm{eval}_{C,B} \odot (\Gamma(\sigma) \otimes B) & = & \sigma \\ \forall \tau \in \mathbf{QA}_t(A, B \multimap C), & \Lambda(\mathrm{eval}_{B,C} \odot (\tau \otimes B)) & = & \tau \end{array}$$

If we use $\multimap\!\!\bullet$ instead of $\multimap$ and restrict ourselves to affine arenas, the proof still works out. □

5.5.4 The Distributive CFC

The bifunctor $\oplus$ of **QCG** is also a bifunctor on **QA** and $\mathbf{QA}_t$. We write $\mathbf{0}$ for the empty quantum arena. The difference between the payoff games $\varnothing$ and $\mathbf{0}$ is that the payoffs for the empty configuration are respectively 0 and -1. We write $\bot$ for the unique strategy of $\mathbf{QA}(\mathbf{1},\mathbf{0})$. We note that $\mathbf{0}$ and $\mathbf{1}$ are not isomorphic.

Proposition 5.5.10. $(\mathbf{QA}, \mathbf{QA}_t, \mathbf{id}, \otimes, \multimap, \mathbf{1}, \oplus, \mathbf{0})$ *is a distributive CFC. Similarly,* $(\mathbf{QA}^a, \mathbf{QA}_t^a, \mathbf{id}, \otimes, \multimap\!\!\bullet, \oplus, \mathbf{0})$ *is an affine distributive CFC.*

Proof. The injection from A to $A \oplus B$ simply has $\mathcal{C}_A$ for esp, with the same quantum valuation as ϖ_A. Similarly, the injection from B to $A \oplus B$ simply has $\mathcal{C}_B$ for esp, with the same quantum valuation as ϖ_B.

The copairing of σ and τ simply has $S \oplus T$ for esp, with quantum valuation induced from $\mathcal{Q}_\sigma$ and $\mathcal{Q}_\tau$.

Proposition 1.2.16 ensures that **QA** is distributive. To prove that $\mathbf{QA}_t$ is distributive too, we just need to remark that $\mathbf{0}_{\mathbf{0}\otimes B}$ and $\mathrm{dis}_{A_\ell, A_r, B}$ are thunkable, and that their inverses are thunkable too. □

It follows that quantum arenas and (negative winning visible quantum) strategies form a non-trivial distributive CFC with a bottom, hence a sound and adequate denotational model of LΛ. If we restrict ourselves to affine objects, this category becomes affine, hence a sound and adequate denotational model for AΛ. While this was not explicit in the proposition, $\oplus$ also defines arbitrary coproducts, and the $\otimes$ is distributive over this infinite coproduct.

Chapter 6

Game Semantics for the Linear Quantum λ-calculus

6.1 Games Model for LQΛ

6.1.1 Semantics for Terms

We use the semantics described in Section 1.4.2 for most of LQΛ, and complete it as described in Table 6.1. We refer in this table to morphisms $\mathbf{new}^{\mathbf{QA}}$, $\mathbf{meas}^{\mathbf{QA}}$ and $U^{\mathbf{QA}}$ which we define in Figs. 6.1 to 6.3.

The semantics for **qubit** is simply the quantum arena $\downarrow_{(\mathbf{qb}:\mathbb{C}^2)} \varnothing$. In contrast to a regular **bit** which has two events, one for true and one for false, a qubit is represented with a single event, with its set of possible values abstracted as a quantum space annotation. The semantics for A^ℓ is computed through a least fixpoint, for the following partial order.

Definition 6.1.1. *For any two quantum arenas A and B , we say that A is a substructure of B if they are substructures as event structures, and the inclusion function preserves polarity, payoff, and quantum space annotations. For any quantum arenas $A_0, A_1, \dots$ with A_i a substructure of A_{i+1} for every $i \in \mathbb{N}$, we define $\bigcup_{i\in\mathbb{N}} A_i$ as the quantum arenas with for events $\bigcup_{i\in\mathbb{N}} |A_i|$ and all the structure induced by the A_i.*

For any quantum arena A , we write $F_A(X) := \mathbf{1} \oplus (A \otimes X)$, and remark that $F_A^n(\mathbf{0})$ is a substructure of $F_A^{n+1}(\mathbf{0})$ for every $n \in \mathbb{N}$.

Definition 6.1.2. *For any quantum arena A, we define*

$$A^\ell := \bigcup_{n\in\mathbb{N}} F_A^n(\mathbf{0})$$

We write $[\ ] \in \mathcal{C}(A^\ell)$ for the configuration $\{\star\} \oplus \varnothing$ and $[x_1; \dots; x_n] \in \mathcal{C}(A^\ell)$ for the configuration $\varnothing \oplus (x_1 \otimes [x_2; \dots; x_n])$.

Quantum primitives:

$$[\![\mathbf{qubit}]\!] := \downarrow_{(\mathbf{qb}:\mathfrak{Q})} \varnothing$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{bit}\end{array}}{\Gamma \vdash \mathbf{new}\ t : \mathbf{qubit}} \right]\!\!\right] := \mathbf{new}^{\mathbf{QA}} \odot \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{bit}\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}\end{array}}{\Gamma \vdash \mathbf{meas}\ t : \mathbf{bit}} \right]\!\!\right] := \mathbf{meas}^{\mathbf{QA}} \odot \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}^{\otimes n}\end{array}}{\Gamma \vdash \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}} \right]\!\!\right] := U^{\mathbf{QA}} \odot \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}\end{array} \right]\!\!\right]$$

with $\mathbf{new}^{\mathbf{QA}}$, $\mathbf{meas}^{\mathbf{QA}}$ and $U^{\mathbf{QA}}$ defined in Figs. 6.1 to 6.3

Lists:

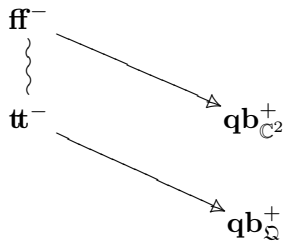
$$[\![A^\ell]\!] := [\![A]\!]^\ell = \mathbf{1} \oplus \left([\![A]\!] \otimes [\![A]\!]^\ell \right)$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : A^\ell\end{array}}{\Gamma \vdash \mathbf{unfold}\ t : \mathbf{1} \oplus (A \otimes A^\ell)} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : A^\ell\end{array} \right]\!\!\right]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{1} \oplus (A \otimes A^\ell)\end{array}}{\Gamma \vdash \mathbf{fold}\ t : A^\ell} \right]\!\!\right] := \left[\!\!\left[\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{1} \oplus (A \otimes A^\ell)\end{array} \right]\!\!\right]$$

Table 6.1: Game Semantics of LQΛ Typing Derivations

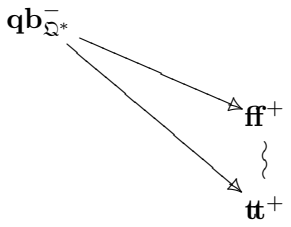
$\mathbf{new}^{\mathbf{QA}} : \quad \mathbf{bit} \quad \nrightarrow \quad \mathbf{qubit}$



$$\begin{array}{rcl} \mathcal{Q}_{\mathbf{new}^{\mathbf{QA}}}(\varnothing) & = & \mathbf{id}_1^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{new}^{\mathbf{QA}}}(\{\mathbf{ff}\}) & = & \mathbf{id}_1^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{new}^{\mathbf{QA}}}(\{\mathbf{tt}\}) & = & \mathbf{id}_1^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{new}^{\mathbf{QA}}}(\{\mathbf{ff}, \mathbf{qb}\}) & = & \mathbf{new}_{\mathbf{ff}}^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{new}^{\mathbf{QA}}}(\{\mathbf{tt}, \mathbf{qb}\}) & = & \mathbf{new}_{\mathbf{tt}}^{\mathbf{CPM}} \end{array}$$

Figure 6.1: The Strategy $\mathbf{new}^{\mathbf{QA}} : \mathbf{bit} \nrightarrow \mathbf{qubit}$

$\mathbf{meas}^{\mathbf{QA}} : \quad \mathbf{qubit} \quad \nrightarrow \quad \mathbf{bit}$



$$\begin{array}{rcl} \mathcal{Q}_{\mathbf{meas}^{\mathbf{QA}}}(\varnothing) & = & \mathbf{id}_1^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{meas}^{\mathbf{QA}}}(\{\mathbf{qb}\}) & = & \mathbf{Tr}_{\mathfrak{Q}} \\ \mathcal{Q}_{\mathbf{meas}^{\mathbf{QA}}}(\{\mathbf{qb}, \mathbf{ff}\}) & = & \mathbf{meas}_{\mathbf{ff}}^{\mathbf{CPM}} \\ \mathcal{Q}_{\mathbf{meas}^{\mathbf{QA}}}(\{\mathbf{qb}, \mathbf{tt}\}) & = & \mathbf{meas}_{\mathbf{tt}}^{\mathbf{CPM}} \end{array}$$

Figure 6.2: The Strategy $\mathbf{meas}^{\mathbf{QA}} : \mathbf{qubit} \nrightarrow \mathbf{bit}$

$U^{\mathbf{QA}} : \quad \mathbf{qubit}^{\otimes n} \quad \nrightarrow \quad \mathbf{qubit}^{\otimes n}$

$$\begin{array}{rcl} \mathcal{Q}_{U^{\mathbf{QA}}}(\varnothing) & = & \mathbf{id}_1^{\mathbf{CPM}} \\ \mathcal{Q}_{U^{\mathbf{QA}}}(\{(\mathbf{qb}^n)\}) & = & \mathbf{Tr}_{\mathfrak{Q}^{\otimes n}} \\ \mathcal{Q}_{U^{\mathbf{QA}}}(\{(\mathbf{qb}^n), (\mathbf{qb}^n)\}) & = & (\!|U|\!) \end{array}$$

Figure 6.3: The Strategy $U^{\mathbf{QA}} : \mathbf{qubit}^{\otimes n} \nrightarrow \mathbf{qubit}^{\otimes n}$ for U unitary

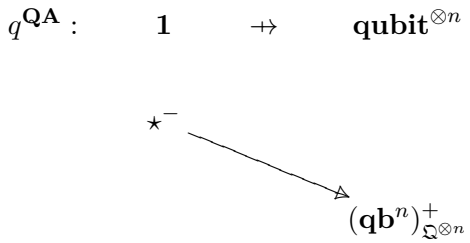


Figure 6.4: The Strategy $q^{\mathbf{QA}} : \mathbf{1} \nrightarrow \mathbf{qubit}^{\otimes n}$ for q quantum state

Lemma 6.1.3. *For any quantum arena A, we have $A^{\ell} \cong \bigoplus_{n\in\mathbb{N}} A^{\otimes n}$.*

Theorem 6.1.4. *If $\Gamma \vdash t : A$ has two typing derivations T and T', then we have $[\![T]\!] = [\![T']\!]$. As it is independent from the typing derivation, we write it $[\![t]\!]^{\Gamma\vdash A}$.*

The proof of this theorem is the same as the proof for LΛ earlier in Theorem 1.4.6, extended without difficulties to LQΛ.

6.1.2 Semantics for Quantum Closures

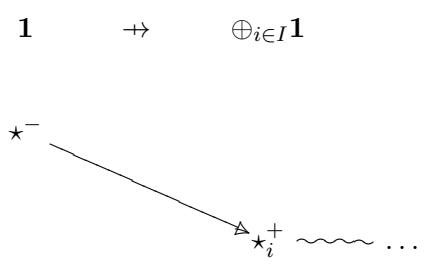
We now extend the semantics to quantum closures. However, **QA** has no "weighted sum" that satisfies the axioms of a convex cone. Indeed, our game model remembers branching points so when representing the closure $\frac{1}{2}c + \frac{1}{2}c$ we will obtain a different strategy than when representing c. While this is a problem for the adequacy and later the full abstraction, this only makes the model more precise than required, so multiple strong properties still hold. So we will simulate the probabilistic sums through the coproduct $\oplus$, and postpone to the next section the amendments we make to our model to obtain full abstraction.

Definition 6.1.5. *For $p_i \in [0,1]$ for all $i \in I$, with $\sum_{i\in I} p_i \leq 1$, we define the quantum strategy* $\mathrm{choice}_{\{p_i \mid i\in I\}} : \mathbf{1} \to \oplus_{i\in I}\mathbf{1}$ *as described in Fig. 6.5. For $\sigma_i : A \nrightarrow B$ $(i \in I)$ negative quantum strategies, we define $\boxplus_{i\in I} p_i \cdot \sigma_i : A \nrightarrow B$ using the copairing as follows:*

$$\boxplus_{i\in I} p_i \cdot \sigma_i := [\sigma_i \mid i \in I] \odot \Big(\mathrm{choice}_{\{p_i \mid i\in I\}} \otimes A\Big)$$

In Fig. 6.6, we provide an example of binary $\boxplus$, where σ is the representation of the term $\lambda().\mathbf{ff}$, τ represents the term $\lambda().\mathbf{tt}$ and ν the quantum closure $\frac{1}{3}[\varnothing, \varnothing, \lambda().\mathbf{ff}] + \frac{2}{3}[\varnothing, \varnothing, \lambda().\mathbf{tt}]$.

Lemma 6.1.6. *The operation $\boxplus$ is, up to isomorphism, associative, commutative, left-*



Quantum valuation:

$$\mathcal{Q}(\varnothing) = \mathcal{Q}(\{\star^-\}) = \mathbf{id_1}$$
$$\forall i \in I, \mathcal{Q}(\{\star^-, \star_i^+\}) = p_i \cdot \mathbf{id_1}$$

Figure 6.5: Strategy choice$_{\{p_i \mid i\in I\}} : \mathbf{1} \to \oplus_{i\in I}\mathbf{1}$

linear and semi-right-linear, i.e.,

$$\begin{array}{rcll}
\boxplus_{i\in I} p_i \cdot \boxplus_{j\in J} q_j \cdot \sigma_{i,j} & \cong & \boxplus_{(i,j)\in I\times J}(p_i q_j)\cdot\sigma_{i,j} & \\
p\sigma \boxplus q\tau & \cong & q\tau \boxplus p\sigma & \\
\tau \odot \left(\boxplus_{i\in I} p_i \cdot \sigma_i\right) & \cong & \boxplus_{i\in I} p_i \cdot (\tau \odot \sigma_i) & \\
\left(\boxplus_{i\in I} p_i \cdot \sigma_i\right) \odot \tau & \cong & \boxplus_{i\in I} p_i \cdot (\sigma_i \odot \tau) & \text{whenever } \tau \text{ thunkable}
\end{array}$$

We use the notation $\boxplus$ rather than $\sum$ as it is **not** idempotent:

$$p \cdot \sigma \boxplus q \cdot \sigma \ncong (p+q)\cdot\sigma$$

We can now define the semantics of quantum closures.

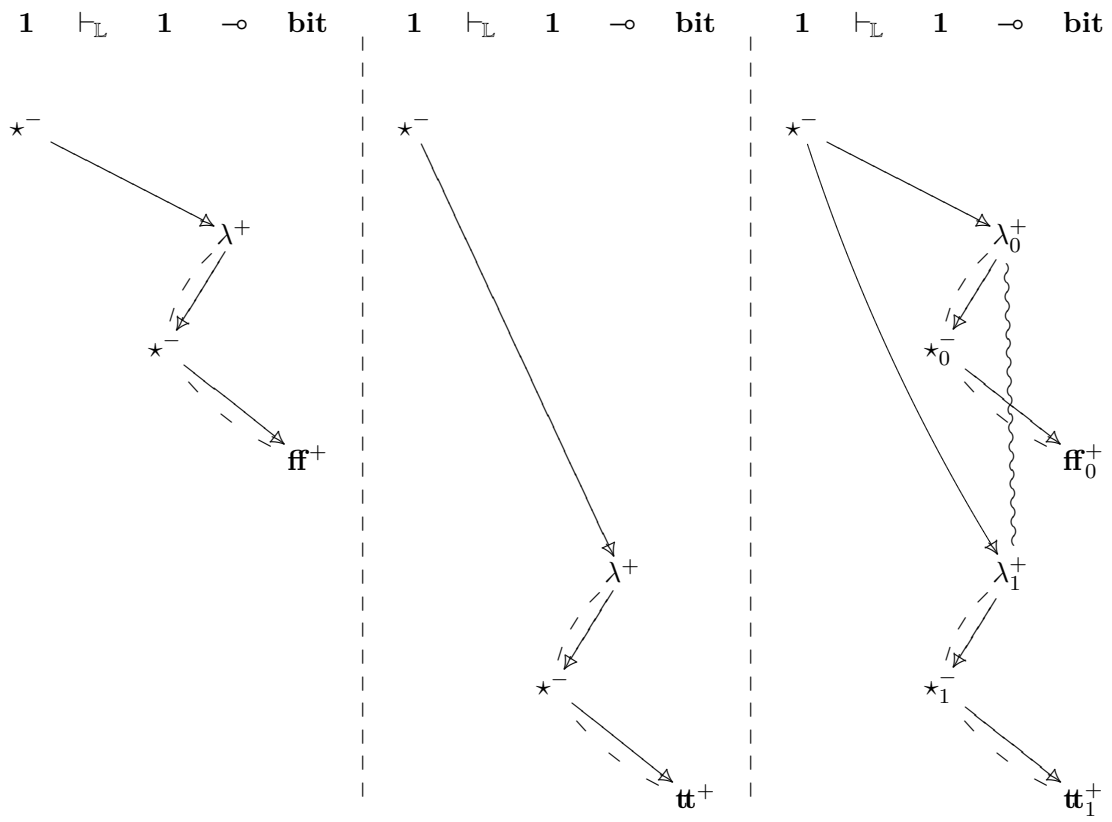
Definition 6.1.7. *If* $\vdash [q, \ell, t] : A$*, we define* $[\![[q, \ell, t]]\!]^{\vdash A}$ *as follows:*

- *we know we have* $\Delta \vdash t : A$ *with* $\Delta = x_1 : \mathbf{qubit}, \ldots, x_n : \mathbf{qubit}$
- *we define* $q^{\mathbf{QA}}$ *as in Fig. 6.4*
- $[\![[q, \ell, t]]\!] := [\![t]\!] \odot q^{\mathbf{QA}}$

and we then define $[\![\sum_i p_i[q_i, \ell_i, t_i]]\!]$ *as* $\boxplus_i p_i [\![[q_i, \ell_i, t_i]]\!]$.

6.1.3 Soundness and Adequacy

This missing idempotence property is very problematic for the full abstraction, as it means that our semantics remembers the point of branching and is able to distinguish the quantum



Quantum valuation of σ (left), τ (middle) and $\nu = 1/3 \cdot \sigma \boxplus 2/3 \cdot \tau$ (right):

$$\forall x \in \mathcal{C}(S), \mathcal{Q}_\sigma(x) = \mathbf{id_1}$$
$$\forall x \in \mathcal{C}(T), \mathcal{Q}_\tau(x) = \mathbf{id_1}$$
$$\mathcal{Q}_\nu(x) = \begin{cases} \mathbf{id_1} & \text{whenever } x \subseteq \{\star^-\} \\ \frac{1}{3}\mathbf{id_1} & \text{whenever } \lambda_0^+ \in x \\ \frac{2}{3}\mathbf{id_1} & \text{whenever } \lambda_1^+ \in x \end{cases}$$

Figure 6.6: Example of binary $\boxplus$

closures $\frac{1}{2}c + \frac{1}{2}c$ and c, while our operational semantics considers both to be the same. The next section focuses on how to handle this problem, but we first state here the properties that we do have despite it, *i.e.*, soundness and adequacy.

Lemma 6.1.8 (Value Substitution)**.** *For every term $\Gamma, x : A \vdash t : B$ and every value $\Delta \vdash v : A$, we have up to isomorphism:*

$$[\![t]\!] \circ ([\![\Gamma]\!] \otimes [\![v]\!]) = [\![t\{x \leftarrow v\}]\!]$$

The proof is the same as for LΛ in Section 1.4.3.

Lemma 6.1.9 (Context Factorisation)**.** *For every term $\Gamma \vdash s : A$ and $\Gamma, \Delta \vdash E[s] : B$, with $E[-]$ an evaluation context, we have a morphism $[\![E]\!] \in \mathbf{QRel}([\![A]\!] \otimes [\![\Delta]\!], [\![B]\!])$ such that*

$$[\![E[s]]\!] = [\![E]\!] \circ ([\![s]\!] \otimes [\![\Delta]\!])$$

The proof is then the same as for LΛ in Section 1.4.3. The following lemma is the invariance lemma restricted to terms. We recall that LQΛ has two reduction systems: one on terms that support every feature of the language but quantum ones, and one on closures that include all the term reductions and support additional reductions for quantum operations.

Lemma 6.1.10 (Term Invariance)**.** *For every pair of terms $\Gamma \vdash t : A$ and $\Gamma \vdash s : A$, we have up to isomorphism:*

$$t \to s \implies [\![t]\!] = [\![s]\!]$$

Proof. Using the same proof as for LΛ, for every pair of terms $\Gamma \vdash t : A$ and $\Gamma \vdash s : A$ that do not use **fold** or **unfold**, we have:

$$t \to s \implies [\![t]\!] = [\![s]\!]$$

This is proven by induction on typing derivations. Since the rules for **fold** and **unfold** are trivial, the proof extends without problems to terms containing **fold** and **unfold**. □

In presence of branching, the game semantics will record the branching while the operational semantics will merge identical branches; this breaks the invariance lemma for closures:

$$\begin{array}{ccccc} [\varnothing, \varnothing, \textbf{if } \mathrm{Coin}_{1/2}() \textbf{ then } () \textbf{ else } ()] & \to^* & [\varnothing, \varnothing, ()] & & \\ \left[\!\!\left[[\varnothing, \varnothing, \textbf{if } \mathrm{Coin}_{1/2}() \textbf{ then } () \textbf{ else } ()]\right]\!\!\right] & = & \frac{1}{2}\mathbf{id}_{\mathbf{1}}^{\mathbf{QA}} \boxplus \frac{1}{2}\mathbf{id}_{\mathbf{1}}^{\mathbf{QA}} & \neq & \mathbf{id}_{\mathbf{1}}^{\mathbf{QA}} \\ [\![[\varnothing, \varnothing, ()]]\!] & = & \mathbf{id}_{\mathbf{1}}^{\mathbf{QA}} & & \end{array}$$

We now obtain an adequacy result slightly different from the one of Section 1.4.3, as a strategy of $\mathbf{QA}(\mathbf{1}, \mathbf{1})$ is not characterised by its probability of convergence. We have to recover the probability of convergence by summing over all the different branches that have been accumulated in σ through the computation.

Proposition 6.1.11 (Soundness and Adequacy)**.** *For every term* $\vdash t : \mathbf{1}$*, we have up to isomorphism:*

$$\mathbb{P}(\llbracket t \rrbracket \Downarrow) = p \iff \mathbb{P}(t \Downarrow) = p$$

where $\mathbb{P}(\sigma \Downarrow)$ *(for* $\sigma : \mathbf{1} \nrightarrow \mathbf{1}$*) is the probability of a positive event being played by* σ*, in other words:*

$$\sum_{\substack{x \in \mathcal{C}(S) \\ |x|=2}} \mathcal{Q}_\sigma(x) = \mathbb{P}(\sigma \Downarrow) \cdot \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$$

This proposition is not obvious to prove as the invariance lemma does not hold on closures. For conciseness, we choose to not provide a proof of this proposition, however it easily arises as a consequence of Theorem 6.2.11 below.

6.1.4 Examples

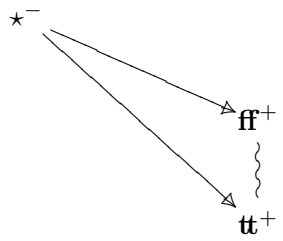
We go through the same examples as in Sections 2.3.2 and 3.2.2 in Figs. 6.7 to 6.11. The strategy for the biased coin, the Bell state and the Bell measurement are pretty straightforward, and nearly identical to their relational semantics.

Indeed, when no higher order is present the game semantics is the same as the relational semantics, but with additional layers of complexities:

- The minimal events of a negative strategy are negative, and because of the receptivity condition on strategies, they are uniquely determined by the game.
- In the absence of higher order, configurations have cardinal at most two, and configurations of size two have one negative event and one positive event.
- The quantum valuation for the empty configuration is $\mathbf{id_1}$ since our strategies are normalised. The quantum valuation for every singleton configuration is uniquely determined by the obliviousness condition on strategies.
- Every configuration of size two of the strategy of a term lives over a point of the web of the relational semantics. For example in the case of the Bell states strategy, the configuration $\{(\mathbf{ff}, \mathbf{ff}), (\mathbf{qb}, \mathbf{qb})\}$ lives over the point of the web $((\mathbf{ff}, \mathbf{ff}), (\mathbf{qb}, \mathbf{qb}))$.
- There is a correspondence between the quantum valuations on configurations of size two, and the quantum annotation on the points of the web of the relational semantics. For example in the case of the Bell states, the configuration $\{(\mathbf{ff}, \mathbf{ff}), (\mathbf{qb}, \mathbf{qb})\}$ has for valuation $\langle\!| \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle |\!\rangle$, which is the same as the quantum annotation on $((\mathbf{ff}, \mathbf{ff}), (\mathbf{qb}, \mathbf{qb}))$ given by the relational semantics of the Bell states.

There are some subtleties in this correspondence. As noted before, the game semantics has a different semantics for the quantum closures $\frac{1}{2}c + \frac{1}{2}c$ and c, while the relational semantics does not. Technically, this is because multiple configurations of size two can correspond to

$$\varnothing \qquad \vdash_{\mathbb{L}} \qquad \mathbf{bit}$$



Quantum valuation:

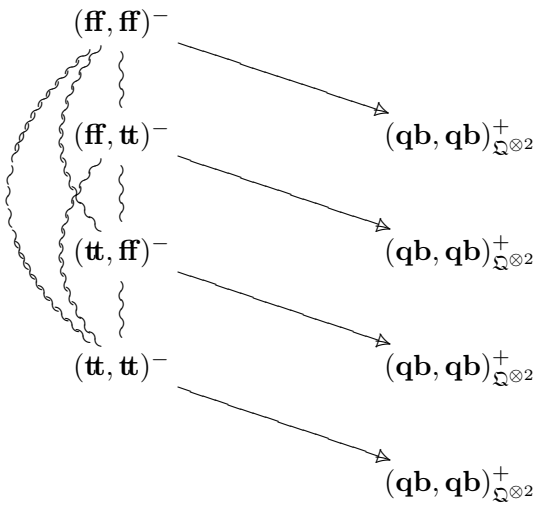
$$\begin{array}{rcl}
\mathcal{Q}^{-,+}_{[\![\mathrm{Coin}_p()]\!]}(\varnothing) & = & \mathbf{id_1} \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Coin}_p()]\!]}(\{\star\}) & = & \mathbf{id_1} \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Coin}_p()]\!]}(\{\star, \mathbf{ff}\}) & = & (1-p)\cdot \mathbf{id_1} \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Coin}_p()]\!]}(\{\star, \mathbf{tt}\}) & = & p \cdot \mathbf{id_1}
\end{array}$$

Term represented:

$$\vdash_{\mathbb{L}} \mathrm{Coin}_p() : \mathbf{bit}$$
$$\mathrm{Coin}_p() := \mathbf{meas} \begin{pmatrix} \sqrt{1-p} & \sqrt{p} \\ \sqrt{p} & -\sqrt{1-p} \end{pmatrix}$$

Figure 6.7: Strategy for the Biased Coin $[\![\mathrm{Coin}_p()]\!]$

$$\mathbf{bit}, \mathbf{bit} \qquad \vdash_{\mathbb{L}} \qquad \mathbf{qubit} \otimes \mathbf{qubit}$$



Quantum valuation:

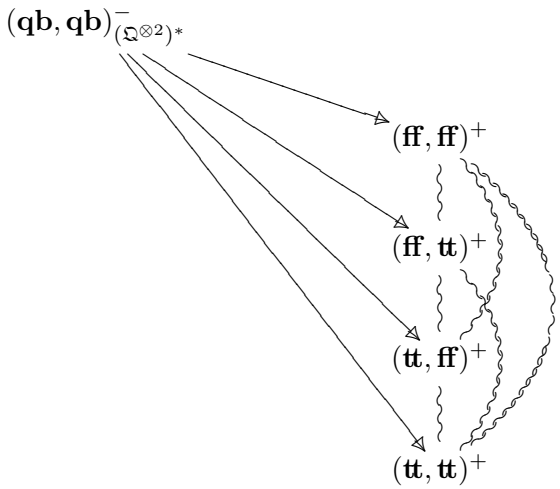
$$\begin{array}{rcl}
\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(x) & = & \mathbf{id_1} \\
\text{whenever} & & |x| \leq 1 \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(\{(\mathbf{ff},\mathbf{ff}),(\mathbf{qb},\mathbf{qb})\}) & = & (\!|\tfrac{1}{\sqrt{2}}|00\rangle + \tfrac{1}{\sqrt{2}}|11\rangle|\!) \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(\{(\mathbf{ff},\mathbf{tt}),(\mathbf{qb},\mathbf{qb})\}) & = & (\!|\tfrac{1}{\sqrt{2}}|01\rangle + \tfrac{1}{\sqrt{2}}|10\rangle|\!) \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(\{(\mathbf{tt},\mathbf{ff}),(\mathbf{qb},\mathbf{qb})\}) & = & (\!|\tfrac{1}{\sqrt{2}}|00\rangle - \tfrac{1}{\sqrt{2}}|11\rangle|\!) \\
\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(\{(\mathbf{tt},\mathbf{tt}),(\mathbf{qb},\mathbf{qb})\}) & = & (\!|\tfrac{1}{\sqrt{2}}|01\rangle - \tfrac{1}{\sqrt{2}}|10\rangle|\!)
\end{array}$$

Term represented:

$b : \mathbf{bit}, b' : \mathbf{bit} \vdash_{\mathbb{L}} \mathrm{Bell}(b, b') : \mathbf{qubit} \otimes \mathbf{qubit}$
$\mathrm{Bell}(b, b') := \mathbf{let}\ q_1 \otimes q_2 = (\mathbf{new}\ b) \otimes (\mathbf{new}\ b')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in\ Nc}\ (q_3 \otimes q_2)$

Figure 6.8: Strategy for the Bell States $[\![\mathrm{Bell}(b, b')]\!]$

$$\mathbf{qubit}, \mathbf{qubit} \quad \vdash_{\mathbb{L}} \quad \mathbf{bit} \otimes \mathbf{bit}$$



Quantum valuation:

$$\begin{array}{rcl} \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\varnothing) & = & \mathbf{id_1} \\ \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\{(\mathbf{qb},\mathbf{qb})\}) & = & \mathbf{Tr}_{\mathbb{Q}^{\otimes 2}} \end{array}$$

$$\begin{array}{ll} \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\{(\mathbf{qb},\mathbf{qb}),(\mathbf{ff},\mathbf{ff})\}): & M \mapsto \frac{m_{11}+m_{14}+m_{41}+m_{44}}{2} \\ \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\{(\mathbf{qb},\mathbf{qb}),(\mathbf{ff},\mathbf{tt})\}): & M \mapsto \frac{m_{22}+m_{23}+m_{32}+m_{33}}{2} \\ \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\{(\mathbf{qb},\mathbf{qb}),(\mathbf{tt},\mathbf{ff})\}): & M \mapsto \frac{m_{11}-m_{14}-m_{41}+m_{44}}{2} \\ \mathcal{Q}^{-,+}_{[\![\mathrm{BellM}(q,q')]\!]}(\{(\mathbf{qb},\mathbf{qb}),(\mathbf{tt},\mathbf{tt})\}): & M \mapsto \frac{m_{22}-m_{23}-m_{32}+m_{33}}{2} \end{array}$$

Term represented:

$q : \mathbf{qubit}, q' : \mathbf{qubit} \vdash_{\mathbb{L}} \mathrm{BellM}(q, q') : \mathbf{bit} \otimes \mathbf{bit}$
$\mathrm{BellM}(q, q') := \mathbf{let}\ q_1 \otimes q_2 = \mathbf{Nc}\ (q \otimes q')\ \mathbf{in\ let}\ q_3 = \mathbf{H}\ q_1\ \mathbf{in}\ (\mathbf{meas}\ q_3) \otimes (\mathbf{meas}\ q_2)$

Figure 6.9: Strategy for the Bell Measure $[\![\mathrm{BellM}(q, q')]\!]$

the same point of the web, in which case the correspondence between quantum valuations of strategies and quantum annotations of weighted relations uses a sum.

The Bell unitary is slightly more interesting, as we have some higher-order behaviour, but the semantics is still almost identical to the relational semantics. We chose to use quantum valuations of the form $\mathcal{Q}^{-,+}$ to highlight the fact that the game semantics allows us to describe terms using only superoperators.

The Quantum teleportation has some very interesting quantum valuations, that we describe in Fig. 6.11 and explain as:

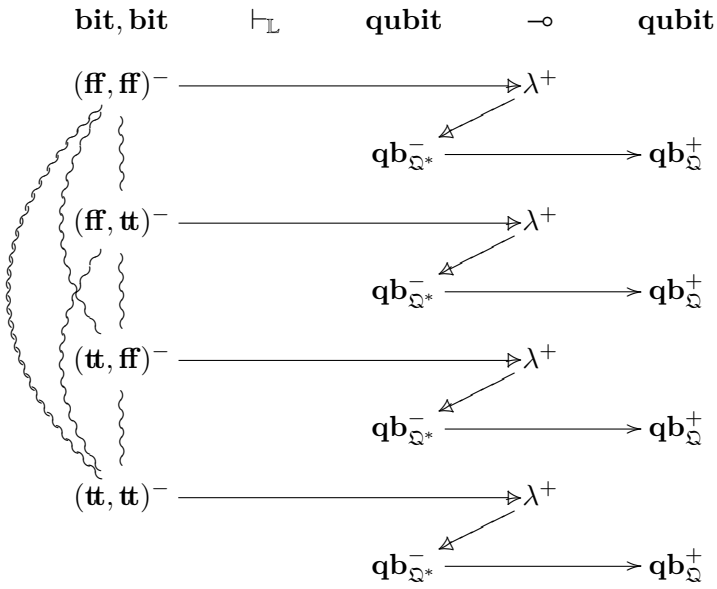
- As long as no positive move is played, or that only $(\lambda, \lambda)^+$ is played, there are no observable quantum effects and all the quantum states are discarded without being used, so the valuation is a trace. We note that operationally, the event $(\lambda, \lambda)^+$ is when the Bell state qubits are computed by the term, but this is not observable.
- If the positive move $(b, b')^+$ is played, but not yet $\mathbf{qb}^+_{\mathfrak{Q}}$, then operationally the Bell measurement was made, hence the $\frac{1}{4}$ of chance of getting the boolean $(b, b')^+$, but since the remaining qubits are kept internally, they are not observable and the quantum valuation is still a trace.
- If the positive move $\mathbf{qb}^+_{\mathfrak{Q}}$ is played but not yet (b, b'), then the Bell unitary was applied. However, since we are still keeping one of the two qubits from the Bell state internally, the second qubit is indistinguishable from $\mathbf{1}_{\mathfrak{Q}} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$, which is a fixpoint of each of the four Bell unitary operations.
- If we consider a maximal configuration, then operationally the full term was executed and the quantum valuation is the same as the quantum annotation from the relational semantics as in Section 3.2.2

We note that we computed Fig. 6.11 by making the following chain of interactive composition:

$$[\![\text{QTelep}()]\!] = \left([\![\lambda q.\text{BellM(x,q)}]\!] \otimes [\![\lambda(b, b').\text{BellU}(b, b')\ x]\!]\right) \odot [\![\text{Bell}(\mathbf{ff}, \mathbf{ff})]\!]$$

6.2 An ≡-Adequate Model

The goal of this section is to amend our model to one that will later be shown to be fully abstract. The two obstacles are that our model remembers branching points, and remembers some causal information like the evaluation order of functions. In our earlier publication [CdVW19], we use a notion called rigid-equivalence to forget those branching points. This notion however does not forget the evaluation order, so proved to be insufficient for full abstraction. So instead, we develop the notion of exhaustive equivalence, as it is



Quantum valuation:

$$\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(x) = \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}} \quad \text{whenever} \quad |x| \leq 2$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{Bell}(b,b')]\!]}(x) = \mathbf{Tr}_{\mathfrak{Q}} \quad \text{whenever} \quad |x| = 3$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{BellU}(b,b')]\!]}(\{(\mathbf{ff},\mathbf{ff}),\lambda,\mathbf{qb},\mathbf{qb}\}) = (\!|\begin{pmatrix}1&0\\0&1\end{pmatrix}|\!)$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{BellU}(b,b')]\!]}(\{(\mathbf{ff},\mathbf{tt}),\lambda,\mathbf{qb},\mathbf{qb}\}) = (\!|\begin{pmatrix}0&1\\1&0\end{pmatrix}|\!)$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{BellU}(b,b')]\!]}(\{(\mathbf{tt},\mathbf{ff}),\lambda,\mathbf{qb},\mathbf{qb}\}) = (\!|\begin{pmatrix}1&0\\0&-1\end{pmatrix}|\!)$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{BellU}(b,b')]\!]}(\{(\mathbf{tt},\mathbf{tt}),\lambda,\mathbf{qb},\mathbf{qb}\}) = (\!|\begin{pmatrix}0&1\\-1&0\end{pmatrix}|\!)$$

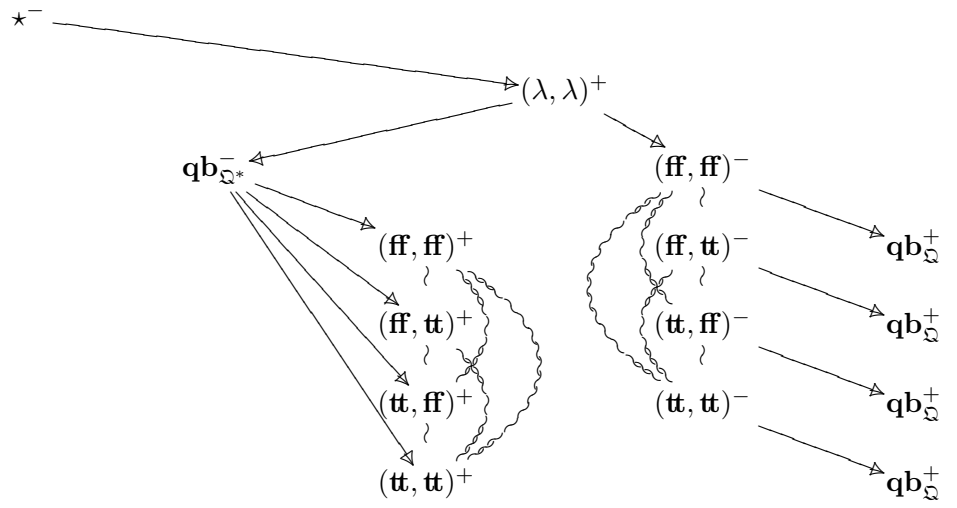
Term represented:

$b : \mathbf{bit}, b' : \mathbf{bit} \vdash_{\mathbb{L}} \mathrm{BellU}(b,b') : \mathbf{qubit} \multimap \mathbf{qubit}$

$$\mathrm{BellU}(b,b') := \lambda q.\mathbf{if}\ b \quad \mathbf{then} \quad \mathbf{if}\ b'\ \mathbf{then}\ \begin{pmatrix}0&1\\-1&0\end{pmatrix}\ q\ \mathbf{else}\ \begin{pmatrix}1&0\\0&-1\end{pmatrix}\ q$$
$$\mathbf{else} \quad \mathbf{if}\ b'\ \mathbf{then}\ \begin{pmatrix}0&1\\1&0\end{pmatrix}\ q\ \mathbf{else}\ \begin{pmatrix}1&0\\0&1\end{pmatrix}\ q$$

Figure 6.10: Strategy for the Bell Unitary $[\![\mathrm{BellU}(b,b')]\!]$

$$\varnothing \quad \vdash_{\mathbb{L}} \quad (\mathbf{qubit} \quad \multimap \quad \mathbf{bit}^{\otimes 2}) \qquad \otimes \qquad (\mathbf{bit}^{\otimes 2} \quad \multimap \quad \mathbf{qubit})$$



Quantum valuation:

$$\begin{array}{lcll}
\mathcal{Q}^{-,+}_{[\![\mathrm{QTelep}()]\!]}(x) & = & \mathbf{Tr} & \text{whenever} \quad \{e^+ \in x\} = \varnothing \text{ or } \{(\lambda,\lambda)\} \\
\mathcal{Q}^{-,+}_{[\![\mathrm{QTelep}()]\!]}(x) & = & \frac{1}{4}\mathbf{Tr} & \text{whenever} \quad \{e^+ \in x\} = \{(\lambda,\lambda),(b,b')\} \text{ for } b,b' \text{ booleans} \\
\mathcal{Q}^{-,+}_{[\![\mathrm{QTelep}()]\!]}(x) & = & \mathbf{Tr} \otimes \mathbf{1}_{\mathfrak{Q}} & \text{whenever} \quad \{e^+ \in x\} = \{(\lambda,\lambda),\mathbf{qb}\}
\end{array}$$

$$\mathcal{Q}^{-,+}_{[\![\mathrm{QTelep}()]\!]}(\{\star^-,(\lambda,\lambda)^+,\mathbf{qb}^-,(b_0,b_0')^+,(b_1,b_1)^-,\mathbf{qb}^+\}) =$$
$$\begin{array}{rlllll}
\mathbf{id}_{\mathfrak{Q}} & \text{if} & b_0 = b_1 & \text{and} & b_0' = b_1' \\
(\!|\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}|\!) & \text{if} & b_0 = b_1 & \text{and} & b_0' \neq b_1' \\
(\!|\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}|\!) & \text{if} & b_0 \neq b_1 & \text{and} & b_0' = b_1' \\
(\!|\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}|\!) & \text{if} & b_0 \neq b_1 & \text{and} & b_0' \neq b_1'
\end{array}$$

Term represented:

$\vdash_{\mathbb{L}} \mathrm{QTelep}(q,q') : (\mathbf{qubit} \multimap \mathbf{bit}^{\otimes 2}) \otimes (\mathbf{bit}^{\otimes 2} \multimap \mathbf{qubit})$
$\mathrm{QTelep}() := \mathbf{let}\ q_1 \otimes q_2 = \mathrm{Bell}(\mathbf{ff},\mathbf{ff})\ \mathbf{in}\ (\lambda q.\mathrm{BellM}(q_1,q)) \otimes (\lambda(b \otimes b').\mathrm{BellU}(b,b'))$

Figure 6.11: Strategy for the Quantum Teleportation $[\![\mathrm{QTelep}()]\!]$

done in our later publication [CdV20]. The idea behind the notion of exhaustive equivalence is to select some "observable" configurations of the game, and say that two strategies are equivalent if they behave the same with respect to those specific configurations.

One of the main difficulties of this exhaustive equivalence is that since we are selecting configurations of the game and abstracting the behaviour of the strategy to only those configurations, we are losing some of the causal information of the strategy, which might mean that when composing strategies, we might miss some causal loops and deadlocks, meaning the equivalence would not behave properly with respect to the interactive composition. In other words, this equivalence between strategies might not be a congruence for the operations involved in the interpretation.

Fortunately, the condition of visibility announced earlier in Definition 5.5.1 ensures the absence of such deadlocks, hence the congruence property. We start by defining visibility.

6.2.1 Visibility and Deadlock-Free Composition

In Definition 5.5.1, we included a condition named visibility. In this subsection, we formally define it, and present one of its properties useful for the proof of adequacy: deadlock free composition, *i.e.*, when checking that a pair of configurations is matching compatible, visibility ensures matching pairs are always compatible.

The notion of visibility we define here comes from [CCW15b], where it was introduced as a weaker form of innocence and used to prove the preservation of conditions like innocence and well-bracketing under interactive composition of strategy. While it shares some intuition with the visibility condition of sequential games as in [HO00], there is no direct correspondence between these two notions of visibility. A small contribution of this thesis is the extension of the definition of visibility to less restricted forms of event structures, in particular ones that are non-forest-like. However, as the semantics of the quantum λ-calculus only uses forest-like games, this contribution is not a focus or this thesis, hence we choose to postpone the generalised proofs to Appendix B.2.

Definition 6.2.1. *In an event structure E, a grounded causal chain, or gcc, is a sequence of events $e_1 \rightarrowtriangle_E e_2 \rightarrowtriangle_E \dots \rightarrowtriangle_E e_n$ with e_1 minimal in E.*

A gcc can be seen as a thread of the computation described by the event structure. For example, in Fig. 6.12, $(\lambda_0, \lambda_1)^- \rightarrowtriangle \star_0^+ \rightarrowtriangle \star_0^- \rightarrowtriangle \star^+$ is the gcc corresponding to the thread where the left hand side function is called. For $\rho = e_1 \rightarrowtriangle_E \dots \rightarrowtriangle_E e_n$ a gcc, we write $\rho \rightarrowtriangle_E e$ for the gcc $e_1 \rightarrowtriangle_E \dots \rightarrowtriangle_E e_n \rightarrowtriangle_E e$, and we write $e \in \rho$ for $\exists i, e = e_i$.

Definition 6.2.2. *A strategy $\sigma : A \nrightarrow B$ with A and B games is called visible if for every gcc $\rho = \rho' \rightarrowtriangle_S s$, there exists $s' \in \rho'$ such that $\sigma(s') \rightarrowtriangle_{A^\perp \parallel B} \sigma(s)$.*

In this situation, we say that s' justifies s in ρ. This justification relation is very similar to the notion of pointers in sequential game semantics. The idea is that from the point of view of the thread represented by the gcc ρ', for the event s to be added to this thread,

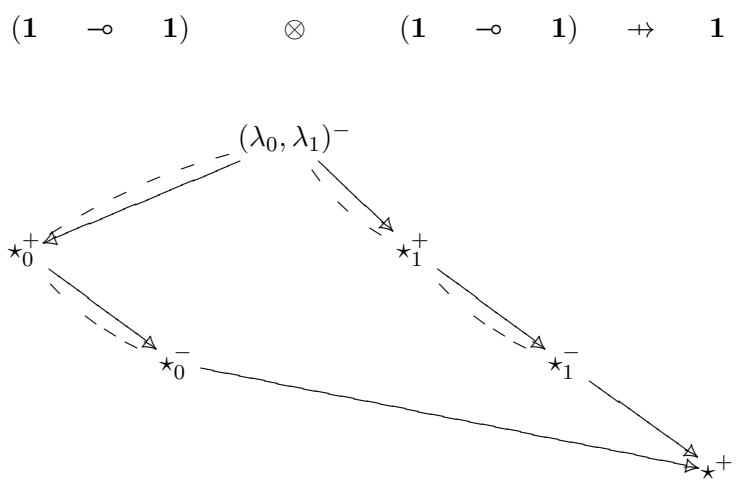


Figure 6.12: Example

the event s needs to be "visible", *i.e.*, there is an event $s' \in \rho'$ which "enables" it. We note that in our diagrams for strategies, we chose to represent the causality of the game through dashed lines, so those dashed lines exactly represent the justification relation.

Another way of understanding visibility is the notion of scope in programming languages. Consider a program which declares two functions $f_0 : \mathbf{1} \multimap \mathbf{bit}$ and $f_1 : \mathbf{bit} \multimap \mathbf{1}$, or in a pseudo programming language:

```
bool f0(u0:unit):{
    \\Body of f0
    }
unit f1(b1:bool):{
    \\Body of f1
    }
```

The function f_0 cannot use in its body the input b_1 of f_1 since it is not in its scope, and similarly the function f_1 cannot use in its body the input u_0 of f_0. In languages with powerful enough primitives, it is possible to bypass this restriction and use communication channels between f_0 and f_1 so that if both are executed in parallels, f_0 would be able to communicate u_0 to f_1 and reciprocally f_1 would be able to communicate b_1 to f_0. This behaviour can be described by the strategy Fig. 6.13. The visibility condition states that there is no such communication method. In particular, the strategy in Fig. 6.13 which describes two functions that use the input of one another is not visible. This strategy breaks the visibility condition because of the gcc $(\star^- \rightarrowtriangle (\lambda_0, \lambda_1)^+ \rightarrowtriangle \mathbf{ff}_1^- \rightarrowtriangle \mathbf{ff}_0^+)$, as $\mathbf{ff}_0^+$ can only be justified by $\star_0^-$ which is not in the gcc.

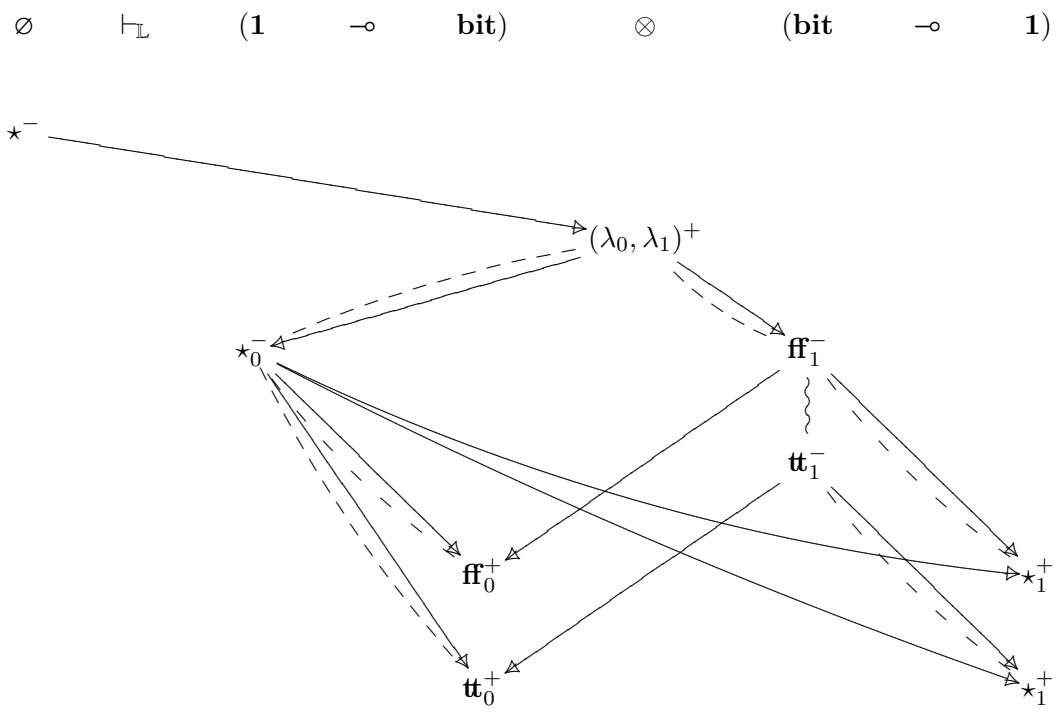


Figure 6.13: Example of non visible strategy

We note that whenever A and B are tree-like, for example arenas, it is equivalent to ask for $\sigma\,\rho$ to be a configuration for every gcc ρ.

Proposition 6.2.3. *Copy-cat is visible.*

This is pretty straightforward to prove, as gccs of copy-cat are always of the form $a^- \rightarrowtriangle a^+ \rightarrowtriangle b^- \rightarrowtriangle b^+ \rightarrowtriangle \ldots$, *i.e.*, alternating between a negative event of $A^\perp$ or A and the corresponding positive event in A or $A^\perp$ respectively, with each negative event being justified by the positive event before it.

Proposition 6.2.4. *The interactive composition of two visible strategies between arenas is visible.*

Theorem 6.2.5 (Deadlock-Free Composition)**.** *We assume A, B, C to be arenas. If $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ are two visible strategies, and $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(T)$ are a pair of matching configurations, then they are matching compatible.*

We refer to [Cas17] for the usual proof for both. However, this proof only covers arenas,

and relies on their forest-like shape. We provide new proofs in Appendix B.2 which cover strategies between polarised[1] N-free[2] games.

6.2.2 The Exhaustive Equivalence ≡

As said before, we define a notion of "observable configurations", which we name "exhaustive configurations" as in Part III they will represent configurations where "every strictly linear function has been called and fully resolved, and every non-linear function that has been called has been fully resolved". For every exhaustive configuration, we are interested in the **CPM** operator obtained by summing all the corresponding valuations in the strategy.

Definition 6.2.6. *A configuration of a game is called* exhaustive *if its payoff is* 0*. We write* $\mathcal{E}(A)$ *for the set of exhaustive configurations of* A*. Assuming* $\sigma : A \nrightarrow B$ *a quantum strategy, and for* $x_A \parallel x_B \in \mathcal{E}(A^\perp \parallel B)$*, we write* $\mathrm{wit}_\sigma(x_A, x_B)$ *for the set of* $\oplus$*-covered configurations* x *such that* $\sigma x = x_A \parallel x_B$*, called witnesses of* $x_A \parallel x_B$*. We then write*

$$\mathcal{Q}_\sigma(x_A, x_B) = \sum_{x \in \mathrm{wit}_\sigma(x_A, x_B)} \mathcal{Q}_\sigma(x) \in \overline{\mathbf{CPM}}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$$

A priori, $\mathcal{Q}_\sigma(x_A, x_B)$ might be an infinite sum, however we can prove that it is always in the finitary fragment of $\overline{\mathbf{CPM}}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$.

Theorem 6.2.7. *For* $\sigma : A \nrightarrow B$ *a quantum strategy and* $x_A \parallel x_B \in \mathcal{E}(A^\perp \parcr B)$*, we have*

$$\frac{\mathbf{Tr}_{\mathcal{H}_B(x_B)}}{\dim \mathcal{H}_B(x_B^-)} \circ \mathcal{Q}_\sigma(x_A, x_B) \circ (\dim \mathcal{H}_A(x_A^-)) \cdot \mathbf{1}_{\mathcal{H}_A(x_A)} \sqsubseteq \mathbf{id}_{\mathbf{1}} \in \mathbf{CPM}_{\leq 1}(\mathbf{1}, \mathbf{1})$$

In particular $\mathcal{Q}_\sigma(x_A, x_B)$ *is finitary, i.e.,* $\mathcal{Q}_\sigma(x_A, x_B) \in \mathbf{CPM}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$*.*

Proof. For $\sigma : A \nrightarrow B$ a quantum strategy, we start by defining $\overline{\sigma} \in \mathbf{QStrat}(\varnothing, A^\perp \parallel B)$ obtained through the compact closure of **QStrat**. We write $C = A^\perp \parallel B$.

We rely on the test strategy from Lemma A.3.2. For every $x_C \in \mathcal{C}(C)$, we have a strategy $\mathrm{test}_C(x_C) \in \mathbf{Strat}(C, \mathbf{1})$. We extend it as a quantum strategy with the valuation:

$$\mathcal{Q}_{\mathrm{test}_C(x_C)}(y) = \mathbf{Tr}_{\mathcal{H}_C(y_C^-)} \otimes \mathbf{1}^\dagger_{\mathcal{H}_{C^\perp}(y_C^+)} = \frac{\mathbf{Tr}_{\mathcal{H}_C(y_C)}}{\dim \mathcal{H}_C(y_C^-)}$$

where $y_C \parallel y_{\mathbf{1}}$ is the projection of y to the game, and y_C^- and y_C^+ are the sets of respectively negative events and positive events of y_C. We check normalisation, obliviousness and the drop condition without difficulties. So $\mathrm{test}_C(x_C) \in \mathbf{QStrat}(C, \mathbf{1})$.

We consider $x_C = x_A \parallel x_B \in \mathcal{E}(A^\perp \parcr B)$. If $x \in \mathrm{wit}_\sigma(x_A, x_B)$, then $\overline{\sigma}\, x = \varnothing \parallel x_C$

[1] A polarised game is a game which has all its minimal events of the same polarity.

[2] See Definition B.2.5 for the definition of N-free games.

and x is $\oplus$-covered, so using Lemma A.3.2, if we can write $\mathrm{test}_C(x_C) \odot \overline{\sigma}$, then $((x_C \parallel \{\star\}) \odot x)$ is defined and $\tau((x_C \parallel \{\star\}) \odot x) = \varnothing \parallel \{\star\}$.

We note that Lemma 5.3.5 ensures that for any quantum strategy ν, for every $z \in \mathcal{C}(N)$ a configuration of the strategy, if $U \subseteq V$ are two finite set of configurations that are positive extensions of z, then $d_\nu(z;U) \sqsupseteq d_\nu(z;V) \in \mathbf{CPM}$. As such, for every possibly infinite set W of configurations that are positive extensions of z, we can define $d_\nu(z;W) \in \mathbf{CPM}$ as the infimum for the $d_\nu(z,W')$ for W' finite subset of W. It follows that

$$\begin{array}{rcl}
\mathbf{id_1} & \sqsupseteq & \mathbf{id_1} - d_\tau(\varnothing, \{y \mid \tau\, y \neq \varnothing\}) \\
& = & \sum_{y \mid \tau\, y \neq \varnothing} \mathcal{Q}_\tau(y) \\
& \sqsupseteq & \sum_{x \in \mathrm{wit}_\sigma(x_A, x_B)} \mathcal{Q}_\tau((x_C \parallel \{\star\}) \odot x) \\
& = & \sum_{x \in \mathrm{wit}_\sigma(x_A, x_B)} \mathcal{Q}_{\mathrm{test}_C(x_C)}(x_C \parallel \{\star\}) \circ \mathcal{Q}_{\overline{\sigma}(x)} \\
& = & \sum_{x \in \mathrm{wit}_\sigma(x_A, x_B)} \frac{\mathbf{Tr}_{\mathcal{H}_C(x_C)}}{\dim \mathcal{H}_C(x_C^-)} \circ \mathcal{Q}_{\overline{\sigma}(x)}
\end{array}$$

Using the compact closure of **QStrat** and **CPM** he have that

$$\begin{array}{rcl}
\frac{\mathbf{Tr}_{\mathcal{H}_C(x_C)}}{\dim \mathcal{H}_C(x_C^-)} \circ \mathcal{Q}_{\overline{\sigma}(x)} & = & \frac{\mathbf{Tr}_{\mathcal{H}_B(x_B)}}{\dim \mathcal{H}_B(x_B^-)} \circ \mathcal{Q}_\sigma(x) \circ \frac{\mathbf{Tr}^\dagger_{\mathcal{H}_A(x_A)}}{\dim \mathcal{H}_A(x_A^+)} \\
& = & \frac{\mathbf{Tr}_{\mathcal{H}_B(x_B)}}{\dim \mathcal{H}_B(x_B^-)} \circ \mathcal{Q}_\sigma(x) \circ (\dim \mathcal{H}_A(x_A^-)) \cdot \mathbf{1}_{\mathcal{H}_A(x_A)}
\end{array}$$

□

We now define the exhaustive equivalence, which states that strategies are exhaustively equivalent if and only if they behave the same on exhaustive configurations.

Definition 6.2.8. *For $\sigma, \tau : A \nrightarrow B$ two quantum strategies, we say that they are exhaustively equivalent, and write $\sigma \equiv \tau$ whenever for all $x_A \parallel x_B \in \mathcal{E}(A^\perp \parr B)$, $\mathcal{Q}_\sigma(x_A, x_B) = \mathcal{Q}_\tau(x_A, x_B)$.*

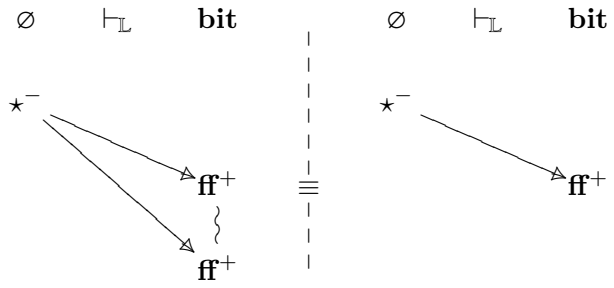
We note that two isomorphic strategies are exhaustively equivalent, and so are strategies rigidly equivalent as defined in [CdVW19].

In Figs. 6.14 and 6.15, we show two examples of exhaustive equivalence. These examples illustrate that the exhaustive equivalence allows us to "merge" configurations of the strategy that have the same projection in the game, and that at the term level correspond to observationally equivalent terms. In particular, Fig. 6.15 shows that the exhaustive equivalence fully forgets in which order functions are called.

We have idempotence and full linearity of $\boxplus$ up to this equivalence:

$$p \cdot \sigma \boxplus q \cdot \sigma \equiv (p+q) \cdot \sigma$$

$$\tau_1 \odot \left(\bigboxplus_{i \in I} p_i \cdot \sigma_i \right) \odot \tau_2 \equiv \bigboxplus_{i \in I} p_i \cdot (\tau_1 \odot \sigma_i \odot \tau_2)$$



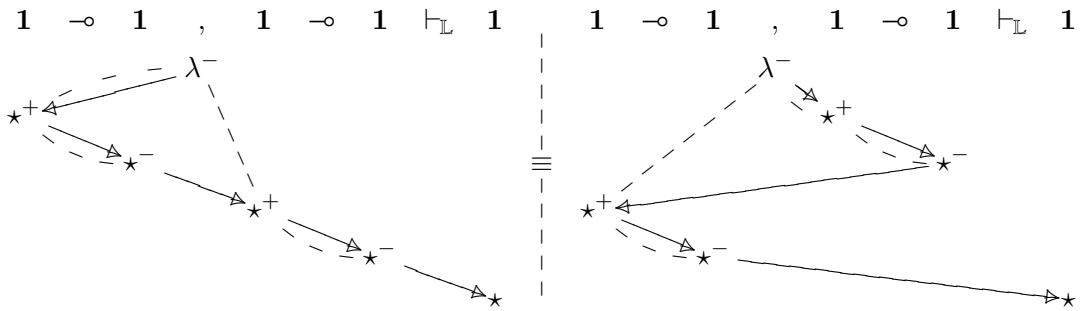
Quantum valuations

$$\mathcal{Q}(x) = \begin{cases} \mathbf{id_1} & \text{if } |x| \leq 1 \\ \frac{1}{2}\mathbf{id_1} & \text{otherwise} \end{cases} \qquad \mathcal{Q}(y) = \mathbf{id_1}$$

Terms represented

$$\vdash_{\mathbb{L}} \mathbf{if}\ \mathrm{Coin}_{\frac{1}{2}}()\ \mathbf{then\ ff\ else\ ff} : \mathbf{bit} \qquad \vdash_{\mathbb{L}} \mathbf{ff} : \mathbf{bit}$$

Figure 6.14: First example of two exhaustively equivalent strategies



Quantum valuations

$$\mathcal{Q}(x) = \mathbf{id_1} \qquad \mathcal{Q}(y) = \mathbf{id_1}$$

Terms represented

$$f_0 : \mathbf{1} \multimap \mathbf{1}, f_1 \mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} f_0(); f_1() : \mathbf{1} \qquad f_0 : \mathbf{1} \multimap \mathbf{1}, f_1 \mathbf{1} \multimap \mathbf{1} \vdash_{\mathbb{L}} f_1(); f_0() : \mathbf{1}$$

Figure 6.15: Second example of two exhaustively equivalent strategies

This will allow us to obtain adequacy results. But before that, we show that ≡ is compatible with the structure of **QA**.

Theorem 6.2.9. *The relation ≡ is a congruence in* **QCG** *restricted to visible strategies and forest-like games, meaning it is compatible with ⊙, ⊠, and ⅋, and a congruence in* **QA***, meaning it is compatible with ⊙, ⊗, ⊸, ⊸• and ⊕.*

Proof. Most proofs are straight forward, except for ⊙. We first consider $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ visible.

$$\mathcal{Q}_{\tau\odot\sigma}(x_A, x_C) = \sum_{z\odot y \in \mathrm{wit}_{\tau\odot\sigma}(x_A,x_C)} \mathcal{Q}_\tau(z) \circ \mathcal{Q}_\sigma(y)$$

We recall that Lemma A.2.1 ensures that $z \odot y$ is $\oplus$-covered if and only if z and y are. Additionally, visibility ensures that matching configurations are compatible. So we have

$$\mathcal{Q}_{\tau\odot\sigma}(x_A, x_C) = \sum_{x_B \in \mathcal{C}(B)} \sum_{z \in \mathrm{wit}_\tau(x_B,x_C)} \mathcal{Q}_\tau(z) \circ \sum_{y \in \mathrm{wit}_\sigma(x_A,x_B)} \mathcal{Q}_\sigma(y)$$

Since both σ and τ are winning, then the right hand side sum is null whenever $\kappa_B(x_B) < 0$ and the left hand side sum is null whenever $\kappa_{B^\perp}(x_B) < 0$. This mean that:

$$\mathcal{Q}_{\tau\odot\sigma}(x_A, x_C) = \sum_{x_B \in \mathcal{E}(B)} \sum_{z \in \mathrm{wit}_\tau(x_B,x_C)} \mathcal{Q}_\tau(z) \circ \sum_{y \in \mathrm{wit}_\sigma(x_A,x_B)} \mathcal{Q}_\sigma(y)$$

$$\mathcal{Q}_{\tau\odot\sigma}(x_A, x_C) = \sum_{x_B \in \mathcal{E}(B)} \mathcal{Q}_\tau(x_A, x_B) \circ \mathcal{Q}_\sigma(x_A, x_B)$$

If $\sigma \equiv \sigma' : A \nrightarrow B$ and $\tau \equiv \tau' : B \nrightarrow C$ then using this equation, $\tau \odot \sigma \equiv \tau' \odot \sigma'$. □

6.2.3 ≡-Adequacy

We now state all the results leading to adequacy up to exhaustive equivalence.

Lemma 6.2.10 (≡-Invariance)**.** *For every closures $\Gamma \vdash c : A$ and $\Gamma \vdash c' : A$*

$$c \to c' \implies [\![c]\!] \equiv \left[\!\left[c'\right]\!\right]$$

Proof. For all the reduction but the last two, he proof is the same as Lemma 3.2.6. For the last two reductions, we rely on ⊞ being associative, commutative, idempotent up to ≡, and linear up to ≡. □

Theorem 6.2.11 (Soundness and ≡-Adequacy)**.** *For every term $\vdash t : \mathbf{1}$, we have*

$$\mathbb{P}(t \Downarrow) = p \iff [\![t]\!] \equiv [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$$

The proof is the same as the one of Theorem 3.2.7.

Corollary 6.2.12. *For every pair of terms $\Gamma \vdash t : A$ and $\Gamma \vdash s : A$, we have*

$$[\![t]\!] \equiv [\![s]\!] \implies t =_{\text{obs}}^{\Gamma \vdash A} s$$

Proof. We take an observation context $\mathcal{O}[_]$ for $\Gamma \vdash A$. A simple proof by induction shows that there exists a function F such that for all term $\Gamma \vdash t : A$ we have $[\![\mathcal{O}[t]]\!] = F([\![t]\!])$, and this function preserves $\equiv$. We assume that $[\![t]\!] \equiv [\![s]\!]$. We have $[\![\mathcal{O}[t]]\!] = F([\![t]\!]) \equiv F([\![s]\!]) = [\![\mathcal{O}[s]]\!]$. Using adequacy, it follows that $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow)$, hence the result. □

6.3 ≡-Full Abstraction for LQΛ

In this section, we tweak the proof of full abstraction of **QRel** from Section 3.2.4 into a proof of full abstraction of **QA** for LQΛ. We postpone to the next section the proof that $\mathbf{QA}^a$ is fully abstract for AQΛ.

6.3.1 Web of a Type

The core of the full abstraction proof is to note that exhaustive configurations in the game semantics take the role of the web from the relational semantics. For example, for $A = \mathbf{1} \multimap \mathbf{bit}$ a type of LQΛ, we have a clear correspondence between the two:

$$\mathcal{E}([\![A]\!]) = \{\{\lambda^+, \star^-, \mathbf{ff}^+\}, \{\lambda^+, \star^-, \mathbf{tt}^+\}\} \qquad |A| = \{(\star, \mathbf{ff}), (\star, \mathbf{tt})\}$$

We recall the definition of web of a type and its associated Hilbert space, and then build a bijection between exhaustive configurations of $[\![A]\!]$ and points of the web $|A|$.

Definition 6.3.1. *For A a type of LQΛ, we define the web $|A|$, and the Hilbert space $\mathcal{H}_A(a)$ for $a \in |A|$.*

$$\begin{array}{rcll} |\mathbf{1}| & := & \{\star\} & \\ |\mathbf{qubit}| & := & \{\mathbf{qb}\} & \\ |A \oplus B| & := & |A| \uplus |B| & \\ |A \otimes B| & := & |A| \times |B| & \\ |A \multimap B| & := & |A| \times |B| & \\ |A^\ell| & := & \{\star\} \uplus (|A| \times |A^\ell|) & \textit{(smallest fixpoint)} \end{array}$$

$$\begin{array}{rccc}
\mathcal{H}_{\mathbf{1}} : & \star & \mapsto & \mathbf{1} \\
\mathcal{H}_{\mathbf{qubit}} : & \mathbf{qb} & \mapsto & \mathfrak{Q} \\
\mathcal{H}_{A\oplus B} : & (0,a) & \mapsto & \mathcal{H}_A(a) \\
 & (1,b) & \mapsto & \mathcal{H}_B(b) \\
\mathcal{H}_{A\otimes B} : & (a,b) & \mapsto & \mathcal{H}_A(a)\otimes\mathcal{H}_B(b) \\
\mathcal{H}_{A\multimap B} : & (a,b) & \mapsto & \mathcal{H}_A(a)\otimes\mathcal{H}_B(b) \\
\mathcal{H}_{A^\ell} : & (0,\star) & \mapsto & \mathbf{1} \\
 & (1,(a,b)) & \mapsto & \mathcal{H}_A(a)\otimes\mathcal{H}_{A^\ell}(b)
\end{array}$$

Definition 6.3.2. *For A a type of LQΛ, we define a bijection $r_A : \mathcal{E}(\llbracket A \rrbracket) \to |A|$ inductively:*

$$\begin{array}{rccc}
r_{\mathbf{1}} : & \{\star^+\} & \mapsto & \star \\
r_{\mathbf{qubit}} : & \{\mathbf{qb}^+\} & \mapsto & \mathbf{qb} \\
r_{A\oplus B} : & x_A \oplus \varnothing & \mapsto & (0, r_A(x_A)) \\
 & \varnothing \oplus x_B & \mapsto & (1, r_B(x_B)) \\
r_{A\otimes B} : & x_A \otimes x_B & \mapsto & (r_A(x_A), r_B(x_B)) \\
r_{A\multimap B} : & x_A \multimap x_B & \mapsto & (r_A(x_A), r_B(x_B)) \\
r_{A^\ell} : & [\,] & \mapsto & (0,\star) \\
 & [x_1;\ldots;x_n] & \mapsto & (1,(r_A(x_1), r_{A^\ell}([x_2;\ldots;x_n])))
\end{array}$$

This definition is well behaving, in particular, for A a type of LQΛ, and for $x \in \mathcal{E}(\llbracket A \rrbracket)$ we have $\mathcal{H}_{\llbracket A \rrbracket_{\mathbf{QA}}}(x) = \mathcal{H}_A(r_A(x))$. But as we will show in Section 6.3.3, it can be reformulated into a functor from the game semantics to the relational semantics.

6.3.2 Test Terms and Generator Terms

To prove full abstraction for our game semantics, we rely on test terms and generator terms as in the relational case. In this section, we recall the definition of test terms and generator terms.

We first recall that the extended web $|A|_e$ of a type A is defined exactly as the web except for $|\mathbf{qubit}|_e = |\mathbf{qubit}| \times \{0,1,2,3\}$. We use the notation $a|i \in |A|_e$ with $a \in |A|$ and $i \in \{0,1,2,3\}^{\#_{\mathbf{qubit}}(a)}$ as formalised in Section 3.2.4.

Definition 6.3.3. *For every element of $|A|_e$, we define a generator term and a test term, written $\Uparrow_a^A$ and $\Downarrow_a^A$ as in Tables 6.2 and 6.3. They are typed by:*

$$\vdash_{\mathbb{L}} \Uparrow_a^A : A \qquad\qquad \vdash_{\mathbb{L}} \Downarrow_a^A : A \multimap \mathbf{1}$$

We note that the terms for **qubit** *are built from the morphisms of Proposition 2.1.18.*

As in the relational case, the role of test terms is to "extract" the coefficient corresponding to a point of the web by "replaying" a given configuration. Formally, we expect that for any term $x : A \vdash_{\mathbb{L}} t : B$, we have:

$$
\begin{array}{lcllcl}
\Uparrow^{\mathbf{1}}_{\star} & := & () & & & \\
\Uparrow^{\mathbf{qubit}}_{\mathbf{qb},0} & := & \mathbf{new}\ \mathbf{ff} & \Uparrow^{\mathbf{qubit}}_{\mathbf{qb},1} & := & \mathbf{new}\ \mathbf{tt} \\
\Uparrow^{\mathbf{qubit}}_{\mathbf{qb},2} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \mathbf{new}\ \mathbf{ff} & \Uparrow^{\mathbf{qubit}}_{\mathbf{qb},3} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{\mathbf{i}}{\sqrt{2}} \\ \frac{\mathbf{i}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \mathbf{new}\ \mathbf{ff} \\
\\
\Uparrow^{A\otimes B}_{(a,b)} & := & \Uparrow^{A}_{a} \otimes \Uparrow^{B}_{b} & \Uparrow^{A\multimap B}_{(a,b)} & := & \lambda x.\ \Downarrow^{A}_{a}\ x; \Uparrow^{B}_{b} \\
\Uparrow^{A\oplus B}_{(0,a)} & := & \mathbf{inj}_{\ell}\ \Uparrow^{A}_{a} & \Uparrow^{A\oplus B}_{(1,b)} & := & \mathbf{inj}_{r}\ \Uparrow^{B}_{b} \\
\Uparrow^{A^{\ell}}_{(0,\star)} & := & [\,] & \Uparrow^{A^{\ell}}_{(1,(a,b))} & := & \Uparrow^{A}_{a} :: \Uparrow^{A^{\ell}}_{b}
\end{array}
$$

Table 6.2: Generator terms $\vdash_{\mathbb{L}} \Uparrow^{A}_{a} : A$

$$
\begin{array}{lcl}
\Downarrow^{\mathbf{1}}_{\star} & := & \lambda().() \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},0} & := & \lambda q.\mathbf{if}\ \mathbf{meas}\ q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},1} & := & \lambda q.\mathbf{if}\ \mathbf{meas}\ q\ \mathbf{then}\ ()\ \mathbf{else}\ \bot \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},2} & := & \lambda q.\mathbf{if}\ \mathbf{meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\mathbf{qb},3} & := & \lambda q.\mathbf{if}\ \mathbf{meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{\mathbf{i}}{\sqrt{2}} \\ -\frac{\mathbf{i}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\\
\Downarrow^{A\otimes B}_{(a,b)} & := & \lambda(x\otimes y).\ \Downarrow^{A}_{a}\ x\ ; \Downarrow^{B}_{b}\ y \\
\Downarrow^{A\multimap B}_{(a,b)} & := & \lambda f.\mathbf{let}\ x = f\ \Uparrow^{A}_{a}\ \mathbf{in}\ \Downarrow^{B}_{b}\ x \\
\Downarrow^{A\oplus B}_{(0,a)} & := & \lambda x.\delta\ (x,\ y.\ \Downarrow^{A}_{a}\ y,\ z.\bot) \\
\Downarrow^{A\oplus B}_{(1,b)} & := & \lambda x.\delta\ (x,\ y.\bot,\ z.\ \Downarrow^{B}_{b}\ z) \\
\Downarrow^{A^{\ell}}_{(0,\star)} & := & \lambda \ell.\mathbf{match}\ \ell\ \mathbf{with}\ ([\,] \mapsto () \mid x :: y \mapsto \bot) \\
\Downarrow^{A^{\ell}}_{(1,(a,b))} & := & \lambda \ell.\mathbf{match}\ \ell\ \mathbf{with}\ \left([\,] \mapsto \bot \;\middle|\; x :: y \mapsto \Downarrow^{A}_{a}\ x\ ; \Downarrow^{A^{\ell}}_{b}\ y\right)
\end{array}
$$

Table 6.3: Test terms $\vdash_{\mathbb{L}} \Downarrow^{A}_{a} : A \multimap \mathbf{1}$

$$O[t] = \mathbf{let}\ x^A = \Uparrow^A_{r_A(x_A)|i}\ \mathbf{in}\ \Downarrow^B_{r_B(x_B)|j}\ t$$

$$\mathcal{Q}_{[\![O[y]]\!]}(\{\star\},\{\star\}) = T_i^{\mathcal{H}_{[\![A]\!](x_A)}} \circ \mathcal{Q}_{[\![t]\!]}(x_A, x_B) \circ G_j^{\mathcal{H}_{[\![B]\!]}(x_B)}$$

where G and T are the morphisms of Proposition 2.1.18. We can deduce this property from the following lemma.

Lemma 6.3.4 (Semantics of Tests and Generators)**.** *For A a type and $(r_A(x_A)|i) \in |A|_e$:*

$$x_A \neq y \implies \mathcal{Q}_{\left[\!\left[\Uparrow^A_{r_A(x_A)|i}\ x\right]\!\right]}(\{\star\}, y) = 0 \text{ and } \mathcal{Q}_{\left[\!\left[\Downarrow^A_{r_A(x_A)|i}\ x\right]\!\right]}(y, \{\star\}) = 0$$

And moreover:

$$\begin{aligned}
\mathcal{Q}_{\left[\!\left[\Uparrow^A_{r_A(x_A)|i}\ x\right]\!\right]}(\{\star\}, x_A) &= G_i^{\mathcal{H}_{[\![A]\!](x_A)}} \in \mathbf{CPM}(\mathbf{1}, \mathcal{H}_{[\![A]\!]}(x_A)) \\
\mathcal{Q}_{\left[\!\left[\Downarrow^A_{r_A(x_A)|i}\ x\right]\!\right]}(x_A, \{\star\}) &= T_i^{\mathcal{H}_{[\![A]\!](x_A)}} \in \mathbf{CPM}(\mathcal{H}_{[\![A]\!]}(x_A), \mathbf{1})
\end{aligned}$$

Proof. The terms for $A = \mathbf{qubit}$ have been created such that it holds for them. We then simply proceed by induction on the type, using the compact closure of **CPM** for the function case. □

We then have all the tools to prove the reverse implication of the full abstraction, and conclude with the full abstraction theorem for LQΛ.

Lemma 6.3.5 (Characterisation by Tests and Generators)**.** *We define the set of observers $O_{v:A\vdash_{\mathbb{L}} B}$ as*

$$O_{v:A\vdash_{\mathbb{L}} B} = \left\{ \mathbf{let}\ v^A = \Uparrow^A_{a|i}\ \mathbf{in}\ \Downarrow^B_{b|j}\ _ \;\middle|\; (a|i) \in |A|_e, (b|j) \in |B|_e \right\}$$

For every pair of terms $x : A \vdash_{\mathbb{L}} t : B$ and $x : A \vdash_{\mathbb{L}} s : B$, we have:

$$\begin{gathered}
\forall \mathcal{O}[_] \in O_{x:A\vdash_{\mathbb{L}} B}, \mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow) \\
\implies \\
[\![t]\!] \equiv [\![s]\!]
\end{gathered}$$

Proof. Using Lemma 6.3.4, we have immediately that for $\mathcal{O}[_] = \mathbf{let}\ v^A = \Uparrow^A_{r_A(x_A)|i}$ $\mathbf{in}\ \Downarrow^B_{r_B(x_B)|j}\ _$, we have $\mathcal{Q}_{[\![\mathcal{O}[t]]\!]}(\star,\star) = T_j^{\mathcal{H}_{[\![B]\!](x_B)}} \circ \mathcal{Q}_{[\![t]\!]}(x_A, x_B) \circ G_i^{\mathcal{H}_{[\![A]\!](x_A)}}$. So if for all observers we have $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow)$, using soundness and adequacy we have for all $(r_A(x_A)|i) \in |A|_e$ and all $(r_B(x_B)|j) \in |B|_e$:

$$T_j^{\mathcal{H}_{[\![B]\!](x_B)}} \circ \mathcal{Q}_{[\![t]\!]}(x_A, x_B) \circ G_i^{\mathcal{H}_{[\![A]\!](x_A)}} = T_j^{\mathcal{H}_{[\![B]\!](x_B)}} \circ \mathcal{Q}_{[\![s]\!]}(x_A, x_B) \circ G_i^{\mathcal{H}_{[\![A]\!](x_A)}}$$

Using Theorem 6.2.7, we know that $\mathcal{Q}_{[\![t]\!]}$ and $\mathcal{Q}_{[\![s]\!]}$ are in **CPM**, so using Proposi-

tion 2.1.18 we deduce that for all $x_A \in \mathcal{E}(\llbracket A \rrbracket)$ and $x_B \in \mathcal{E}(\llbracket x_B \rrbracket)$ we have:

$$\mathcal{Q}_{\llbracket t \rrbracket}(x_A, x_B) = \mathcal{Q}_{\llbracket s \rrbracket}(x_A, x_B) \qquad \square$$

Theorem 6.3.6 (≡-Full Abstraction)**.** *For every term $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma \vdash_{\mathbb{L}} s : A$, we have*

$$\llbracket t \rrbracket \equiv \llbracket s \rrbracket \iff t =_{\text{obs}} s$$

Proof. The direct implication is exactly Corollary 6.2.12. We now assume $t =_{\text{obs}} s$. We write $P = \bigotimes_{(x_i:A_i)\in\Gamma} A_i$.

We consider $t' = \textbf{let } v = \bigotimes_{(v_i:A_i)\in\Gamma} v_i \textbf{ in } s$ and $s' = \textbf{let } v = \bigotimes_{(v_i:A_i)\in\Gamma} v_i \textbf{ in } s$. It follows that $t' =_{\text{obs}} s'$. In particular, for every $\mathcal{O}[_] \in O_{v:P_{\mathbb{L}}\, A}$, we have $\mathbb{P}(\mathcal{O}[t'] \Downarrow) = \mathbb{P}(\mathcal{O}[s'] \Downarrow)$. It follows from the previous lemma that $\llbracket t' \rrbracket \equiv \llbracket s' \rrbracket$. From the definition of the semantics, it follows immediately that $\llbracket t \rrbracket \equiv \llbracket s \rrbracket$. $\square$

6.3.3 Relational Collapse of LQΛ

Since both **QRel** and **QA** quotiented by ≡ are fully abstract models of LQΛ, it is reasonable to expect a relationship between those two. In this section, we write $\llbracket - \rrbracket_{\mathbf{QA}}$ for the games model of LQΛ, and use $\llbracket - \rrbracket_{\mathbf{QRel}}$ for the relational model of LQΛ. We will relate the two models through an intermediate category: the category of quantum relations on arenas.

Definition 6.3.7. *The category* **QARel** *has quantum arenas as objects, and quantum relations on arenas as morphisms, where a quantum relation on arenas from A to B is a function R with $\forall x_A \in \mathcal{E}(A), x_B \in \mathcal{E}(B), R(x_A, x_B) \in \overline{CPM}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_A))$. The composition is the relational composition*

$$(R' \circ R)(x_A, x_C) := \sum_{x_B \in \mathcal{E}(B)} R(x_C, x_B) \circ R(x_A, x_B)$$

The identity quantum relation on arenas is $\mathbf{id}_A^{\mathbf{QARel}}(x, y) = \mathbf{id}_{\mathcal{H}(x)}^{\overline{\mathbf{CPM}}}$ if $x = y$ and 0 otherwise.

A quantum relation on arenas is called finitary if all its quantum annotations are in the finitary fragment of $\overline{\mathbf{CPM}}$.

Proposition 6.3.8. *The category $(\mathbf{QARel}, \mathbf{QARel}, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ is a non-trivial distributive CFC with a bottom. We have an identity-on-objects functor from* **QA** *to* **QARel** *which preserves all the structure, and is given by*

$$\sigma \mapsto \mathcal{Q}_\sigma(-, -)$$

The image of this functor is included in the set of finitary maps of **QARel***, and this functor is ≡-faithful: two maps are exhaustively equivalent if and only if they have the same image by this functor.*

This proposition is essentially a corollary of Theorem 6.2.9. We note that while the objects of **QARel** are the same as **QA**, the maps are the same as **QRel**, so the additional structure $(\otimes, \multimap, \dots)$ behaves as in **QA** for objects and the same as in **QRel** for morphisms.

Theorem 6.3.9 (Factorisation)**.** *Up to isomorphism in* **QRel** *we have:*

$$\begin{array}{ccc} \mathrm{LQ}\Lambda & \xrightarrow{[\![-]\!]_{\mathbf{QRel}}} & \mathbf{QRel} \\ {\scriptstyle [\![-]\!]_{\mathbf{QA}}}\downarrow & & \uparrow{\scriptstyle \mathcal{E}} \\ \mathbf{QA} & \xrightarrow[\mathcal{Q}(-,-)]{} & \mathbf{QARel} \end{array}$$

Where $\mathcal{E}$ is the functor $f \in \mathbf{QARel}(A, B) \mapsto f \in \mathbf{QRel}(\mathcal{E}(A), \mathcal{E}(B))$. The isomorphism being used is $R_A \in \mathbf{QRel}(\mathcal{E}([\![A]\!]_{\mathbf{QA}}), [\![A]\!]_{\mathbf{QRel}})$ defined by $R_A(x, a) = \mathbf{id}^{\overline{\mathbf{CPM}}}_{\mathcal{H}_A(x)}$ if $a = r_A(x)$ and 0 otherwise. The proof of this theorem is direct: we make an induction over the typing derivations, relying on the fact that $\equiv$ is a congruence on **QA**. We note that this factorisation allows us to deduce the full abstraction of any of the two models from the other one. The category **QRel** is significantly bigger than the image of **QA** by the functor, containing a lot of weighted relations that have no computational meaning, describing "systems" that execute some behaviours with "probabilities" greater than one, or even infinite when the annotations are in the infinitary fragment of $\overline{\mathbf{CPM}}$. While not all strategies of **QA** come from QΛ, Theorem 5.3.15 ensures that they always have a computational meaning.

6.4 Affine Quantum Semantics

In this section, we tweak the proof of full abstraction of **QA** for LQΛ into a proof of full abstraction of $\mathbf{QA}^a$ for AQΛ.

While we do not define a relational model for AQΛ, this does not prevent us from defining a variation to the web of types more adapted to AQΛ, and to extend the bijection r_A and the test terms to the affine web. We sum up the differences in Table 6.4

Using exactly the same reasoning and intermediate lemmas as in the strictly linear case, we obtain

Theorem 6.4.1 ($\equiv$-Full Abstraction)**.** *For every term* $\Gamma \vdash_{\mathbb{A}} t : A$ *and* $\Gamma \vdash_{\mathbb{A}} s : A$*, we have*

$$[\![t]\!] \equiv [\![s]\!] \iff t =_{\mathrm{obs}} s$$

We also have a similar relational collapse with a category of quantum relations on arenas, though there is a small subtlety as we cannot discard the notion of thunkability in the affine case.

Definition 6.4.2. *The category* $\mathbf{QARel}^a$ *is the full subcategory of* **QARel** *restricted to only affine arenas. A quantum relation on arenas* $R \in \mathbf{QARel}^a(A, B)$ *is called thunkable if*

Strictly Linear Case	**Affine Case**
$\kappa_{[\![A\multimap B]\!]}(\{\lambda^+\}) := 1$	$\kappa_{[\![A\multimap B]\!]}(\{\lambda^+\}) := 0$
$\lvert A \multimap B\rvert := \lvert A\rvert \times \lvert B\rvert$	$\lvert A \multimap B\rvert := \{\lambda\} \sqcup (\lvert A\rvert \times \lvert B\rvert)$
$\mathcal{H}_{A\multimap B} : (a,b) \mapsto \mathcal{H}_A(a) \otimes \mathcal{H}_B(b)$	$\mathcal{H}_{A\multimap B} : \begin{cases} (a,b) \mapsto \mathcal{H}_A(a) \otimes \mathcal{H}_B(b) \\ \lambda \quad\quad \mapsto \mathbf{1} \end{cases}$
$r_{A\multimap B} : x_A \multimap x_B \mapsto (r_A(x_A), r_B(x_B))$	$r_{A\multimap B} : \begin{cases} x_A \multimap x_B \mapsto (r_A(x_A), r_B(x_B)) \\ \{\lambda^+\} \quad\quad \mapsto \lambda \end{cases}$
$\Uparrow^{A\multimap B}_{(a,b)} := \lambda x.\, \Downarrow^A_a \ x; \Uparrow^B_b$	$\Uparrow^{A\multimap B}_{(a,b)} := \lambda x.\, \Downarrow^A_a \ x; \Uparrow^B_b$
	$\Uparrow^{A\multimap B}_{\lambda} := \lambda x.\bot$
$\Downarrow^{A\multimap B}_{(a,b)} := \lambda f.\mathbf{let}\ x = f\ \Uparrow^A_a\ \mathbf{in}\ \Downarrow^B_b\ x$	$\Downarrow^{A\multimap B}_{(a,b)} := \lambda f.\mathbf{let}\ x = f\ \Uparrow^A_a\ \mathbf{in}\ \Downarrow^B_b\ x$
	$\Downarrow^{A\multimap B}_{\lambda} := \lambda f.()$

Table 6.4: Differences between the semantics of LQΛ and AQΛ

- *for every $a \in \min A$, there exists a unique $b \in \min B$, such that $R(\{a\},\{b\}) \neq 0$, and*
- *for every $a \in \min A$, $b \in \min B$ with $R(\{a\},\{b\}) \neq 0$, we have*

$$\mathbf{Tr}_{\mathcal{H}_B(\{b\})} \circ R(\{a\},\{b\}) = \mathbf{Tr}_{\mathcal{H}_A(\{a\})}$$

We write $\mathbf{QARel}^a_t$ for the subcategory of thunkable maps.

Proposition 6.4.3. *The category $(\mathbf{QARel}^a, \mathbf{QARel}^a_t, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ is an affine non-trivial distributive CFC with a bottom. We have an identity-on-objects functor from $\mathbf{QA}^a$ to $\mathbf{QARel}^a$ which preserves all the structure, and is given by*

$$\sigma \mapsto \mathcal{Q}_\sigma(-,-)$$

The image of this functor is included in the set of finitary maps of $\mathbf{QARel}^a$, and this functor is $\equiv$-faithful: two maps are exhaustively equivalent if and only if they have the same image by this functor.

If we had defined a relational model for AQΛ, we would expect a similar factorisation theorem

$$\begin{array}{ccc} \mathrm{AQ}\Lambda_! & \xrightarrow{[\![-]\!]_{\mathbf{QRel}^a}} & \mathbf{QRel}^a \\ {\scriptstyle [\![-]\!]_{\mathbf{QA}^a}}\downarrow & & \uparrow{\scriptstyle \mathcal{E}} \\ \mathbf{QA}^a & \xrightarrow[\mathcal{Q}(-,-)]{} & \mathbf{QARel}^a \end{array}$$

Part III

Fully Abstract Models for the Full Quantum λ-calculus

Overview of Part III

In this part, we tackle the issue of modelling replicable functions, *i.e.*, functions that can be used multiple times.

In the seventh chapter, we define the (full) quantum λ-calculus, which is the language studied in [PSV14]. We then look at the categorical structure required to model it.

In the eighth chapter, we present the category of quantum relations on $\sim$-arenas, which is a variation of the relational model presented in [PSV14]. This model was known to be sound and adequate, but full abstraction was an open question. The main difference between the category defined in [PSV14] and the model we define in this chapter is that we use quantum arenas as objects, instead of weighted sets (the webs). As such, this chapter starts with an introduction on how to represent the replication in arenas: the event structures with symmetry. We refer to [CCW19] for a more in-depth study of symmetry.

In the ninth chapter, we present our game model for the quantum λ-calculus. While similar models for probabilistic languages existed, this model is one of the main contribution of this thesis. We prove soundness and adequacy of this model, which was the main result of our work in [CdVW19].

In the tenth and last chapter, we present the proof of full abstraction of our game model, which induces the full abstraction of the relational model. The proof method is an adaptation of the method of [ETP14] on probabilistic coherence spaces to our quantum game models.

Chapter 7

The Quantum λ-calculus

7.1 The Quantum λ-calculus

7.1.1 Motivation

The language QΛ has a crucial restriction: variables and functions must be used linearly. This linearity restriction exists because of the non-duplicability of quantum data. Indeed, the function $\lambda().\mathbf{meas}\ q$ cannot be used multiple times, as it consumes the qubit q when executed. However, plenty of functions of QΛ do not have this problem, and could be allowed to be duplicated without contradicting the quantum no-cloning theorem. For example, the function $\lambda().\mathbf{new}\ \mathbf{ff}$ does not consume any quantum data, so we should be able to duplicate it at will. As a guiding example in this part, we will use the term $\mathrm{Map}(f, \ell)$ which applies the replicable function f to every element of the list ℓ:

$$\begin{array}{rcl} \mathrm{Map}(f,\ell) & := & \mathbf{let\ rec}\ m\ \ell' = \\ & & \quad \mathbf{match}\ \ell'\ \mathbf{with}\ \begin{array}{|lcl} [] & \mapsto & [] \\ x :: \ell'' & \mapsto & (f\ x) :: (m\ \ell'') \end{array} \\ & & \mathbf{in}\ m\ \ell \end{array}$$

We will type this term in the following way, writing $!(\mathbf{qubit} \multimap \mathbf{qubit})$ for the type of replicable functions on qubits.

$$f :!(\mathbf{qubit} \multimap \mathbf{qubit}), \ell : \mathbf{qubit}^\ell \vdash_{\mathbb{L}} \mathrm{Map}(f,\ell) : \mathbf{qubit}^\ell$$

7.1.2 Syntax and Typing System

In this section, we define the full quantum λ-calculus, a language for which a sound and adequate model was already known, but no model was known to be fully abstract before our work. Up to some small syntactical changes, this is the same language as in [PSV14]. This language combines quantum primitives with non-linear behaviour, meaning that functions

that cannot be used multiple times due to physical impossibility are restricted to be linear by the typing system, while functions that are physically replicable will be able to be typed with a ! indicating they can be used as many times as one wishes to.

Similarly to previously defined languages, the language we consider here has two variations. We write $\mathrm{LQ}\Lambda_!$ for the language where functions can either be strictly linear or non-linear, and $\mathrm{AQ}\Lambda_!$ for the language where functions can either be affine or non-linear. Finally, we write $\mathrm{Q}\Lambda_!$ when we refer indifferently to either of the two variants. For typing judgements, we use $\vdash_{\mathbb{L}}$, $\vdash_{\mathbb{A}}$ and $\vdash$ respectively. Due to the similarity of $\mathrm{Q}\Lambda_!$ with $\mathrm{Q}\Lambda$, a lot of the content of this section is redundant with the content of previous parts, but we find it more practical to have all the definitions at the same place. Types of $\mathrm{Q}\Lambda_!$ are

$$A, B, \cdots ::= \mathbf{1} \mid \mathbf{qubit} \mid A \oplus B \mid A \otimes B \mid A \multimap B \mid !(A \multimap B) \mid A^{\ell}$$

The type $!(A \multimap B)$ represents non-linear functions, in contrast to functions of type $A \multimap B$ which are linear in $\mathrm{LQ}\Lambda_!$ and affine in $\mathrm{AQ}\Lambda_!$. As frequently done in call-by-value languages, we choose to only replicate functions. Whenever needed, terms of type $!A$ can usually be simulated through terms of the type $!(\mathbf{1} \multimap A)$. While we continue to use $A, B, \ldots$ to name types, we will often use $F, G, \ldots$ to range over types of the form $A \multimap B$. The terms of the language $\mathrm{Q}\Lambda_!$ are simply the terms of $\mathrm{Q}\Lambda$ together with a recursion operator:

$$\begin{array}{rcllllllll} t, s, \ldots & ::= & x & \mid & \lambda x^A.t & \mid & t\ s & \mid & () & \mid & t; s & \mid & \bot_V^A \\ & \mid & \mathbf{inj}_\ell\ t & \mid & \mathbf{inj}_r\ t & \mid & \delta\ (t,\ s_1,\ s_2) & \mid & t \otimes s & \mid & \mathbf{let}\ x^A \otimes y^B = t\ \mathbf{in}\ s & & \\ & \mid & \mathbf{meas}\ t & \mid & \mathbf{new}\ t & \mid & \mathbf{U}\ t & \mid & \mathbf{fold}\ t & \mid & \mathbf{unfold}\ t & & \\ & \mid & \mathbf{let\ rec}\ f^{!(A \multimap B)}\ x^A = t\ \mathbf{in}\ s & & & & & & & & & & \end{array}$$

The values of the language are the following:

$$v, w, \cdots ::= x \mid \lambda x^A.t \mid () \mid v \otimes w \mid \mathbf{inj}_\ell\ v \mid \mathbf{inj}_r\ v \mid \mathbf{fold}\ v$$

We note that $\bot^A$ and the infinitely looping recursion $\mathbf{let\ rec}\ f^{!(\mathbf{1} \multimap A)}\ x^{\mathbf{1}} = f\ x\ \mathbf{in}\ f\ ()$ are observationally equivalent[1]. As previously, we will often omit the typing annotations in terms, *e.g.*, the ones of x^A and $f^{!(A \multimap B)}$. In Table 7.1, we recall the syntactic sugar defined in previous parts, together with the definition of bounded recursions operator $\mathbf{let\ rec}\ [\mathbf{n}]$ which are forced to diverge after $n \in \mathbb{N}$ calls.

$$\begin{array}{lcl} \mathbf{let\ rec}\ [0]\ f\ x = t\ \mathbf{in}\ s & := & \bot_{\mathrm{FV}(s) \setminus \{f\}} \\ \mathbf{let\ rec}\ [n]\ f\ x = t\ \mathbf{in}\ s & := & \mathbf{let}\ f\ x = (\mathbf{let\ rec}\ [n-1]\ f\ x = t\ \mathbf{in}\ t)\ \mathbf{in}\ s \end{array}$$

In Tables 7.2 and 7.3, we list the typing rules of the language, writing $!\Omega$ for the sequence $x_1 :!F_1, \ldots, x_n :!F_n$ whenever $\Omega = x_1 : F_1, \ldots, x_n : F_n$. This typing system allows us to type $\mathrm{Map}(f, \ell)$ as follows:

$$f :!(\mathbf{qubit} \multimap \mathbf{qubit}), \ell : \mathbf{qubit}^{\ell} \vdash_{\mathbb{L}} \mathrm{Map}(f, \ell) : \mathbf{qubit}^{\ell}$$

[1] See Section 7.1.4 for the definition of observational equivalence in $\mathrm{Q}\Lambda_!$.

The typing rules of $\mathrm{Q}\Lambda_!$ are the same as the ones of $\mathrm{Q}\Lambda$, with the following exceptions. Firstly, all the rules that merge two contexts Γ and Δ into Γ, Δ now merge the contexts $!\Omega, \Gamma$ and $!\Omega, \Delta$ into $!\Omega, \Gamma, \Delta$. Indeed, replicable variables can be used multiple times in the term, so can appear in multiple branches of the typing derivation. Secondly, the axiom rule is restricted to types that are *linear*, *i.e.*, not of the form $!F$, as a special axiom-dereliction rule is added for types of the form $!F$. Thirdly, we have four new typing rules which we detail here.

$$\frac{!\Omega, f :!(A \multimap B), x : A \vdash t : B \quad !\Omega, \Gamma, f :!(A \multimap B) \vdash s : C}{!\Omega, \Gamma \vdash \mathbf{let\ rec}\ f^{!(A \multimap B)}\ x^A = t\ \mathbf{in}\ s : C}$$

This rule is the recursion rule. A notable point of this rule is the $!\Omega$ on the left hand side branch, which allows us to use in recursive definitions some external functions, but only non-linear ones. The right hand side branch can have both linear and non-linear variables in its context.

$$\frac{}{x :!F \vdash x : F}$$

This rule is the axiom-dereliction rule. It means that a non-linear function can be used linearly, *i.e.*, a function that can be used "as many time as we want" can be used "once". We note that even though the regular axiom rule cannot be used on non-linear functions, so we still have $x :!F \vdash x :!F$ using the axiom-dereliction rule followed by the promotion rule presented just below.

$$\frac{!\Omega \vdash v : A \multimap B \quad v \text{ value}}{!\Omega \vdash v :!(A \multimap B)}$$

This rule is the promotion rule. If says that if a linear function only uses non-linear variables, then we can replicate this function and obtain a non-linear function. Since the promotion rule leaves no syntactic trace in terms, we do not have uniqueness of typing. Indeed, we could have $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$ with A and B differing because of the presence or absence of one or more ! inside them.

$$\frac{\Gamma \vdash t : B \quad x \notin \mathrm{FV}(t)}{\Gamma, x :!F \vdash t : B}$$

This rule is the exponential weakening rule. In $\mathrm{AQ}\Lambda_!$, it is redundant with the weakening rule, but in $\mathrm{LQ}\Lambda_!$ it states that every non-linear function can be discarded, *i.e.*, a function that can be used "as many time as we want" can be used "zero times".

Function type

$$\begin{array}{rcl} \textbf{let } x = t \textbf{ in } s & := & (\lambda x.s)\ t \\ \textbf{let } f\ x = t \textbf{ in } s & := & (\lambda f.s)\ (\lambda x.t) \end{array}$$

Unit type

$$\begin{array}{rcl} \lambda().t & := & \lambda x.x\ ;\ t \end{array}$$

Tensor type

$$\begin{array}{rcl} A_1 \otimes \ldots \otimes A_n & := & (A_1 \otimes \ldots) \otimes A_n \\ x_1 \otimes \ldots \otimes x_n & := & (x_1 \otimes \ldots) \otimes x_n \\ \textbf{let } x_1 \otimes \ldots \otimes x_n = t \textbf{ in } s & := & \begin{cases} \textbf{let } y \otimes x_n = t \textbf{ in} \\ \textbf{let } x_1 \otimes \ldots = y \textbf{ in} \\ s \end{cases} \\ \lambda(x_1 \otimes \ldots \otimes x_n).t & := & \lambda z.\textbf{let } x_1 \otimes \ldots \otimes x_n = z \textbf{ in } t \end{array}$$

Sum type

$$\begin{array}{rcl} \textbf{bit} & := & \mathbf{1} \oplus \mathbf{1} \\ \textbf{ff} & := & \textbf{inj}_\ell\ () \\ \textbf{tt} & := & \textbf{inj}_r\ () \\ \textbf{if } t \textbf{ then } s_t \textbf{ else } s_f & := & \delta\ (t,\ x.x; s_f,\ y.y; s_t) \\ \textbf{match } t \textbf{ with } \begin{array}{l} \mid \textbf{inj}_\ell\ x \ \mapsto\ s_\ell \\ \mid \textbf{inj}_r\ y \ \mapsto\ s_r \end{array} & := & \delta\ (t,\ x.s_\ell,\ y.s_r) \end{array}$$

List type

$$\begin{array}{rcl} [\,] & := & \textbf{fold}\ (\textbf{inj}_\ell\ ()) \\ t :: s & := & \textbf{fold}\ (\textbf{inj}_r\ (t \otimes s)) \\ \textbf{match } t \textbf{ with } \begin{array}{l} \mid [\,] \ \mapsto\ s_1 \\ \mid x :: y \ \mapsto\ s_2 \end{array} & := & \delta\ (\textbf{unfold } t,\ z.z; s_1,\ z'.\textbf{let } x \otimes y = z' \textbf{ in } s_2) \end{array}$$

Quantum primitives

$$\begin{array}{rcll} p \cdot t + (1-p) \cdot s & := & \begin{cases} \textbf{if meas } \left(\begin{pmatrix} \sqrt{1-p} & \sqrt{1-p} \\ \sqrt{p} & -\sqrt{p} \end{pmatrix}\ (\textbf{new ff}) \right) \\ \textbf{then } t \\ \textbf{else } s \end{cases} & \\ p \cdot t & := & p \cdot t + (1-p) \cdot \bot & \\ \sum_{i=1}^n p_i \cdot t_i & := & p_1 \cdot t_1 + \sum_{i=2}^n \frac{p_i}{1-p_1} \cdot t_i & \text{with } \frac{0}{0} = 0 \end{array}$$

Recursion

$$\begin{array}{rcl} \textbf{let rec } [0]\ f\ x = t \textbf{ in } s & := & \bot_{\mathrm{FV}(s) \setminus \{f\}} \\ \textbf{let rec } [n]\ f\ x = t \textbf{ in } s & := & \textbf{let } f\ x = (\textbf{let rec } [n-1]\ f\ x = t \textbf{ in } t) \textbf{ in } s \end{array}$$

Table 7.1: Syntactic Sugar for $\mathrm{Q}\Lambda_!$

Structural rules

$$\frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, y : B, x : A, \Delta \vdash t : C}\ \text{permutation} \qquad \frac{\Gamma \vdash_{\mathbb{A}} t : B \qquad x \notin \mathrm{FV}(t)}{\Gamma, x : A \vdash_{\mathbb{A}} t : B}\ \begin{array}{l}(\mathrm{A}\Lambda\ \text{only})\\ \text{weakening}\end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash \bot^{A}_{x_1,\ldots,x_n} : A}\ \text{bottom}$$

λ-calculus

$$\frac{A\ \text{linear}}{x : A \vdash x : A}\ \text{axiom} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^{A}.t : A \multimap B}\ \text{abstraction}$$

$$\frac{!\Omega, \Gamma \vdash t : A \multimap B \qquad !\Omega, \Delta \vdash s : A}{!\Omega, \Gamma, \Delta \vdash t\ s : B}\ \text{application}$$

Unit type

$$\frac{}{\vdash () : \mathbf{1}}\ \text{skip} \qquad \frac{!\Omega, \Gamma \vdash t : \mathbf{1} \qquad !\Omega, \Delta \vdash s : A}{!\Omega, \Gamma, \Delta \vdash t\ ;\ s : A}\ \text{sequence}$$

Tensor type

$$\frac{!\Omega, \Gamma \vdash t : A \qquad !\Omega, \Delta \vdash s : B}{!\Omega, \Gamma, \Delta \vdash t \otimes s : A \otimes B}\ \text{pair}$$

$$\frac{!\Omega, \Gamma \vdash t : A \otimes B \qquad x : A, y : B, !\Omega, \Delta \vdash s : C}{!\Omega, \Gamma, \Delta \vdash \mathbf{let}\ x \otimes y = t\ \mathbf{in}\ s : C}\ \text{let-pair}$$

Sum type

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathbf{inj}_{\ell}^{A \oplus B}\ t : A \oplus B}\ \text{left-injection} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathbf{inj}_{r}^{A \oplus B}\ t : A \oplus B}\ \text{right-injection}$$

$$\frac{!\Omega, \Gamma \vdash t : A \oplus B \qquad x : A, !\Omega, \Delta \vdash s_1 : C \qquad y : B, !\Omega, \Delta \vdash s_2 : C}{!\Omega, \Gamma, \Delta \vdash \delta\ (t,\ x^{A}.s_1,\ y^{B}.s_2) : C}\ \text{case}$$

List type

$$\frac{\Gamma \vdash t : A^{\ell}}{\Gamma \vdash \mathbf{unfold}\ t : \mathbf{1} \oplus (A \otimes A^{\ell})}\ \text{unfold} \qquad \frac{\Gamma \vdash t : \mathbf{1} \oplus (A \otimes A^{\ell})}{\Gamma \vdash \mathbf{fold}\ t : A^{\ell}}\ \text{fold}$$

Table 7.2: Typing Rules for QΛ$_!$, part 1

Quantum primitives

$$\frac{\Gamma \vdash t : \mathbf{bit}}{\Gamma \vdash \mathbf{new}\ t : \mathbf{qubit}}\ \text{new} \qquad \frac{\Gamma \vdash t : \mathbf{qubit}}{\Gamma \vdash \mathbf{meas}\ t : \mathbf{bit}}\ \text{meas}$$

$$\frac{\Gamma \vdash t : \mathbf{qubit}^{\otimes n} \qquad U \text{ of arity } n}{\Gamma \vdash \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}}\ \text{unitary}$$

Exponential rules

$$\frac{}{x :!F \vdash x : F}\ \text{axiom-dereliction} \qquad \frac{\Gamma \vdash t : B \qquad x \notin \mathrm{FV}(t)}{\Gamma, x :!F \vdash t : B}\ !\text{-weakening}$$

$$\frac{!\Omega \vdash v : A \multimap B \qquad v \text{ value}}{!\Omega \vdash v :!(A \multimap B)}\ \text{promotion}$$

$$\frac{!\Omega, f :!(A \multimap B), x : A \vdash t : B \qquad !\Omega, \Gamma, f :!(A \multimap B) \vdash s : C}{!\Omega, \Gamma \vdash \mathbf{let\ rec}\ f^{!(A \multimap B)}\ x^{A} = t\ \mathbf{in}\ s : C}\ \text{recursion}$$

Table 7.3: Typing Rules for QΛ!, part 2

7.1.3 Operational Semantics

Similarly to QΛ, we have two reduction systems: one at the level of terms that takes care of all the reductions that are "structural", and one at the level of quantum closures which inherits the "structural" reductions from the first one and adds some "quantum" reductions. We redefine quantum closures as follows. Those reduction systems are nearly identical to the reduction systems of QΛ in Table 3.1. We use the same definition for closures, which we recall here.

Definition 7.1.1. *A simple (quantum) closure is a triple $[q, \ell, t]$ where:*

- *t is a term.*
- *q is the quantum store, $q \in \mathfrak{Q}^{\otimes n}$ for some $n \in \mathbb{N}$.*
- *ℓ is a sequence of n variables written $|x_1 \dots x_n\rangle$. It is an ordering of all the free variables of t, and can be seen as a function which to each free variable of the term indicate where its value is stored.*

We say it is terminal *if it is of the form $[q, \ell, v]$ with v a value, or $[q, \ell, \bot]$. We write $\vdash [q, \ell, t] : A$ whenever*

$$x_1 : \mathbf{qubit}, \dots, x_n, \mathbf{qubit} \vdash t : A$$

$$
\begin{array}{lcll}
() \ ; \ t & \to & t & \\
(\lambda x.t) \ v & \to & t\{x \leftarrow v\} & \\
\textbf{let rec } f \ x = t \textbf{ in } s & \to & s\left\{\begin{array}{c} f \leftarrow \lambda x. \\ \textbf{let rec } f \ x = t \textbf{ in } t \end{array}\right\} & \\
\textbf{let } x \otimes y = v \otimes w \textbf{ in } t & \to & t\{x \leftarrow v, y \leftarrow w\} & \\
\delta \ (\mathbf{inj}_\ell \ v, \ x.t, \ y.s) & \to & t\{x \leftarrow v\} & \\
\delta \ (\mathbf{inj}_r \ v, \ x.t, \ y.s) & \to & s\{y \leftarrow v\} & \\
\textbf{unfold } (\textbf{fold } v) & \to & v & \\
E[t] & \to & E[s] & \text{whenever} \quad t \to s \\
E[\bot_V] & \to & \bot_{\mathrm{FV}(E[\bot_V])} & \text{whenever} \quad E[_] \neq _
\end{array}
$$

$$
\begin{array}{lcllllll}
\text{with } E[_] & ::= & _ & | & E[_] \ ; \ t & & & \\
& | & E[_] \ t & | & v \ E[_] & & & \\
& | & E[_] \otimes t & | & v \otimes E[_] & | & \textbf{let } x \otimes y = E[_] \textbf{ in } t & \\
& | & \mathbf{inj}_\ell \ E[_] & | & \mathbf{inj}_r \ E[_] & | & \delta \ (E[_], \ x.t_1, \ y.t_2) &
\end{array}
$$

Table 7.4: Reduction Rules for QΛ! terms

$$
\frac{t_0 \to t_1}{[q, \ell, E[t_0]] \to [q, \ell, E[t_1]]}
$$

$$
\frac{}{[q, |x_1 \ldots x_n\rangle, E[\textbf{new ff}]] \to [q \otimes |0\rangle, |x_1 \ldots x_{n+1}\rangle, E[x_{n+1}]]}
$$

$$
\frac{}{[q, |x_1 \ldots x_n\rangle, E[\textbf{new tt}]] \to [q \otimes |1\rangle, |x_1 \ldots x_{n+1}\rangle, E[x_{n+1}]]}
$$

$$
\frac{k\text{-}\mathbf{meas}^{\mathbf{Hilb}}(q) = (q_0, q_1)}{[q, |x_1 \ldots x_n\rangle, E[\textbf{meas } x_k]] \to \begin{array}{lll} & ||q_0||^2 & \left[\frac{q_0}{||q_0||}, |x_1 \ldots x_{k-1} x_{k+1} \ldots x_n\rangle, E[\textbf{ff}]\right] \\ + & ||q_1||^2 & \left[\frac{q_1}{||q_1||}, |x_1 \ldots x_{k-1} x_{k+1} \ldots x_n\rangle, E[\textbf{tt}]\right] \end{array}}
$$

$$
\frac{\hat{\sigma} \circ \left(U \otimes \mathfrak{Q}^{\otimes n-k}\right) \circ \hat{\sigma}^{-1}(q) = q' \qquad \hat{\sigma} \text{ permutation map induced by } \sigma \in \mathfrak{S}_n}{[q, |x_1 \ldots x_n\rangle, E[\mathbf{U} \ (x_{\sigma(1)} \otimes \ldots \otimes x_{\sigma(k)})]] \to [q', |x_1 \ldots x_n\rangle, E[x_{\sigma(1)} \otimes \ldots \otimes x_{\sigma(k)}]]}
$$

$$
\frac{c \to c' \quad d \text{ terminal} \quad 0 < p \leq 1}{pc + (1-p)d \to pc' + (1-p)d} \qquad \frac{c \to c' \quad d \to d' \qquad 0 \leq p \leq 1}{pc + (1-p)d \to pc' + (1-p)d'}
$$

Table 7.5: Reduction Rules for QΛ! closures

Definition 7.1.2. *A (quantum) closure c is a discrete probability distribution of simple closures, written as a formal sum $\sum_i p_i c_i$ with c_i simple closures, $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. We say it is terminal if all the c_i are terminal. We write $\vdash \sum_i p_i c_i : A$ if $\vdash c_i : A$ for every simple closure c_i.*

We describe the reduction rules for $\mathrm{Q}\Lambda_!$ in Tables 7.4 and 7.5. We note that they refer to primitives of the category **Hilb** defined in Section 2.3.2. The reduction rules are the same as the ones for $\mathrm{Q}\Lambda$, with the following exception:

- There is an additional rule for recursion.
- In substitutions $t\{x \leftarrow v\}$, the variable x might appear more than once in t,

Proposition 7.1.3. *The following properties hold*

Subject Reduction *For every term $\Gamma \vdash t : A$, if $t \to s$ then $\Gamma \vdash s : A$. Similarly, for every closure $\vdash c : A$, if $c \to d$ then $\vdash d : A$.*

Determinism *For any term t, there exists at most one term s such that $t \to s$. Similarly for any closure c, there exists at most one closure d such that $c \to d$.*

Partial Normalisation *For any term t which does not contain* **let rec** *, there is no infinite sequence $t \to t_1 \to t_2 \to \ldots$. Similarly, for any closure c which does not contain* **let rec***, there is no infinite sequence $c \to c_1 \to c_2 \to \ldots$.*

Progress *For any closed term $\vdash t : A$, either t is a value, or $\bot$, or there exists a term s such that $t \to s$. Similarly for any closure c, either c is terminal, or there exists a closure d such that $c \to d$.*

Contrary to $\mathrm{Q}\Lambda$, subject reduction is non-trivial, and looking for example at the β-reduction $(\lambda x^A.t)\ v \to t\{x \leftarrow v\}$, one must distinguish the case where A is linear from the case where $A = !F$.

- Whenever A is linear, we obtain a typing derivation for $t\{x \leftarrow v\}$ by considering a typing derivation for t and replacing the axiom for x by a typing derivation for v.
- Whenever $A = !F$, we note that every typing derivation for $v : !F$ has a promotion rule at the root, so we can deduce from them the typing derivations for $v : F$. We obtain a typing derivation for $t\{x \leftarrow v\}$ by taking a typing derivation for t and replacing each of the axiom-derelictions for x by a typing derivation for $v : F$.

We recall that the determinism of quantum reductions is at the level of closures, *i.e.*, probability distributions of simple closures. The reduction system describes non-deterministic behaviours by encapsulating them in a probability distribution. The partial normalisation is also non-trivial to prove, and is proved in two steps: (1) we prove that the reduction

system on terms satisfies partial normalisation using the same proof as for the normalisation of the simply-typed λ-calculus (2) we deduce that the reduction system on quantum closures satisfies partial normalisation by noting that we reduce all the terms of a sum $\sum_i p_i c_i$ at once (except the terminal ones that cannot be reduced).

7.1.4 Convergence and Observational Equivalence

We extend the notions of convergence and observational equivalence to $\mathrm{Q}\Lambda_!$. We also state the approximation lemma, which will be key in proving that the denotational semantics for $\mathrm{Q}\Lambda_!$ are adequate.

Definition 7.1.4 (Convergence)**.** *For c a closure and v a value, we define the probability that c converges to v, written $\mathbb{P}(c \Downarrow v)$, as the supremum of the $p \in [0,1]$ such that $p = \sum_{i=1}^n p_i$ and $c \to^* \sum_{i=1}^n p_i[q_i, \ell_i, v] + (1-p)c'$ with c' any closure.*

By partial normalisation, if c does not contain **let rec**, then the supremum is reached. Whenever $\vdash c : \mathbf{1}$, we write $\mathbb{P}(c \Downarrow)$ for $\mathbb{P}(c \Downarrow ())$. Whenever $\vdash t : \mathbf{1}$, we write $\mathbb{P}(t \Downarrow)$ for $\mathbb{P}([\varnothing, \varnothing, t] \Downarrow)$. If c does contain **let rec**, the supremum might not be reached, which can be problematic to prove adequacy results. To take care about closures containing **let rec**, we will need the following lemma.

Lemma 7.1.5 (Approximation Lemma)**.** *For $\vdash c : \mathbf{1}$ a closure, if we write c_n for the closure c where every* **let rec** *has been replaced by a* **let rec [n]***, then $\mathbb{P}(c \Downarrow) = \sup_n \mathbb{P}(c_n \Downarrow)$.*

Proof. We start by noting that every reduction reduces by at most one the k of every **let rec [k]** in the closure, and that as long as we did not reach 0, the behaviour of **let rec [k]** and **let rec** are the same. From this it follows that if $c \to^k \sum_{i=1}^n p_i[q_i, \ell_i, ()] + (1-p)c'$, then we have $c_k \to^k \sum_{i=1}^n p_i[q_i, \ell_i, ()] + (1-p)d$ for some closure d. Since suprema commute, we have the expected result. □

An observation context $\mathcal{O}[_]$ for $\Gamma \vdash A$ with Γ a typing context and A a type, is a term with a unique hole $\mathcal{O}[_]$ such that for every $\Gamma \vdash t : A$, we have $\vdash t : \mathbf{1}$.

We say that two terms $\Gamma \vdash t_1 : A$ and $\Gamma \vdash t_2 : A$ are observationally equivalent, and we write $t_1 =_{\mathrm{obs}}^{\Gamma \vdash A} t_2$, if for every observation context $\mathcal{O}[_]$ for $\Gamma \vdash A$, we have

$$\mathbb{P}(\mathcal{O}[t_1] \Downarrow) = \mathbb{P}(\mathcal{O}[t_2] \Downarrow)$$

We will often keep the annotation $\Gamma \vdash A$ implicit.

7.2 Categorical Pre-Model for the Quantum λ-Calculus

In this section, we provide the general shape of a semantics for the quantum λ-calculus, which we will specialise later into the quantum relational model and the quantum game

model for LQΛ! and AQΛ!. We do not aim for a perfect categorical model that captures all the properties of the language, the goal of this section is only to factor out some of the definitions and propositions.

In particular, we will not study in this section the semantics of quantum closures and their reductions, as the relational model and the game models significantly differ in their representation of weighted sums of closures. As another example, we expect the equation for list to be $A^\ell = \mathbf{1} \oplus (A \otimes A^\ell)$ since this is the equation both models satisfy, even though asking for $A^\ell \cong \mathbf{1} \oplus (A \otimes A^\ell)$ would be enough.

7.2.1 The Categorical Pre-Model

As for Λ, we assume we have a non-trivial distributive CFC with a bottom $(\mathcal{C}, \mathcal{V}, J, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0}, \bot)$. When considering the affine language, we additionally assume the CFC is affine. On top of this CFC, we require the following structure:

- For every object A, an object A^ℓ such that $A^\ell = \mathbf{1} \oplus (A \otimes A^\ell)$.
- An object $\mathbf{qubit}$ together with two morphisms $\mathbf{meas}^{\mathcal{C}} \in \mathcal{C}(\mathbf{qubit}, \mathbf{bit})$, $\mathbf{new}^{\mathcal{C}} \in \mathcal{C}(\mathbf{bit}, \mathbf{qubit})$ and a morphism $\mathbf{U}^{\mathcal{C}} \in \mathcal{C}(\mathbf{qubit}^{\otimes n}, \mathbf{qubit}^{\otimes n})$ for every unitary U of arity n. We do not expect them to satisfy any particular axiom.
- A full sub-SMC $(\mathcal{F}, \otimes, \mathbf{1})$ of $(\mathcal{V}, \otimes, \mathbf{1})$, which contains at least all the objects of the form $A \multimap B$ for $A, B \in \mathcal{V}$.
- A linear exponential comonad $(!, \epsilon, \delta, \mathrm{w}, \mathrm{c}, \mathrm{m}, \mathrm{m}_{\mathbf{1}})$ on $\mathcal{F}$, as defined in Section 1.2.4.
- A recursor $\mathbf{Y} : \mathcal{F}(W \otimes !(A \multimap B), A \multimap B) \to \mathcal{F}(W, !(A \multimap B))$ whenever W is of the form $\bigotimes_i !F_i$, satisfying the following axioms:
  - for $f' \in \mathcal{F}(W', W)$ with W' of the form $\bigotimes_j !F'_j$ we have
    $$\mathbf{Y}(f) \circ f' = \mathbf{Y}(f \circ (f' \otimes !(A \multimap B)))$$
  - for $\mathrm{contr}_{W,\mathbf{1},\mathbf{1}} \in \mathcal{F}(W, W \otimes W)$ and $\mathrm{dig}_{W \otimes !(A \multimap B)} \in \mathcal{F}(W \otimes !(A \multimap B), !(W \otimes !(A \multimap B)))$ the morphisms defined below using the the linear exponential comonad, we have
    $$\mathbf{Y}(f) = !f \circ \mathrm{dig}_{W \otimes !(A \multimap B)} \circ (W \otimes \mathbf{Y}(f)) \circ \mathrm{contr}_{W,\mathbf{1},\mathbf{1}}$$

Definition 7.2.1. *We call "pre-model of QΛ!" a category that comes with all the structure described above.*

In QΛ, whenever we had to merge two typing context Γ and Δ into Γ, Δ, the semantics was simply a tensor of their semantics. Now, in QΛ!, we will often have to merge $!\Omega, \Gamma$ and $!\Omega, \Delta$ into $!\Omega, \Gamma, \Delta$. At the semantics level, we will simply use the contraction $\mathrm{c}_F \in \mathcal{V}(!F, !F \otimes !F)$, which we encapsulate in the following morphism.

$[\![\mathbf{1}]\!]$	:=	$\mathbf{1}$	$[\![A \oplus B]\!]$	:=	$[\![A]\!] \oplus [\![B]\!]$
$[\![\mathbf{qubit}]\!]$	:=	$\mathbf{qubit}$	$[\![A \otimes B]\!]$	:=	$[\![A]\!] \otimes [\![B]\!]$
$[\![A \multimap B]\!]$	:=	$[\![A]\!] \multimap\!\bullet [\![B]\!]$	$\left[\!\left[A^\ell\right]\!\right]$	:=	$[\![A]\!]^\ell$
$[\![!(A \multimap B)]\!]$	:=	$!([\![A]\!] \multimap\!\bullet [\![B]\!])$			

Table 7.6: Denotational Semantics for QΛ! Types

Definition 7.2.2. *For every objec W of the form $W = \bigotimes_i !F_i$, and two other objects G, D, we write* $\mathrm{contr}_{W,G,D} \in \mathcal{F}(W \otimes (G \otimes D), (W \otimes G) \otimes (W \otimes D))$ *obtained from the associator, braiding and contraction morphisms.*

In the semantics of the promotion rule, we will need to replicate a context $!\Omega$. For that, we rely on the monoidality morphism $\mathrm{m}_{F,F'} \in \mathcal{V}(!F \otimes !F', !(F \otimes F'))$ and the digging morphism $\delta_F \in \mathcal{V}(!F, !!F)$.

Definition 7.2.3. *For W an object of the form $W = \bigotimes_i !F_i$, we write* $\mathrm{dig}_W \in \mathcal{F}(W, !W)$ *obtained from the associator, monoidality and digging morphisms.*

We describe the semantics of types of QΛ! in Table 7.6, and the semantics of typing derivations of QΛ! in Tables 7.7 to 7.9. The semantics of the promotion uses $[\![-]\!]_{\mathbf{v}}$ which we define similarly to the linear case:

Definition 7.2.4. *If V is a typing derivation for a value $\Gamma \vdash v : A$, then we define its value-semantics $[\![V]\!]_{\mathbf{v}} \in \mathcal{V}([\![\Gamma]\!], [\![A]\!])$ using the same inductive definition as $[\![V]\!]$, up to the following changes: we propagate the $[\![-]\!]_{\mathbf{v}}$ in every rule (except in the abstraction rule where we do not replace the $[\![-]\!]$ by a $[\![-]\!]_{\mathbf{v}}$), and we remove all the occurrences of J.*

Proposition 7.2.5. *We always have $J([\![V]\!]_{\mathbf{v}}) = [\![V]\!]$*

Theorem 7.2.6. *If $\Gamma \vdash t : A$ has two typing derivations T and T', then we have $[\![T]\!] = [\![T']\!]$. As the interpretation is independent from the typing derivation, we write it $[\![t]\!]^{\Gamma \vdash A}$ We define $[\![-]\!]_{\mathbf{v}}^{\Gamma \vdash A}$ similarly.*

The proof is the same as Theorem 1.4.6. In the affine case, this is possible because by definition of **destr**, we necessarily have

$$\mathbf{destr}_{[\![!F]\!]} = \mathrm{w}_{[\![F]\!]}$$

7.2.2 Term Invariance for the Categorical Model

We can now prove that the reduction system of QΛ! on terms satisfies invariance. For that, we start by the value substitution lemma, then we prove the context factorisation lemma.

Structural rules

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x:A, y:B, \Delta \vdash t : C\end{array}}{\Gamma, y:B, x:A, \Delta \vdash t : C} \right]\!\!\right] := [\![T]\!] \circ ([\![\Gamma]\!] \otimes \mathrm{br}_{[\![B]\!],[\![A]\!]} \otimes [\![\Delta]\!])$$

$$\left[\!\!\left[\dfrac{}{\Gamma \vdash \bot_V^A : A} \right]\!\!\right] := 0_{[\![A]\!]} \circ 0^{-1}_{\mathbf{0}\otimes[\![\Gamma]\!]} \circ (\bot \otimes [\![\Gamma]\!]) \circ \mathrm{lu}^{-1}_{[\![\Gamma]\!]}$$

AQΛ! only: $\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash_{\mathbb{A}} t : B\end{array}}{\Gamma, x:A \vdash_{\mathbb{A}} t : B} \right]\!\!\right] := [\![T]\!] \circ ([\![\Gamma]\!] \otimes J(\mathbf{destr}_{[\![A]\!]}))$

Functions type

$$\left[\!\!\left[\dfrac{A \text{ linear}}{x:A \vdash x : A} \right]\!\!\right] := \mathbf{id}_{[\![A]\!]}$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma, x:A \vdash t : B\end{array}}{\Gamma \vdash \lambda x^A.t : A \multimap B} \right]\!\!\right] := J\left(([\![A]\!] \multimap [\![T]\!]) \circ \mathrm{fun}_{[\![\Gamma]\!],[\![A]\!]}\right)$$

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ !\Omega, \Gamma \vdash t : A \multimap B & !\Omega, \Delta \vdash s : A\end{array}}{!\Omega, \Gamma, \Delta \vdash t\ s : B} \right]\!\!\right] := \mathrm{eval}_{[\![B]\!],[\![A]\!]} \circ \left([\![T]\!] \otimes^{\ell} [\![S]\!]\right) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})$$

Exponential rules

$$\left[\!\!\left[\dfrac{}{x :!F \vdash x : F} \right]\!\!\right] := J(\epsilon_F)$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : B\end{array}}{\Gamma, x :!F \vdash t : B} \right]\!\!\right] := [\![T]\!] \circ ([\![\Gamma]\!] \otimes J(\mathrm{w}_F))$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ !\Omega \vdash v : A \multimap B\end{array}}{!\Omega \vdash v :!(A \multimap B)} \right]\!\!\right] := J(!\,[\![T]\!]_{\mathbf{v}} \circ \mathrm{dig}_{[\![!\Omega]\!]})$$

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ !\Omega, f :!(A \multimap B), x : A \vdash t : B & !\Omega, \Gamma, f :!(A \multimap B) \vdash s : C\end{array}}{!\Omega, \Gamma \vdash \mathbf{let\ rec}\ f^{!(A\multimap B)}\ x^A = t\ \mathbf{in}\ s : C} \right]\!\!\right] := \begin{array}{l}[\![S]\!] \circ J((\\ [\![!\Omega, \Gamma]\!] \otimes \mathbf{Y}\left(\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ !\Omega, f :!(A \multimap B), x : A \vdash t : B\end{array}}{!\Omega, f :!(A \multimap B) \vdash \lambda x^A.t : A \multimap B} \right]\!\!\right]\right)\\ ) \circ \mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],\mathbf{1}})\end{array}$$

Table 7.7: Denotational Semantics of QΛ! Typing Derivations, Part 1

Unit type

$$\left[\!\!\left[\dfrac{}{\vdash () : \mathbf{1}} \right]\!\!\right] := \mathbf{id_1}$$

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ !\Omega, \Gamma \vdash t : \mathbf{1} & !\Omega, \Delta \vdash s : A\end{array}}{!\Omega, \Gamma, \Delta \vdash t \ ; \ s : A} \right]\!\!\right] := \begin{array}{l}\mathrm{lu}_{[\![A]\!]} \circ \Big([\![T]\!] \otimes^{\ell} [\![S]\!] \Big) \\ \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})\end{array}$$

Tensor type

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ !\Omega, \Gamma \vdash t : A & !\Omega, \Delta \vdash s : B\end{array}}{!\Omega, \Gamma, \Delta \vdash t \otimes s : A \otimes B} \right]\!\!\right] := \Big([\![T]\!] \otimes^{\ell} [\![S]\!] \Big) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})$$

$$\left[\!\!\left[\dfrac{\begin{array}{cc}\mathrm{T} & \mathrm{S}\\ \vdots & \vdots \\ !\Omega, \Gamma \vdash t : A \otimes B & !\Omega, x : A, y : B \Delta \vdash s : C\end{array}}{!\Omega, \Gamma, \Delta \vdash \mathbf{let}\ x^A \otimes y^B = t\ \mathbf{in}\ s : C} \right]\!\!\right] := \begin{array}{l}[\![S]\!] \circ ([\![T]\!] \otimes [\![!\Omega, \Delta]\!]) \\ \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})\end{array}$$

Sum type

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : A\end{array}}{\Gamma \vdash \mathbf{inj}_{\ell}^{A \oplus B}\ t : A \oplus B} \right]\!\!\right] := \iota_{\ell}^{[\![A]\!] \oplus [\![B]\!]} \circ [\![T]\!]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : B\end{array}}{\Gamma \vdash \mathbf{inj}_{r}^{A \oplus B}\ t : A \oplus B} \right]\!\!\right] := \iota_{r}^{[\![A]\!] \oplus [\![B]\!]} \circ [\![T]\!]$$

$$\left[\!\!\left[\dfrac{\begin{array}{ccc}\mathrm{T} & \mathrm{S1} & \mathrm{S2}\\ \vdots & \vdots & \vdots \\ !\Omega, \Gamma \vdash t : A \oplus B & !\Omega, x : A, \Delta \vdash s_1 : C & !\Omega, y : B, \Delta \vdash s_2 : C\end{array}}{!\Omega, \Gamma, \Delta \vdash \delta\ (t,\ x^A.s_1,\ y^B.s_2) : C} \right]\!\!\right] := \begin{array}{l}[[\![S1]\!]\ ;\ [\![S2]\!]] \circ \mathrm{dis}^{-1}_{[\![A]\!],[\![B]\!],[\![!\Omega,\Delta]\!]} \\ \circ ([\![T]\!] \otimes [\![!\Omega, \Delta]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})\end{array}$$

List type

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : A^{\ell}\end{array}}{\Gamma \vdash \mathbf{unfold}\ t : \mathbf{1} \oplus (A \otimes A^{\ell})} \right]\!\!\right] := [\![T]\!]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{1} \oplus (A \otimes A^{\ell})\end{array}}{\Gamma \vdash \mathbf{fold}\ t : A^{\ell}} \right]\!\!\right] := [\![T]\!]$$

Table 7.8: Denotational Semantics of QΛ! Typing Derivations, Part 2

Quantum primitives

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{bit}\end{array}}{\Gamma \vdash \mathbf{new}\ t : \mathbf{qubit}} \right]\!\!\right] := \mathbf{new}^{\mathcal{C}} \circ [\![T]\!]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}\end{array}}{\Gamma \vdash \mathbf{meas}\ t : \mathbf{bit}} \right]\!\!\right] := \mathbf{meas}^{\mathcal{C}} \circ [\![T]\!]$$

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash t : \mathbf{qubit}^{\otimes n}\end{array}}{\Gamma \vdash \mathbf{U}\ t : \mathbf{qubit}^{\otimes n}} \right]\!\!\right] := \mathbf{U}^{\mathcal{C}} \circ [\![T]\!]$$

Table 7.9: Denotational Semantics of QΛ! Typing Derivations, Part 3

Lemma 7.2.7 (Value Substitution)**.** *For every term* $!\Omega, \Gamma, x : A \vdash t : B$ *and every value* $!\Omega, \Delta \vdash v : A$*:*

$$[\![t]\!] \circ ([\![!\Omega, \Gamma]\!] \otimes [\![v]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]}) = [\![t\{x \leftarrow v\}]\!]$$

Proof. If A is not of the form $!F$, then the proof is the same as for Λ, relying on the first property we asked for $\mathbf{Y}$ to substitute under recursion. If $A = !F$, then any typing derivation for v must have a promotion as the bottom-most rule, and in every typing derivation of t, the axiom for x is an axiom-dereliction rule. We simply proceed by induction similarly to the case where A is not $!F$, relying on the properties of linear exponential comonads, more precisely the interaction between the ! functor and the digging morphism δ used in the promotion rule, the dereliction morphism ϵ used in the axiom-dereliction rule, and the contraction morphisms c used when merging contexts. □

As in the case of Λ, evaluation contexts never capture variables, so $\mathrm{FV}(t) \subseteq \mathrm{FV}(E[t])$. This allows us to state the following lemma.

Lemma 7.2.8 (Context Factorisation)**.** *For every term* $!\Omega, \Gamma \vdash s : A$ *and* $!\Omega, \Gamma, \Delta \vdash E[s] : B$*, with* $E[-]$ *an evaluation context, we have a morphism* $[\![E]\!] \in \mathcal{C}([\![A]\!] \otimes [\![!\Omega, \Delta]\!], [\![B]\!])$ *such that*

$$[\![E[s]]\!] = [\![E]\!] \circ ([\![s]\!] \otimes [\![!\Omega, \Delta]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})$$

Proof. This lemma can be rewritten

$$[\![E[x]]\!] \circ ([\![!\Omega, \Gamma]\!] \otimes [\![s]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]}) = [\![E[x]\{x \leftarrow s\}]\!]$$

So compared to the value-substitution Lemma 7.2.7, we generalised the value v into any term s, and restricted the term t to the special case $E[x]$. Those changes compensate for each others, so the same proof holds. □

Lemma 7.2.9 (Invariance)**.** *For every pair of terms* $\Gamma \vdash t : A$ *and* $\Gamma \vdash s : A$

$$t \to s \implies [\![t]\!] = [\![s]\!]$$

Proof. We proceed by induction on $\to$. The rule for the sequence and the lists are true. The rules for the λ-abstraction, the pairs, and the discriminator directly follow from the value substitution lemma. We look at the reduction rule:

$$\mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ s \to s \left\{ \begin{array}{c} f \leftarrow \lambda x. \\ \mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ t \end{array} \right\}$$

We look at the semantics of the left hand side:

$$[\![\mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ s]\!] = [\![s]\!] \circ J\left([\![!\Omega, \Gamma]\!] \otimes \mathbf{Y}\left([\![\lambda x.t]\!]\right)\right) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],\mathbf{1}})$$

We write r for the typed term $!\Omega \vdash \lambda x^A.\mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ t : !(A \multimap B)$. Using the substitution lemma, we look at the semantics of the right hand side:

$$[\![s\{x \leftarrow r\}]\!] = [\![s]\!] \circ ([\![!\Omega, \Gamma]\!] \otimes [\![r]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],\mathbf{1}})$$

So it is sufficient to prove the $[\![r]\!] = J(\mathbf{Y}([\![\lambda x.t]\!]))$. We unfold the semantics of r:

$$[\![r]\!] = J(!([\![A]\!] \multimap [\![\mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ t]\!]) \circ \mathrm{fun}_{[\![!\Omega]\!],[\![A]\!]}) \circ \mathrm{dig}_{[\![!\Omega]\!]})$$

$$[\![\mathbf{let\ rec}\ f\ x = t\ \mathbf{in}\ t]\!] = [\![t]\!] \circ J\left([\![!\Omega]\!] \otimes \mathbf{Y}\left([\![\lambda x.t]\!]\right)\right) \circ J(\mathrm{contr}_{[\![!\Omega]\!],\mathbf{1},\mathbf{1}})$$

Using naturality of fun in $\mathcal{V}$, we obtain

$$[\![r]\!] = J\left(!\left([\![\lambda x.t]\!] \circ ([\![!\Omega]\!] \otimes \mathbf{Y}\left([\![\lambda x.t]\!]\right)) \circ \mathrm{contr}_{[\![!\Omega]\!],\mathbf{1},\mathbf{1}}\right) \circ \mathrm{dig}_{[\![!\Omega]\!]}\right)$$

Using the linear exponential comonad and the naturality of digging, we have:

$$[\![r]\!] = J\left(!\,[\![\lambda x.t]\!] \circ \mathrm{dig}_{[\![!\Omega, f:!(A\multimap B)]\!]} \circ ([\![!\Omega]\!] \otimes \mathbf{Y}\left([\![\lambda x.t]\!]\right)) \circ \mathrm{contr}_{[\![!\Omega]\!],\mathbf{1},\mathbf{1}}\right)$$

We then rely on the second property we asked for $\mathbf{Y}$:

$$\mathbf{Y}([\![\lambda x.t]\!]) =! [\![\lambda x.t]\!] \circ \mathrm{dig}_{[\![!\Omega, f:!(A\multimap B)]\!]} \circ ([\![!\Omega]\!] \otimes \mathbf{Y}\left([\![\lambda x.t]\!]\right)) \circ \mathrm{contr}_{[\![!\Omega]\!],\mathbf{1},\mathbf{1}}$$

So exactly $[\![r]\!] = J(\mathbf{Y}([\![\lambda x.t]\!]))$. This proves the invariance of the recursion rule. Remains the evaluation context rules. We use the context factorisation lemma and obtain

$$[\![E[t]]\!] = [\![E]\!] \circ ([\![t]\!] \otimes [\![!\Omega, \Delta]\!]) \circ J(\mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]})$$

Invariance by the reduction rule for evaluation context and bottom follows immediately from this factorisation property. □

Chapter 8

Relational Semantics for the Quantum λ-calculus

8.1 Modelling Replication

8.1.1 Quantum Relations on Arenas instead of Webs

The quantum relational model for LQΛ$_!$ is more complex than the one for LQΛ. In particular, objects are significantly more complex, as they shall come with a permutation group. For example, take the type $!(\mathbf{1} \multimap \mathbf{qubit}^\ell)$, and consider three different function calls which return respectively one, three and one qubits. The returned value is represented by a **CPM** operator from $\mathbf{1}$ to Hilbert space $\mathfrak{Q} \otimes \mathfrak{Q}^{\otimes 3} \otimes \mathfrak{Q}$. However, QΛ$_!$ satisfies a uniformity property: a function cannot distinguish between its different calls, hence from the point of view of the called function, those three calls are unordered and can be shuffled around, whereas the tensor $\mathfrak{Q} \otimes \mathfrak{Q}^{\otimes 3} \otimes \mathfrak{Q}$ puts them in a specific order. To enforce this property, we must add to the type a group of permutations under the action of which quantum valuations must be invariant, ensuring that the function does not behave fundamentally differently on different calls.

The category defined in [PSV14] tackles this issue perfectly. However, for conciseness we choose instead to take an approach much more similar to the solution used in our game model. Instead of extending **QRel** (quantum relations) and **QA** (quantum strategies) into two denotational models for LQΛ$_!$, and building a collapse from **QA** to **QRel** by relating exhaustive configurations to points of the web, we will extend **QARel** (quantum relations on arenas) and **QA** (quantum strategies), into two denotational models $\sim$-**QARel** and $\sim$-**QA** for LQΛ$_!$, and build a collapse from $\sim$-**QA** to $\sim$-**QARel**.

Similarly to how we have proved a full and faithful functor that preserves the structure from **QARel** to **QRel** (*i.e.*, those two categories have "the same maps" but different objects), we will have a full and faithful functor from $\sim$-**QARel** to the model presented in

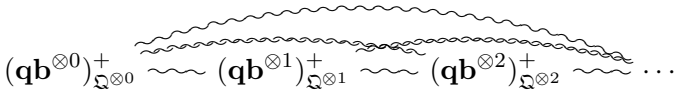


Figure 8.1: Arena for $\mathbf{qubit}^{\ell}$.

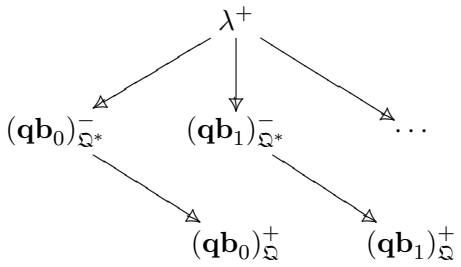


Figure 8.2: Arena for $!(\mathbf{qubit} \multimap \mathbf{qubit})$.

[PSV14].

8.1.2 Arenas and the Notion of Symmetry

We recall the guiding example

$$f : !(\mathbf{qubit} \multimap \mathbf{qubit}), \ell : \mathbf{qubit}^{\ell} \vdash_{\mathbb{L}} \mathrm{Map}(f, \ell) : \mathbf{qubit}^{\ell}$$

When giving a semantics to LQΛ!, we will extend the semantics of LQΛ. In particular, the arena for $\mathbf{qubit}^{\ell}$ will be the same as in the case of LQΛ, *i.e.*, the arena described in Fig. 8.1. To represent $!(\mathbf{qubit} \multimap \mathbf{qubit})$, we use the arena described in Fig. 8.2, which is simply the arena of $\mathbf{qubit} \multimap \mathbf{qubit}$ with the function call copied countably many times:

- It starts with a single positive event λ^{+} signifying "the function is ready to be called".
- This minimal event is followed by a (countable) infinity of events $(\mathbf{qb}_n)^{-}_{\mathfrak{Q}^*}$, all compatible with each other, each of them representing a different call to the function, either with the same or with a different value as argument.
- Each event $(\mathbf{qb}_n)^{-}_{\mathfrak{Q}}$ is followed by an event $(\mathbf{qb}_n)^{+}_{\mathfrak{Q}}$ representing the output of the function when called.

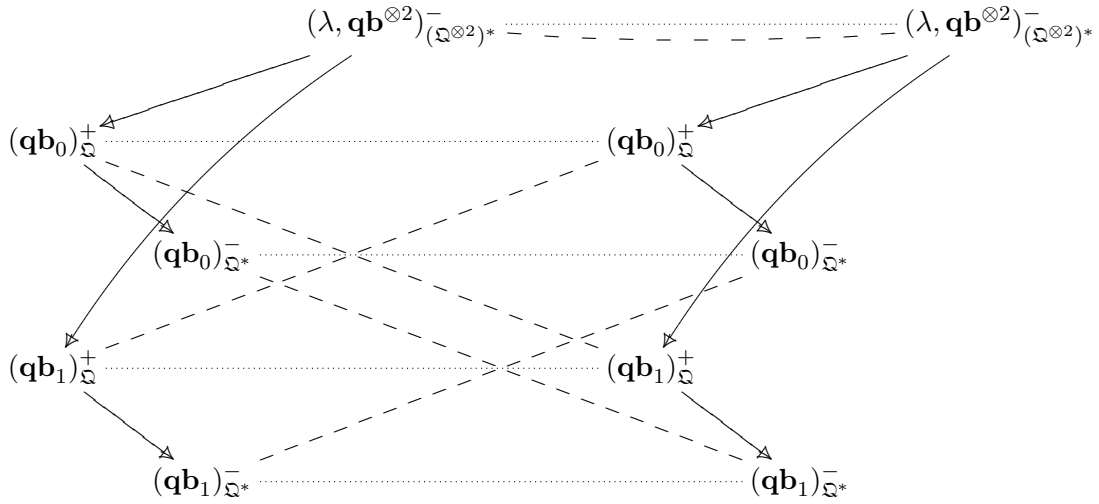
However, without additional structure, Fig. 8.2 is missing some of the information of the type: $!(\mathbf{qubit} \multimap \mathbf{qubit})$ is not the same as $(\mathbf{qubit} \multimap \mathbf{qubit}) \otimes (\mathbf{qubit} \multimap \mathbf{qubit}) \otimes \ldots$. Indeed, as said earlier, ! satisfies a notion of uniformity, meaning that this is the same function duplicated, not infinitely many different functions. To represent this information, we follow [CCW19] and introduce the notion of a $\sim$-arena. In a $\sim$-arena, we have a relation

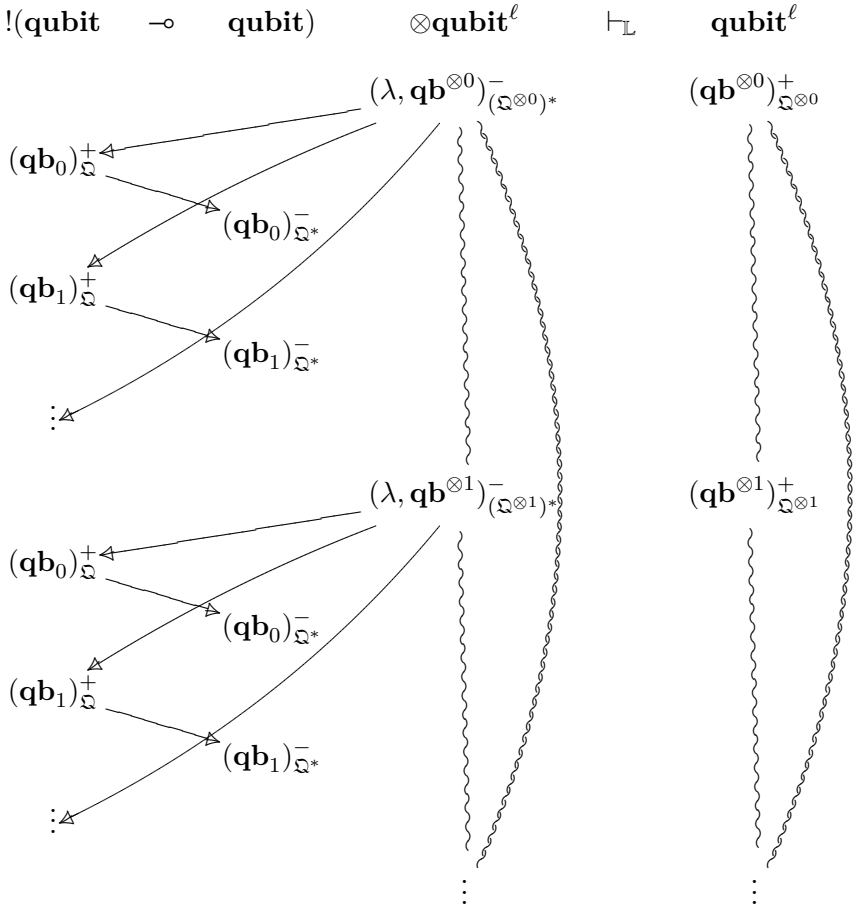
$\simeq$ between configurations called symmetry, which means "those configurations are the same up to changing copy indices of replicable function calls". In Fig. 8.2, the equivalence classes for the relation $\simeq$ would be:

$$\begin{array}{ccccccc}
\varnothing & & & & & & \\
\{\lambda^+\} & & & & & & \\
\{\lambda^+, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}\} & \simeq & \{\lambda^+, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}\} & \simeq & \cdots \\
\{\lambda^+, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0)^+_{\mathfrak{Q}}\} & \simeq & \{\lambda^+, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1)^+_{\mathfrak{Q}}\} & \simeq & \cdots \\
\{\lambda^+, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}\} & \simeq & \{\lambda^+, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_2)^-_{\mathfrak{Q}^*}\} & \simeq & \cdots \\
\{\lambda^+, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0)^+_{\mathfrak{Q}}\} & \simeq & \{\lambda^+, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1)^+_{\mathfrak{Q}}\} & \simeq & \cdots
\end{array}$$

In $\sim$-arenas, the relation $\simeq$ will have two sub-relations $\simeq^+$ and $\simeq^-$, corresponding to "those configurations are the same up to Player changing copy indices of replicable function calls" and "those configurations are the same up to Opponent changing copy indices of replicable functions calls". In Fig. 8.2, all the symmetries $\simeq$ are in fact negative symmetries $\simeq^-$, since it is Opponent which has the agency of changing copy indices by choosing which copy of the function they call. Conversely, in the $\sim$-arena for $!(\mathbf{qubit} \multimap \mathbf{qubit}) \otimes \mathbf{qubit}^\ell \vdash \mathbf{qubit}^\ell$ described in Fig. 8.3, all the symmetries $\simeq$ are in fact positive symmetries $\simeq^+$, since we use the dual of the $\sim$-arena described in Fig. 8.2, so now it is Player who has the agency to choose which copy of the function to use. In the case of the quantum relational model on arenas, we will not use $\simeq^+$ and $\simeq^-$, but those sub-symmetries will be central to the definition of quantum $\sim$-strategies.

In practice, we will need more than just the equivalence classes of configurations for $\simeq$, $\simeq^-$ and $\simeq^+$. We will need to know *how* two configurations are symmetric. For example, in Fig. 8.3, the configuration $\{(\lambda, \mathbf{qb}^{\otimes 2})^-_{(\mathfrak{Q}^{\otimes 2})^*}, (\mathbf{qb}_0)^+_{\mathfrak{Q}}, (\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1)^+_{\mathfrak{Q}}, (\mathbf{qb}_1)^-_{\mathfrak{Q}^*}\}$ is symmetric to itself in two different ways, represented respectively by dotted and dashed lines below:





Symmetry:

$x \simeq y$ whenever there is an order-isomorphism between x and y

Figure 8.3: $\sim$-Arena for $!(\mathbf{qubit} \multimap \mathbf{qubit}) \otimes \mathbf{qubit}^\ell \vdash_{\mathbb{L}} \mathbf{qubit}^\ell$

We formally define those arenas with symmetry in Section 8.2

8.1.3 Quantum Valuations and Symmetry

On those $\sim$-arenas, we will consider quantum relations which we define in Section 8.4, and quantum $\sim$-strategies which we define in Section 9.3.2. Both those notions will leverage the symmetries of the $\sim$-arenas in some way, representing the fact that different copies of a replicable function are indistinguishable.

For that, we will lift the bijections describing how configurations are symmetric to one another into a set of morphisms of **CPM** describing some permutations of Hilbert spaces, and ask the quantum valuation to be preserved under those morphisms.

Considering once again the example of Fig. 8.3, we can look at the configuration

$$\{(\lambda, \mathbf{qb}^{\otimes 2})^-_{\mathfrak{Q}^{\otimes 2}}, (\mathbf{qb}_0)^+_{\mathfrak{Q}}, (\mathbf{qb}_0)^-_{\mathfrak{Q}}, (\mathbf{qb}_1)^+_{\mathfrak{Q}}, (\mathbf{qb}_1)^-_{\mathfrak{Q}}\}$$

on the left hand side, and at the configuration $\{(\lambda, \mathbf{qb}^{\otimes 2})^+_{\mathfrak{Q}^{\otimes 2}}\}$ on the right hand side. If we write $\mathcal{Q}$ the quantum annotation from the quantum relations on $\sim$-arenas for those configurations, we then have:

$$\mathcal{Q} \in \overline{\mathbf{CPM}}((\mathfrak{Q}^* \otimes \mathfrak{Q}) \otimes (\mathfrak{Q}^* \otimes \mathfrak{Q}) \otimes \mathfrak{Q}^{\otimes 2}, \mathfrak{Q}^{\otimes 2})$$

The two auto-symmetries on the left hand side can be lifted into the identity morphism, and the morphism $\mathrm{br}^{\mathbf{CPM}}_{(\mathfrak{Q}^*\otimes\mathfrak{Q}),(\mathfrak{Q}^*\otimes\mathfrak{Q})} \otimes \mathbf{id}^{\mathbf{CPM}}_{\mathfrak{Q}^{\otimes 2}}$. So we will expect the following condition to be satisfied:

$$\mathcal{Q} = \mathcal{Q} \circ (\mathrm{br}^{\mathbf{CPM}}_{(\mathfrak{Q}^*\otimes\mathfrak{Q}),(\mathfrak{Q}^*\otimes\mathfrak{Q})} \otimes \mathbf{id}^{\mathbf{CPM}}_{\mathfrak{Q}^{\otimes 2}})$$

8.2 Event Structures with Symmetry

The goal of this section is to define and state basic properties about the objects of our two categories of quantum relations on $\sim$-arenas and quantum $\sim$-strategies. More details about this notion of symmetry can be found in [CCW19, Win07].

8.2.1 The Category of Event Structures with Symmetry

Definition 8.2.1. *For $(E, \leq)$ a poset, we say that θ is an order-isomorphism from a down-closed set x to a down-closed set y if $\theta : x \to y$ is a bijection that preserves and reflects*[1] *the order $\leq$.*

When considering order-isomorphisms, we will use both the functional notation $\theta : e \mapsto e'$ and the relational notation $\theta = \{(e, e'), \dots\}$. If x' is a down-closed subset of x, we write $\theta|_{x'}$ for θ restricted to x'. If $\simeq$ denotes a set of order-isomorphisms, then we write $\theta : x \simeq y$ when $\theta \in \simeq$ and θ is an order isomorphism from x to y.

[1] A bijection reflects a relation if its inverse preserves it.

Definition 8.2.2. *A $\sim$-es $(|E|, \leq_E, \mathrm{Con}_E, \simeq_E)$ is an event structure together with a set $\simeq_E$ of order-isomorphisms between configurations such that*

Groupoid *The set $\simeq_E$ contains all the identity order-isomorphisms, and is stable under composition and inverse.*

Restriction *For every $\theta : x \simeq_E y$, if $x' \subseteq x$, then $\theta|_{x'}(x') \in \mathcal{C}(E)$ and $\theta|_{x'} \in \simeq_E$.*

Extension *For every $\theta : x \simeq_E y$, if $x \subseteq x'$ then there exists a non-necessarily unique $y \subseteq y' \in \mathcal{C}(E)$ and $\phi : x' \simeq_E y'$ such that $\phi|_x = \theta$.*

We write $\mathcal{C}_{\simeq}(E)$ for the equivalence classes of configurations of E, and $\mathbf{x}$ for the equivalence class containing $x \in \mathcal{C}(E)$.

$$\mathbf{x} = \{y \in \mathcal{C}(E) \mid \exists \theta : x \simeq_E y\}$$

While we do not have explicit copy indices on events in the generality of $\sim$-arenas, the intuition "$\theta : x \simeq y$ means that x and y are the same up to copy indices" will still stand.

A particularly interesting subset of $\simeq$ is the set of auto-symmetries:

Definition 8.2.3. *For E a $\sim$-es and $x \in \mathcal{C}(E)$, we define $\mathcal{A}_E(x)$ the set of auto-symmetries over x, i.e., $\mathcal{A}_E(x) = \{\theta : x \simeq_E x\}$.*

Lemma 8.2.4. *Whenever $\theta : x' \simeq x$, we have $|\mathcal{A}_E(x')| = |\mathcal{A}_E(x)|$*

Proof. The operation $\phi \mapsto \theta^{-1} \circ \phi \circ \theta$ forms a bijection between $\mathcal{A}_E(x)$ and $\mathcal{A}_E(x')$. □

We can extend the notion of polarity to $\sim$-es.

Definition 8.2.5. *A $\sim$-esp $(|E|, \leq_E, \mathrm{Con}_E, p_E \simeq_E)$ is both an esp[2] and a $\sim$-es, such that every symmetry $\theta \in \simeq$ preserves the polarities.*

For E a $\sim$-es, we consider $\theta : x \simeq_E y$ and $\phi : x' \simeq_E y'$. We write $\theta \subseteq \phi$ if $x \subseteq x'$ and $\phi|_x = \theta$. If E is a $\sim$-esp, we define $\theta \subseteq^- \phi$ and $\theta \subseteq^+ \phi$ similarly.

To build a category, we need a notion of maps of $\sim$-esps. Similarly to the notion of maps of esps, this notion can be seen as a notion of "simulation" of one event structure by the other.

Definition 8.2.6. *Maps of $\sim$-es are maps of event structures $f : E \to E'$ that preserve the symmetry:*

$$(e, e') \in \theta \in \simeq_E \text{ and } e \in \mathrm{dom}(f) \implies e' \in \mathrm{dom}(f)$$

$$\theta : x \simeq y \implies \{(f(e), f(e')) \mid (e, e') \in \theta\} : f(x) \simeq_{E'} f(y)$$

We write $f\,\theta := \{(f(e), f(e')) \mid (e, e') \in \theta\}$. For $f, g : E \rightharpoonup E'$ two maps of $\sim$-es, we say that f and g are symmetric and write $f \simeq g$ if for every $x \in \mathcal{C}(E)$ we have

$$\{(f(e), g(e)) \mid e \in x\} : f(x) \simeq_{E'} g(x)$$

[2] An event structure with polarities. See Definition 4.4.1.

As in the case of event structures, the definition covers both partial ($\rightharpoonup$) and total ($\to$) maps, but we will almost always use total maps. We define similarly maps of $\sim$-esps as maps of esps that preserve the symmetry.

Proposition 8.2.7. *The relation $\simeq$ forms a congruence for the category $\sim$-**ES** of $\sim$-es and total maps of $\sim$-es, and for the category $\sim$-**ESP** of $\sim$-esps and total maps of $\sim$-esps.*

We recall that a congruence is simply an equivalence relation on maps that is compatible with the structure of the category: if $f \simeq f'$ and $g \simeq g'$ then $f \circ g \simeq f' \circ g'$. In fact, this relation is also compatible with the additional structures which we define now. For E and E' two $\sim$-esp:

- We define $E^\perp$ as the esp $E^\perp$ with the same symmetry.
- For $\theta : x \simeq_E y$ and $\theta' : x' \simeq_{E'} y'$, we define $\theta \parallel \theta'$ as the order-isomorphism between $x \parallel x'$ and $y \parallel y'$ obtained by the disjoint union of θ and θ'. We define $E \parallel E'$ as the esp $E \parallel E'$ with for symmetries every $\theta \parallel \theta'$ for $\theta \in \simeq_E$ and $\theta' \in \simeq_{E'}$.
- For $\theta : x \simeq_E y$ we define $\theta \oplus \varnothing$ as the order-isomorphism between $x \oplus \varnothing$ and $y \oplus \varnothing$ induced by θ. For $\theta' : x' \simeq_{E'} y'$, we define $\varnothing \oplus \theta'$ symmetrically. We then define $E \oplus E'$ as the esp $E \oplus E'$ with for symmetries every $\theta \oplus \varnothing$ and every $\varnothing \oplus \theta'$ for $\theta \in \simeq_E$ and $\theta' \in \simeq_{E'}$.
- We define the empty $\sim$-esp $\varnothing$ as the esp $\varnothing$ with the trivial symmetry. Up to isomorphism, it is a unit for both $\parallel$ and $\oplus$.

Proposition 8.2.8. *The category $\sim$-**ESP** of $\sim$-esps and total maps between them forms an SMC for $\parallel$, and a cocartesian category for the coproduct $\oplus$. $\sim$-**ESP** also has arbitrary coproducts. The relation $\simeq$ is a congruence for those additional structures.*

8.2.2 Games with Symmetry

The $\sim$-esps defined above are very general, and it is possible to define symmetries that cannot be simply explained through the idea of "exchanging copy indices". This level of generality can be very problematic in some instances, and would be a major obstacle to the definition of the category of $\sim$-strategies. That is why in this section we refine it as hinted before by distinguishing two sub-symmetries of $\simeq$ corresponding to "only Player is changing copy indices" and "only Opponent is changing copy indices". We call $\sim$-games those refined $\sim$-esps. The notion of $\sim$-game coincides with the notion of thin concurrent game defined in [CCW19], modulo the representability condition which was added later in [Cla20]. The name "thin concurrent games" comes from the fact that this refinement of $\simeq$ into $(\simeq, \simeq^+, \simeq^-)$ is central to the category of "thin concurrent strategies" developed in this same paper. The quantum $\sim$-strategies (see Section 9.3.2) we will define are quantum variants of those thin concurrent strategies. Since we will not rely on the

polarised symmetries $\simeq^+$ and $\simeq^-$ for the relational model, we include them in the definition of $\sim$-games but postpone to Definition 9.1.1 the details of the conditions they must satisfy.

Definition 8.2.9. *A $\sim$-game is an esp E together with*

- *three sets $\simeq_E, \simeq_E^+, \simeq_E^-$ called symmetry, positive symmetry and negative symmetry, such that $(E, \simeq_E)$, $(E, \simeq_E^+)$ and $(E, \simeq_E^-)$ are $\sim$-esps, and $\simeq_E^+$ and $\simeq_E^-$ are subsets of $\simeq_E$;*
- *a selection of canonical configurations $\underline{(-)} : \mathcal{C}_\simeq(E) \to \mathcal{C}(E)$ such that $\underline{\mathbf{x}} \in \mathbf{x}$ for all $\mathbf{x} \in \mathcal{C}_\simeq(E)$;*
- *an implicit total order on $\mathcal{C}_\simeq(E)$;*

satisfying the properties described in Definition 9.1.1. We write $\mathcal{A}_E(x)$, $\mathcal{A}_E^+(x)$ and $\mathcal{A}_E^-(x)$ for the sets of auto-symmetries, auto-positive-symmetries and auto-negative-symmetries over x.

We do not assume any specific property on the implicit total ordering of $\mathcal{C}_\simeq(E)$, as we just need it to exist and be fixed. This ordering will be used when ranging over $\mathcal{C}_\simeq(E)$ with operations that are not commutative, like $\|_{\mathbf{x} \in \mathcal{C}_\simeq(E)} \dots$ in the following subsection. In contrast with [CCW19] for example, our $\sim$-games come with an explicit choice of canonical representatives for equivalence classes $\underline{(-)}$. This will be very useful when considering quantum valuations in later sections. The core of the additional restrictions on $\sim$-games are to ensure that $\simeq_E^-$ and $\simeq_E^+$ indeed correspond to Opponent choices and Player choices respectively. We postpone them to Definition 9.1.1 as they are only used for defining $\sim$-strategies. We now extend the structure of $\sim$-**ESP** to $\sim$-games, starting with maps.

Definition 8.2.10. *Maps of $\sim$-games are maps of esps that are maps of $\sim$-esps (for the three symmetries). We write $f \simeq g$ whenever two maps of $\sim$-games f and g are symmetric as maps of $\sim$-esps.*

In particular, we do not expect the canonical selection of configuration to be preserved by maps. For E, E' two $\sim$-games:

- We define $E^\perp$ as the esp $E^\perp$ with $\simeq^-$ and $\simeq^+$ exchanged, and the same $\underline{-}$.
- We define $E \parallel E'$ as the $\sim$-esp $E \parallel E'$ for the three symmetries and $\underline{\mathbf{x} \parallel \mathbf{y}} := \underline{\mathbf{x}} \parallel \underline{\mathbf{y}}$.
- We define $E \oplus E'$ as the $\sim$-esp $E \oplus E'$ for the three symmetries, and $\underline{\mathbf{x} \oplus \varnothing} := \underline{\mathbf{x}} \oplus \varnothing$, $\underline{\varnothing \oplus \mathbf{y}} := \varnothing \oplus \underline{\mathbf{y}}$.
- We define the empty $\sim$-game $\varnothing$ as $(\varnothing, \{\mathbf{id}_\varnothing\}, \{\mathbf{id}_\varnothing\}, \{\mathbf{id}_\varnothing\}, \underline{\varnothing} = \varnothing)$. Up to isomorphism, it is a unit for both $\parallel$ and $\oplus$.

Proposition 8.2.11. *The category $\sim$-**Game** of $\sim$-games and total maps between them forms an SMC for $\parallel$, and a cocartesian category for the coproduct $\oplus$. $\sim$-**Game** also has arbitrary coproducts. The relation $\simeq$ is a congruence.*

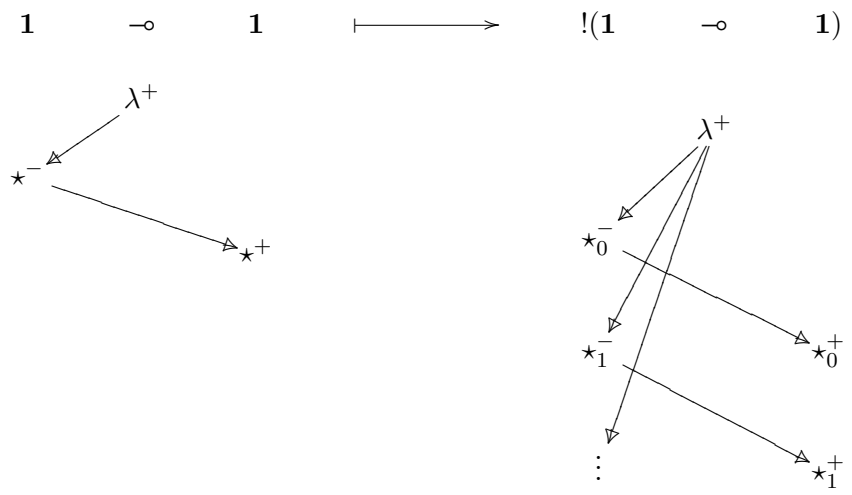


Figure 8.4: The ! operation on $\mathbf{1} \multimap \mathbf{1}$

8.2.3 The Linear Exponential Comonad !

The goal of this subsection is to build the operation ! that will allow us to compute the $\sim$-game for $!(A \multimap B)$ from the $\sim$-game for $A \multimap B$, and then show that it is a linear exponential comonad[3]. The definition of ! can be found in [CCW19] and is a variation of the definition of the exponential in AJM games, but the remainder of the development of this subsection, starting with anti-maps, is a contribution of this thesis.

We illustrate the ! construction in Fig. 8.4. Something notable is that ! does not affect the minimal event λ^+. Operationally, this is because the minimal event of $!(A \multimap B)$ corresponds to the computation done before the function call, and is not duplicated if the function is called more than once. Categorically, this is because ! is naturally defined on negative[4] $\sim$-games, and then lifted to positive[5] games by adding the minimal event λ^+. We postpone the lifting of ! to positive $\sim$-games to Section 8.3.2, and we first focus on its action on negative $\sim$-games.

Definition 8.2.12. *For $(E, \sim_E, \sim_E^+, \sim_E^-)$ a negative $\sim$-game, we define the negative $\sim$-game $(!E, \sim_{!E}, \sim_{!E}^+, \sim_{!E}^-)$ as the esp $E \parallel E \parallel \dots$ (so its events are $|!E| = \mathbb{N} \times |E|$) together with the symmetries:*

$$\begin{array}{lll}
\theta : (x_0 \parallel x_1 \parallel \dots) \simeq_{!E} (y_0 \parallel y_1 \parallel \dots) & \text{if} & \exists \sigma \text{ perm. of } \mathbb{N}, \quad \forall n \in \mathbb{N}, \theta|_{x_n} : x_n \simeq_E y_{\sigma(n)} \\
\theta : (x_0 \parallel x_1 \parallel \dots) \simeq_{!E}^+ (y_0 \parallel y_1 \parallel \dots) & \text{if} & \forall n \in \mathbb{N}, \theta|_{x_n} : x_n \simeq_E^+ y_n \\
\theta : (x_0 \parallel x_1 \parallel \dots) \simeq_{!E}^- (y_0 \parallel y_1 \parallel \dots) & \text{if} & \exists \sigma \text{ perm. of } \mathbb{N}, \quad \forall n \in \mathbb{N}, \theta|_{x_n} : x_n \simeq_E^- y_{\sigma(n)}
\end{array}$$

[3] See Definition 1.2.21.

[4] All the minimal events of the $\sim$-game are negative.

[5] All the minimal events of the $\sim$-game are positive.

And the canonical selection for $\mathbf{x} = \mathbf{x_0} \parallel \mathbf{x_1} \parallel \ldots \in \mathcal{C}_{\simeq}(!E)$*:*

$$\underline{\mathbf{x}} := \|_{\mathbf{y} \in \mathcal{C}_{\simeq}(E)} \underline{\mathbf{y}}^{\|\mathrm{card}\{n \mid x_n \in \mathbf{y}\}}$$

For $(E, \sim_E, \sim_E^+, \sim_E^-)$ *a positive* $\sim$*-game, we define the positive* $\sim$*-game* $(?E, \sim_{?E}, \sim_{?E}^+, \sim_{?E}^-)$ *dually.*

We have $(!E)^\perp = ?(E^\perp)$. For the canonical selection, we cannot simply take $\underline{\mathbf{x}} = \underline{\mathbf{x_0}} \parallel \underline{\mathbf{x_1}} \parallel \ldots$ as we need the canonical selection to be the same for every $x' \in \mathbf{x}$. This is why we order the $\underline{\mathbf{x_i}}$ in an "canonical" way. For that we rely on the implicit total order on $\mathcal{C}_{\simeq}(E)$ from Definition 8.2.9.

We would like to show that ! forms a linear exponential comonad in $\sim$-**Game**, but this is incorrect. While ! will be a linear exponential comonad for both quantum relations on $\sim$-arenas and quantum $\sim$-strategies, the maps of $\sim$-**Game** are conveying an operational meaning too different from the strategies. For example, we do not have a weakening map from $!E$ to $\varnothing$, as maps of $\sim$-**Game** are total. However, we do always have a (unique) map from $\varnothing$ to $(!E)^\perp$. In the following, we define a notion of "anti-map", and show that for anti-maps of $\sim$-games, ! is a linear exponential comonad. In later sections, we will show that we can lift every anti-map of $\sim$-games into a quantum relation on $\sim$-arenas, and into a quantum $\sim$-strategy. This lifting will allow us to deduce that ! is a linear exponential comonad in both $\sim$-**QARel** and $\sim$-**QA** from the fact that it is a linear exponential comonad for anti-maps.

Definition 8.2.13. *An anti-map of* $\sim$*-games from* A *to* B *is a map of* $\sim$*-games from* $B^\perp$ *to* $A^\perp$*. We write* $\sim$-$\mathbf{Game}^\perp$ *for the category of* $\sim$*-games and total anti-maps between them.*

Proposition 8.2.14. *The category* $(\sim\text{-}\mathbf{Game}^\perp, \|, \varnothing)$ *is an SMC. The relation* $\simeq$ *is a congruence for* $\sim$-$\mathbf{Game}^\perp$ *too.*

Proposition 8.2.15. *The SMC* $(\sim\text{-}\mathbf{Game}^\perp_\ominus, \|, \varnothing)$ *of negative* $\sim$*-games and (total) anti-maps of* $\sim$*-games has a linear exponential comonad* ! *up to* $\simeq$*.*

Proof. The ! functor is given by $!f = f \parallel f \parallel \ldots$. To prove that ! is a linear exponential comonad for anti-maps, we need to check that ? is the dual of a linear exponential comonad for maps. The comonoidality, coweakening, cocontraction, codereliction, codigging maps are given by:

$$\begin{array}{rccc} \mathrm{cm}_\varnothing : & ?\varnothing & \to & \varnothing \\ & - & \mapsto & - \end{array} \qquad \begin{array}{rccc} \mathrm{cm}_{A,B} : & ?(A \parallel B) & \to & ?A \parallel ?B \\ & (n,(0,a)) & \mapsto & (0,(n,a)) \\ & (n,(1,b)) & \mapsto & (1,(n,b)) \end{array}$$

$$\begin{array}{rccc} \mathrm{cw}_A : & \varnothing & \to & ?A \\ & - & \mapsto & - \end{array} \qquad \begin{array}{rccc} \mathrm{cc}_A : & ?A \parallel ?A & \to & ?A \\ & (0,(n,a)) & \mapsto & (2n,a) \\ & (1,(n,a)) & \mapsto & (2n+1,a) \end{array}$$

$$\begin{array}{rccc} \mathrm{c}\epsilon_A : & A & \to & ?A \\ & a & \mapsto & (0,a) \end{array} \qquad \begin{array}{rccc} \mathrm{c}\delta_A : & ??A & \to & ?A \\ & (n,(m,a)) & \mapsto & (\iota(n,m),a) \end{array}$$

where $\iota(n,m)$ is a bijection between $\mathbb{N}\times\mathbb{N}$ and $\mathbb{N}$. They satisfy the dual axioms of a linear exponential comonad (described in Section 1.2.4) up to $\simeq$. In fact they are true up to $\simeq^+$. Taking the corresponding anti-maps, we obtain $(!,\epsilon,\delta,\mathrm{w},\mathrm{c},\mathrm{m},\mathrm{m_1})$ which satisfies the axioms up to $\simeq$. □

8.3 Quantum Games and Arenas with Symmetry

8.3.1 Quantum Payoff Games with Symmetry

We extend the previous definitions with Hilbert space annotations and payoff.

Definition 8.3.1. *A quantum payoff $\sim$-game, is a quantum payoff game*[6] *E together with $\simeq_E, \simeq_E^+, \simeq_E^-, =$ such that:*

- *$(E, \simeq_E, \simeq_E^+, \simeq_E^-, =)$ is a $\sim$-game*
- *$\forall\theta : x \simeq_E y, \kappa_E(x) = \kappa_E(y)$ and $\forall e \in x, \mathcal{H}_E(e) = \mathcal{H}_E(\theta(e))$*

We define $\mathcal{E}_\simeq(E)$ as the set of equivalence classes of exhaustive configurations:

$$\mathcal{E}_\simeq(E) = \{\mathbf{x} \in \mathcal{C}_\simeq(E) \mid \kappa_E(\underline{\mathbf{x}}) = 0\}$$

We note that whenever $\theta : x \simeq y$, we have $\forall e \in |E|, \mathcal{H}_E(e) = \mathcal{H}_E(\theta(e))$ but that does not mean that $\mathcal{H}_E(x) = \mathcal{H}_E(y)$. Indeed, $\mathcal{H}_E(x)$ and $\mathcal{H}_E(y)$ are necessarily tensors involving the same Hilbert spaces, but not necessarily in the same order. We can lift this reordering of Hilbert spaces into a permutation isomorphism[7] in $\mathcal{H}_E(\theta) \in \mathbf{CPM}(\mathcal{H}_E(x), \mathcal{H}_E(y))$.

Definition 8.3.2. *For E a quantum payoff $\sim$-game, we define the $\sim$-Scott category of E as the category with objects $\mathcal{C}(E)$ and morphisms:*

$$(x \ {}^{-}\!\supseteq \overset{\theta}{\simeq} \subseteq^{+} y) \in \sim\text{-}Scott(x,y) \text{ whenever } \begin{cases} x \ {}^{-}\!\supseteq x' \\ \theta : x' \simeq_E y' \\ y' \subseteq^{+} y \end{cases}$$

[6] So an alternating race-free esp with an Hilbert space annotation on events and a payoff annotation on configurations. See Definition 4.4.4.

[7] A permutation isomorphism is an isomorphism obtained by composing only associator and braiding isomorphisms.

For E a quantum payoff $\sim$-game, $\mathcal{H}$ extends to a contravariant functor from the $\sim$-Scott category to **CPM**, *as follows:*

$$\begin{array}{lcll}
\mathcal{H}_E(x) & := & \bigotimes_{e\in x} \mathcal{H}(e) & \\
\mathcal{H}_E(x \subseteq^+ y) & := & \mathbf{id}^{\mathbf{CPM}}_{\mathcal{H}_E(x)} \otimes \mathbf{Tr}_{\mathcal{H}_E(y\backslash x)} & \in \ \mathbf{CPM}(\mathcal{H}_E(y), \mathcal{H}_E(x)) \\
\mathcal{H}_E(x \ {}^{-}\!\supseteq y) & := & \mathbf{id}^{\mathbf{CPM}}_{\mathcal{H}_E(y)} \otimes \mathbf{Tr}^{\dagger}_{\mathcal{H}_B(x\backslash y)} & \in \ \mathbf{CPM}(\mathcal{H}_E(y), \mathcal{H}_E(x)) \\
\mathcal{H}_E(\theta : x \simeq_E y) & := & \mathcal{H}_E(\theta) & \in \ \mathbf{CPM}(\mathcal{H}_E(y), \mathcal{H}_E(x))
\end{array}$$

This allows us to express the idea that "quantum annotation is preserved by symmetry", which we call uniformity.

Definition 8.3.3. *For E a quantum payoff $\sim$-game, and for $G \subseteq \sim\text{-}Scott(x, y)$, we define*[8]

$$\mathcal{H}_E(G) := \sum_{g\in G} \frac{\mathcal{H}_E(g)}{|G|} \in \mathbf{CPM}(\mathcal{H}_E(y), \mathcal{H}_E(x))$$

A function $f \in \overline{\mathbf{CPM}}(\mathcal{H}_E(x), \mathcal{H}_E(y))$ is uniform if

$$f = \mathcal{H}_E(\mathcal{A}_E(y)) \circ f \circ \mathcal{H}_E(\mathcal{A}_E(x))$$

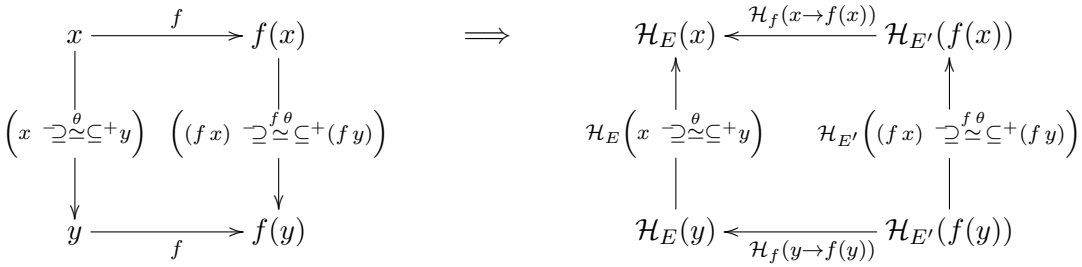
The operations $(_)^\perp, \oplus, \parr$ and $\boxtimes$ on quantum payoff games extend to quantum payoff $\sim$-games, with the symmetry respectively behaving as for the operations $(_)^\perp, \oplus, \|$ and again $\|$ on $\sim$-games.

Definition 8.3.4. *A map of quantum payoff $\sim$-games is a total map $f : E \to E'$ of $\sim$-games such that f preserves the Hilbert space annotation and is payoff non-decreasing:*

$$\forall e \in |E|, \mathcal{H}_E(e) = \mathcal{H}_{E'}(f(e)) \qquad\qquad \forall x \in \mathcal{C}(E), \kappa_E(x) \leq \kappa_{E'}(f(x))$$

The payoff non-decreasing condition implies that a map must send non-losing configurations to non-losing configurations. We note that while $\forall e \in |E|, \mathcal{H}_E(e) = \mathcal{H}_{E'}(f(e))$, that does not mean that $\forall x \in \mathcal{C}(E), \mathcal{H}_E(x) = \mathcal{H}_{E'}(f\,x)$, that just means they are related by a permutation isomorphism.

Proposition 8.3.5. *A map of quantum payoff $\sim$-games $f : E \to E'$ induces a permutation isomorphism $\mathcal{H}_f(x \to f(x)) \in \mathbf{CPM}(\mathcal{H}_{E'}(f(x)), \mathcal{H}_E(x))$ for every $x \in \mathcal{C}(E)$, such that*



[8] We note that G is necessarily finite, of cardinality at most $|x|!$.

Definition 8.3.6. *An anti-map of quantum payoff $\sim$-games from A to B is a map of quantum payoff $\sim$-games from $B^\perp$ to $A^\perp$.*

Proposition 8.3.7. *The category $\sim$-**QGame** (resp. $\sim$-**QGame**$^\perp$) of quantum payoff $\sim$-games and total maps (resp. anti-maps) between them forms two SMCs for $⅋$ and $\boxtimes$, and a cocartesian (resp. cartesian) category for the coproduct $\oplus$. $\sim$-**Game** also has arbitrary coproducts (resp. products). The relation $\simeq$ is a congruence.*

It is not a $\star$-autonomous category. The closure only exists at the level of strategies, but not of maps of $\sim$-games.

Definition 8.3.8. *For E a negative quantum payoff $\sim$-game, we define $!E$ as the $\sim$-game $!E$ together with*

$$\forall (n,e) \in |!E| = \mathbb{N} \times |E|, \mathcal{H}_{!E}((n,e)) := \mathcal{H}_E(e)$$

$$\kappa_{!E}(x_0 \parallel x_1 \parallel \dots) := \boxtimes_{n\in\mathbb{N}} \kappa_E(x_n) = \begin{cases} -1 & \text{if } \exists n \in \mathbb{N}, \kappa_E(x_n) = -1 \text{ and } x_n \neq \varnothing \\ 0 & \text{if } \forall n \in \mathbb{N}, \kappa_E(x_n) = 0 \text{ or } x_n = \varnothing \\ +1 & \text{otherwise} \end{cases}$$

For E a positive quantum payoff $\sim$-game, we define $?E$ dually.

We have $(!E)^\perp = ?(E^\perp)$. For the payoff, note that even if $\kappa_E(\varnothing) \neq 0$, we have $\kappa_{!E}(\varnothing) = 0$. This represents the fact that since ! means "as many times as we want", we can always choose "zero times".

Proposition 8.3.9. *The category $(\sim\text{-}\mathbf{QGame}^\perp_\ominus, \boxtimes, \varnothing)$ of negative initially-non-losing[9] quantum payoff $\sim$-games and total anti-maps between them forms a linear exponential comonad up to $\simeq$.*

This follows directly from the fact that $\sim$-$\mathbf{Game}^\perp_\ominus$ forms a linear exponential comonad. Indeed, one just need to check that the morphisms of the linear exponential comonads are defined (the restriction to initially-non-losing payoff $\sim$-games is to ensure that all of them are indeed payoff non-decreasing), and the commutation of the diagrams exactly corresponds to the commutation of the diagrams in $\sim$-$\mathbf{Game}^\perp_\ominus$.

8.3.2 Quantum Arenas with Symmetry

In Section 5.5, we refined the notion of games to obtain a notion of arena, tailored to the QΛ language. We now extend this notion with symmetries.

Definition 8.3.10. *A $\sim$-arena is a quantum payoff $\sim$-game which is an arena[10] with trivial minimal symmetry:*

$$\theta : \{a\} \simeq \{a'\} \implies a = a'$$

[9] E is initially-non-losing if $\kappa_E(\varnothing) \geq 0$.

[10] So a positive, well-opened, tree-like and initially losing payoff quantum game. See Definition 5.5.1.

We say that a ∼-arena is affine if it is affine as an arena, i.e., every singleton configuration has payoff 0*.*

The "trivial minimal symmetry" condition makes it possible to lift the property of decomposition of quantum arenas to ∼-arenas.

Lemma 8.3.11. *Every ∼-arena A can be decomposed as $A = \bigoplus_{i\in I} \downarrow_{a_i:H_i} A_i$ with A_i some negative quantum payoff ∼-games.*

The operations $\otimes, \multimap, \multimap\!\!\bullet, (_)^\ell$ extend to ∼-arenas, with canonical configurations chosen as follows:

$$\underline{\mathbf{x} \otimes \mathbf{y}} := \underline{\mathbf{x}} \otimes \underline{\mathbf{y}} \qquad \underline{\mathbf{x} \multimap \mathbf{y}} := \underline{\mathbf{x}} \multimap \underline{\mathbf{y}}$$

$$\begin{cases} \underline{[\,]} & := \{\star\} \oplus \varnothing \\ \underline{[\mathbf{x_1}; ...; \mathbf{x_n}]} & := \varnothing \oplus \left(\underline{\mathbf{x_1}} \otimes \underline{[\mathbf{x_2}; ...; \mathbf{x_n}]}\right) \end{cases}$$

We now want to lift the linear exponential comonad ! on negative ∼-games to ∼-arenas. We note that we will only ever need ! to be defined on the ∼-arenas representing functions $A \multimap B$ (or $A \multimap\!\!\bullet\, B$), and tensor $\otimes$ of such ∼-arenas. For technical convenience, we will not try to define ! on every ∼-arena, and will restrict ourselves to the following notion of functional ∼-arena.

Definition 8.3.12. *A functional ∼-arena is a ∼-arena F with a single minimal event and*

$$\mathcal{H}_F(\min F) = \mathbf{1} \qquad \kappa_F(\{\min F\}) \geq 0$$

A functional ∼-arena can always be decomposed as $A = \downarrow_{m:\mathbf{1}} N$ with N a negative initially-non-losing game. In particular, for any A, B ∼-arenas, $(A \multimap B)$ and $(A \multimap\!\!\bullet\, B)$ are functional ∼-arenas.

Definition 8.3.13. *For $F = \downarrow_{m:\mathbf{1}} N$ a functional ∼-arena, we define $!F$ as the functional ∼-arena $\downarrow_{m:\mathbf{1}} !N$.*

Theorem 8.3.14. *The category $(\sim\text{-}\mathbf{QArena}^{\perp}_{\text{fun}}, \otimes, \mathbf{1})$ of functional ∼-arenas and anti-maps between them is an SMC, and has a linear exponential comonad ! up to $\simeq$*

This theorem follows from Proposition 8.3.9, as a functional ∼-arena is just a negative initially-non-losing quantum payoff ∼-game lifted by $\downarrow_{m:\mathbf{1}}$, and this lifting is actually a full and faithful functor from $\sim\text{-}\mathbf{QGame}^{\perp}_{\ominus}$ to $\sim\text{-}\mathbf{QArena}^{\perp}_{\text{fun}}$ which preserves all the structure.

8.4 Quantum Relations on Arenas with Symmetry

In this section, we define the category ∼-**QARel** of quantum ∼-arenas and quantum relations on ∼-arenas. We show it is a pre-model for LQΛ!, and then use it to model quantum

closures, giving rise to a sound and adequate model for LQΛ$_!$. The category $\sim$-**QARel** is an extension of **QARel** as defined in Section 6.3.3. We recall that objects of **QARel** are quantum payoff arenas and morphisms of **QARel**(A, B) are quantum valuations on pairs of configurations in $\mathcal{E}(A) \times \mathcal{E}(B)$ — keep in mind the correspondence between exhaustive configurations of arenas in **QARel** and points of the webs in **QRel**. When defining $\sim$-**QARel**, we will take quantum $\sim$-arenas as objects, and as morphisms in $\sim$-**QARel**(A, B) we will take quantum valuations on pairs of equivalence classes of exhaustive configurations of $\mathcal{E}_{\simeq}(A) \times \mathcal{E}_{\simeq}(B)$. This choice preserves the correspondence with the web in more standard relational models like the one of [PSV14]. As we detail in Definition 8.4.1, we expect those quantum valuations to be uniform with respect to the symmetry of the game.

8.4.1 Example

We consider the example $\mathrm{Map}(f, \ell)$. We recall its definition here:

$$\begin{array}{rcl} \mathrm{Map}(f,\ell) & := & \textbf{let rec } m\ \ell' = \\ & & \quad \textbf{match } \ell' \textbf{ with } \begin{array}{|lcl} | \ [] & \mapsto & [] \\ | \ x :: \ell'' & \mapsto & (f\ x) :: (m\ \ell'') \end{array} \\ & & \textbf{in } m\ \ell \end{array}$$

$$f : !(\textbf{qubit} \multimap \textbf{qubit}), \ell : \textbf{qubit}^{\ell} \vdash_{\mathbb{L}} \mathrm{Map}(f, \ell) : \textbf{qubit}^{\ell}$$

We describe in Fig. 8.5 the esp, $\simeq$, $\simeq^+$ and $\simeq^-$ of its $\sim$-arena. The canonical configurations $\underline{\ }$ are given by the definition of ! and $\multimap$, but could be arbitrarily chosen without significant consequences, as what matters is only their existence. Every canonical exhaustive configuration of the left hand side $\Gamma = !(\textbf{qubit} \multimap \textbf{qubit}) \otimes \textbf{qubit}^{\ell}$ is of the form:

$$x_{n,m} := \{(\mathbf{qb}_0)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0)^+_{\mathfrak{Q}}\} \parallel \ldots \parallel \{(\mathbf{qb}_n)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_n)^+_{\mathfrak{Q}}\} \parallel \{(\lambda, \mathbf{qb}^m)^+_{\mathfrak{Q}^{\otimes m}}\}$$

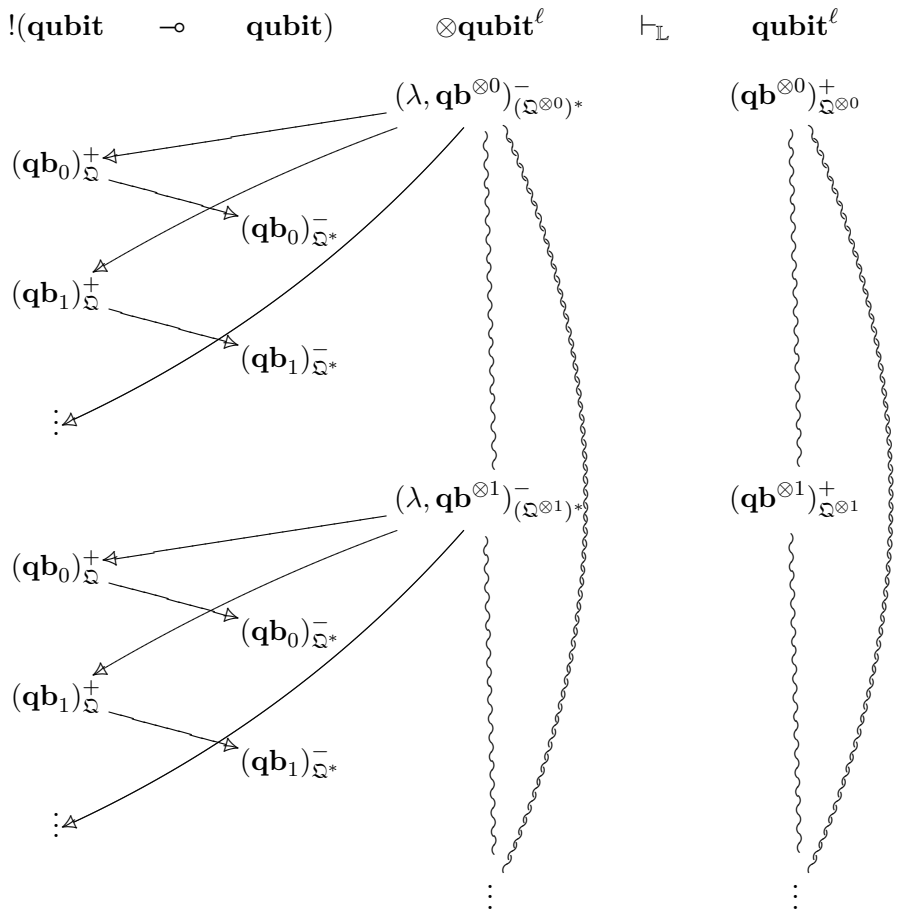
On the right hand side $\textbf{qubit}^{\ell}$, they are of the form:

$$y_k := \{(\lambda, \mathbf{qb}^k)^+_{\mathfrak{Q}^{\otimes k}}\}$$

We write $\mathbf{x_{n,m}}$ and $\mathbf{y_k}$ for their respective equivalence classes, and as said earlier, we will weight equivalence classes rather than single configurations. The quantum relation on $\sim$-arenas for Map is given as follows. In every case, we have

$$[\![\mathrm{Map}(f, \ell)]\!]\,(\mathbf{x_{n,m}}, \mathbf{y_k}) \in \overline{\mathbf{CPM}}((\mathfrak{Q}^* \otimes \mathfrak{Q})^{\otimes n} \otimes \mathfrak{Q}^{\otimes m}, \mathfrak{Q}^{\otimes k})$$

where $(\mathfrak{Q}^* \otimes \mathfrak{Q})^{\otimes n}$ corresponds to the n calls to the function f, $\mathfrak{Q}^{\otimes m}$ corresponds to the list ℓ of size m, and $\mathfrak{Q}^{\otimes k}$ corresponds to the output list of size k. Unless $m = n = k$, we have $[\![\mathrm{Map}(f, \ell)]\!]\,(\mathbf{x_{n,m}}, \mathbf{y_k}) = 0$. Indeed, the Map operator takes a list and returns a list of the same size, so we must have $m = k$, and calls the function as many times as the number of elements in the list, so we must have $n = m$.



Symmetry:

$\theta : x \simeq y$	whenever θ is an order-isomorphism and is the identity on minimal events
$\theta : x \simeq^{+} y$	whenever θ is an order-isomorphism and is the identity on minimal events
$\theta : x \simeq^{-} y$	whenever θ is an identity

Figure 8.5: $\sim$-Arena for $!(\mathbf{qubit} \multimap \mathbf{qubit}) \otimes \mathbf{qubit}^{\ell} \vdash_{\mathbb{L}} \mathbf{qubit}^{\ell}$

If $m = n = k$, we will have $[\![\mathrm{Map}(f,\ell)]\!](\mathbf{x_{n,m}}, \mathbf{y_k}) \in \overline{\mathbf{CPM}}((\mathfrak{Q}^* \otimes \mathfrak{Q})^{\otimes n} \otimes \mathfrak{Q}^{\otimes n}, \mathfrak{Q}^{\otimes n})$. The operational action of Map is to take every element of the list, to apply a function to it and then to return the list of outputs. Up to the compact closure of $\overline{\mathbf{CPM}}$, this $[\![\mathrm{Map}(f,\ell)]\!](x_{n,m}, y_k)$ is a morphism of $\overline{\mathbf{CPM}}((\mathfrak{Q}^* \otimes \mathfrak{Q})^{\otimes n}, (\mathfrak{Q}^{\otimes n})^* \otimes \mathfrak{Q}^{\otimes n})$.

Writing $\mathrm{split}^n_{A,B}$ for the natural transformation from $(A \otimes B)^{\otimes n}$ to $A^{\otimes n} \otimes B^{\otimes n}$ obtained from the braiding and the associator, we could be tempted to simply take $[\![\mathrm{Map}(f,\ell)]\!](\mathbf{x_{n,m}}, \mathbf{y_k})$ as $\Lambda(\mathrm{split}^n_{\mathfrak{Q}^*,\mathfrak{Q}})$, which is $\mathrm{split}^n_{\mathfrak{Q}^*,\mathfrak{Q}}$ up to compact closure. However, we want our semantics to satisfy some properties of uniformity with respect to replication, in particular we do not want to say that "we use the first copy of the function for the first element of the list, we use the second copy of the function for the second element of the list, …", and want instead to represent "we use any copy of the function for the first element of the list, we use any other copy of the function for the second element of the list, …". So we take:

$$[\![\mathrm{Map}(f,\ell)]\!](\mathbf{x_{n,m}}, \mathbf{y_k}) = \mathcal{H}_{[\![\Gamma]\!]}(\mathcal{A}_{[\![\Gamma]\!]}(x_{n,n})) \circ \Lambda(\mathrm{split}^n_{\mathfrak{Q}^*,\mathfrak{Q}})$$

8.4.2 The Category ∼-QARel

We define here the category ∼-**QARel** as the category with "the objects of ∼-**QA** and the morphisms of **QRel**", similarly to how **QARel** (see Section 6.3.3) is the category with "the objects of **QA** and the morphisms of **QRel**". Up to our knowledge, this category does not appear in the literature.

Definition 8.4.1. *We define the category* ∼-**QARel** *as follows:*

- *Its objects are quantum* ∼*-arenas*
- *Its morphisms* $\sigma \in$ ∼-$\mathbf{QARel}(A, B)$ *are* $\overline{\mathbf{CPM}}$*-weighted relations between equivalence classes of exhaustive configurations*

$$\sigma : ((\mathbf{x_A}, \mathbf{x_B}) \in \mathcal{E}_{\simeq}(A) \times \mathcal{E}_{\simeq}(B)) \mapsto \overline{\mathbf{CPM}}(\mathcal{H}_A(\underline{\mathbf{x_A}}), \mathcal{H}_B(\underline{\mathbf{x_B}}))$$

which are uniform, i.e., for every $(\mathbf{x_A}, \mathbf{x_B}) \in \mathcal{E}_{\simeq}(A) \times \mathcal{E}_{\simeq}(B)$*, we have*

$$\sigma(\mathbf{x_A}, \mathbf{x_B}) = \mathcal{H}_B(\mathcal{A}_B(\mathbf{x_B})) \circ \sigma(\mathbf{x_A}, \mathbf{x_B}) \circ \mathcal{H}_A(\mathcal{A}_A(\mathbf{x_A}))$$

As a comparison, [PSV14] defines the following category $\overline{\mathbf{CPMs}}^{\oplus}$:

- An object is a set $\mathfrak{A}$ called the web together with for every element $a \in \mathfrak{A}$ a pair (n_a, G_a) of an integer n_a and a group G_a of permutations of $\{1, \ldots, n\}$.
- A morphism $f \in \overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A}, \mathfrak{B})$ is a collection of morphisms $f_{a,b} \in \overline{\mathbf{CPM}}(\mathbb{C}^{n_a}, \mathbb{C}^{n_b})$ for $(a, b) \in \mathfrak{A} \times \mathfrak{B}$ which are uniform in the following sense:

$$f_{a,b} = \sum_{g \in G_a} \frac{\mathcal{H}(g)}{|G_a|} \circ f_{a,b} \circ \sum_{g' \in G_b} \frac{\mathcal{H}(g')}{|G_b|}$$

where $\mathcal{H}(g)$ for $g \in G_a$ is a permutation morphism of $\mathbf{CPM}(\mathbb{C}^{n_a}, \mathbb{C}^{n_a})$ corresponding to the permutation g.

Theorem 8.4.2. *There is a full and faithful functor from* $\sim$**-QARel** *to the category* $\overline{\mathbf{CPMs}}^{\oplus}$ *defined in [PSV14].*

This functor sends a $\sim$-arena A to the web $\mathcal{E}_{\simeq}(A)$, together with $n_{\mathbf{x_A}} = \dim(\mathcal{H}_A(\underline{\mathbf{x_A}}))$ and $G_{\mathbf{x_A}}$ being the group of permutations induced by $\mathcal{A}_A(\underline{\mathbf{x_A}})$. Its action on morphisms is simply pre-composing and post-composing by the isomorphism between the Hilbert spaces H and $\mathbb{C}^{\dim H}$.

The category **QARel** is a full subcategory of $\sim$-**QARel** (an arena is a $\sim$-arena with trivial symmetry). We can extend the structure of **QARel** (from Proposition 6.3.8) to $\sim$-**QARel**, and obtain the following.

Proposition 8.4.3. $(\sim\text{-}\mathbf{QARel}, \sim\text{-}\mathbf{QARel}, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ *is a non-trivial distributive CFC with a bottom.*

We now go through all the structure required in Section 7.2 to be a pre-model of LQΛ!.

The Lists

For every $\sim$-arena A, we have a $\sim$-arena A^ℓ such that $A^\ell = \mathbf{1} \oplus (A \otimes A^\ell)$.

The Quantum Primitives

We also have a $\sim$-arena $\mathbf{qubit} = \downarrow_{\mathbf{qb}:\mathfrak{Q}} \varnothing$, and some quantum relations on $\sim$-arenas $\mathbf{meas}^{\sim\text{-}\mathbf{QARel}}$, $\mathbf{new}^{\sim\text{-}\mathbf{QARel}}$, and $\mathbf{U}^{\sim\text{-}\mathbf{QARel}}$ defined from the morphisms of **Hilb** as below:

$$\begin{array}{rcl} \mathbf{meas}^{\sim\text{-}\mathbf{QARel}}(\{\star\}, \{b\}) & := & (\!|\mathbf{meas}_b^{\mathbf{Hilb}}|\!) \\ \mathbf{new}^{\sim\text{-}\mathbf{QARel}}(\{b\}, \{\star\}) & := & (\!|\mathbf{new}_b^{\mathbf{Hilb}}|\!) \\ \mathbf{U}^{\sim\text{-}\mathbf{QARel}}(\{q\}, \{q\}) & := & (\!|U|\!) \end{array}$$

Where $(\!|-|\!)$ is the functor from **Hilb** to **CPM** defined in Section 2.1.5.

The Functional Sub-SMC

We take $\mathcal{F}$ the full sub-SMC of $\sim$-**QARel** with all the functional $\sim$-arenas (see Definition 8.3.12), and note that $!A$ is always defined in this subcategory.

The Linear Exponential Comonad

We lift the linear exponential comonad from $\sim$-$\mathbf{QArena}^{\perp}_{\mathrm{fun}}$ into a linear exponential comonad for $\mathcal{F}$.

Definition 8.4.4. *For $f \in \sim\text{-}\mathbf{QArena}^{\perp}_{\mathrm{fun}}(A,B)$, we define its lifting $\widehat{f} \in \mathcal{F}(A,B)$ as follows:*

$$\widehat{f}(\mathbf{x_A},\mathbf{x_B}) := \begin{cases} \mathcal{H}_f(\underline{\mathbf{x_B}} \to f(\underline{\mathbf{x_B}})) \circ \mathcal{H}_A(\Re[f(\underline{\mathbf{x_B}})]) \circ \mathcal{H}_A\left(\mathcal{A}_A(\underline{\mathbf{x_A}})\right) & \text{if } f(\underline{\mathbf{x_B}}) \in \mathbf{x_A} \\ 0 & \text{otherwise} \end{cases}$$

where for every $x_A \in \mathcal{E}(A)$ we have arbitrarily chosen $\Re[x_A] : x_A \simeq \underline{\mathbf{x_A}}$.

While $\Re[f(\underline{\mathbf{x_B}})]$ appears in the definition, different choices for it have no influence on the resulting map for $\mathcal{F}$. Indeed, assume $\theta : f(\underline{\mathbf{x_B}}) \simeq \underline{\mathbf{x_A}}$, we have

$$\begin{aligned} \mathcal{H}_A(\Re[f(\underline{\mathbf{x_B}})]) \circ \mathcal{H}_A\left(\mathcal{A}_A(\underline{\mathbf{x_A}})\right) &= \mathcal{H}_A(\Re[f(\underline{\mathbf{x_B}})]) \circ \textstyle\sum_{\phi \in \mathcal{A}_A(\underline{\mathbf{x_A}})} \frac{\mathcal{H}_A(\phi)}{|\mathcal{A}_A(\underline{\mathbf{x_A}})|} \\ &= \mathcal{H}_A(\theta) \circ \textstyle\sum_{\phi \in \mathcal{A}_A(\underline{\mathbf{x_A}})} \frac{\mathcal{H}_A(\theta^{-1} \circ \Re[f(\underline{\mathbf{x_B}})] \circ \phi)}{|\mathcal{A}_A(\underline{\mathbf{x_A}})|} \\ &= \mathcal{H}_A(\theta) \circ \mathcal{H}_A\left(\mathcal{A}_A(\underline{\mathbf{x_A}})\right) \end{aligned}$$

To prove that this lifting is indeed a map of $\sim$-**QARel**, we have to prove that the annotation is preserved by auto-symmetries. It is trivial for auto-symmetries on the A side, and for auto-symmetries on the B side one must use the Proposition 8.3.5 and the diagram satisfied by $\mathcal{H}_f$.

This notion of lifting is actually inspired by a similar notion of lifting for strategies, defined in [CCW19], and used later in Section 9.3.2 in the context of quantum $\sim$-strategies.

Lemma 8.4.5 (Lifting lemma)**.** *The lifting $\widehat{-}$ is a symmetric monoidal functor from $(\sim\text{-}\mathbf{QArena}^{\perp}_{\mathrm{fun}}, \otimes, \mathbf{1})$ to $(\mathcal{F}, \otimes, \mathbf{1})$. Moreover,*

$$f \simeq f' \implies \widehat{f} = \widehat{f'}$$

Proposition 8.4.6. *The modality ! forms a linear exponential comonad $(!, \epsilon, \delta, \mathrm{w}, \mathrm{c}, \mathrm{m}, \mathrm{m_1})$ on $\mathcal{F}$.*

Proof. We simply lift all the morphisms from the linear exponential comonad of $\sim$-$\mathbf{QArena}^{\perp}_{\mathrm{fun}}$, and just need to provide a definition for the functor !.

For $\underline{\mathbf{x}} = \underline{\mathbf{x_1}} \parallel \ldots \parallel \underline{\mathbf{x_n}}$ (with only non-empty configurations), and $\underline{\mathbf{y}} = \underline{\mathbf{y_1}} \parallel \ldots \parallel \underline{\mathbf{y_m}}$ (with only non-empty configurations), we define $!\sigma(\mathbf{x},\mathbf{y}) = 0$ whenever $n \neq m$, and as follows whenever $n = m$:

$$!\sigma(\mathbf{x},\mathbf{y}) := \mathcal{H}_{!B}(\mathcal{A}_{!B}(\underline{\mathbf{y}})) \circ \left(\sum_{\substack{\iota \text{ perm} \\ \text{of } \{1,\ldots,n\}}} \bigotimes_{i=1}^{n} \mathrm{br}_\iota \circ \sigma(\mathbf{x_i}, \mathbf{y}_{\iota(\mathbf{i})}) \right) \circ \mathcal{H}_{!A}(\mathcal{A}_{!A}(\underline{\mathbf{x}}))$$

where br_ι is the permutation morphism corresponding to $\iota : \{1,\ldots,n\} \to \{1,\ldots,n\}$ obtained from the associator and braiding isomorphisms. All the diagrams but the naturality follows from the functoriality of the lifting, and their commutation up to $\simeq$ become commutations up to equality thanks to the lifting Lemma 8.4.5. We then

check the naturality diagrams without difficulties. □

The Recursor

We define the recursor $\mathbf{Y}$ through a supremum.

Definition 8.4.7. *For $f, g \in \sim\text{-}\mathbf{QARel}(A, B)$, we say that $f \leq g$ if for every $(\mathbf{x_A}, \mathbf{x_B}) \in \mathcal{E}_{\simeq}(A) \times \mathcal{E}_{\simeq}(B)$, $f(\mathbf{x_A}, \mathbf{x_B}) \sqsubseteq g(\mathbf{x_A}, \mathbf{x_B})$ for the Loewner order.*

Proposition 8.4.8. *The poset $(\sim\text{-}\mathbf{QARel}(A, B), \leq)$ is a dcpo with a minimal element*

$$\bot_{A,B} : (\mathbf{x_A}, \mathbf{x_B}) \mapsto 0$$

*This dcpo is an enrichment of $\sim$-**QARel**, i.e., all the operations of $\sim$-**QARel** are monotone and continuous for this dcpo.*

Continuity and monotonicity of structure of $\sim$-**QARel** for $\leq$ follows from the fact that the operations of $\overline{\mathbf{CPM}}$ are continuous and monotone for the Loewner order $\sqsubseteq$.

Definition 8.4.9. *For $f \in \sim\text{-}\mathbf{QARel}(W \otimes !(A^\perp \otimes B), A^\perp \otimes B)$ with W of the form $\bigotimes_i !F_i$, we define the operation*

$$\begin{array}{rccl} \mathcal{F}_f : & \sim\text{-}\mathbf{QARel}(W, !(A^\perp \otimes B)) & \to & \sim\text{-}\mathbf{QARel}(W, !(A^\perp \otimes B)) \\ & y & \mapsto & !f \circ \mathrm{dig}_{W \otimes !(A^\perp \otimes B)} \circ (W \otimes y) \circ \mathrm{contr}_{W,\mathbf{1},\mathbf{1}} \end{array}$$

And the recursor

$$\mathbf{Y}(f) := \lim_n \mathcal{F}_f^n \left(\bot_{W,!(A^\perp \otimes B)} \right)$$

This definition relies on Proposition 8.4.8, which ensures that $\mathcal{F}_f^n \left(\bot_{W,!(A^\perp \otimes B)} \right) \leq \mathcal{F}_f^m \left(\bot_{W,!(A^\perp \otimes B)} \right)$ whenever $n \leq m$.

Lemma 8.4.10. *The recursor $\mathbf{Y}$ satisfies the two axioms required in Section 7.2, i.e.,*

$$\mathbf{Y}(f) \circ f' = \mathbf{Y}(f \circ (f' \otimes !(A^\perp \otimes B)))$$

$$\mathbf{Y}(f) = !f \circ \mathrm{dig}_{W \otimes !(A^\perp \otimes B)} \circ (W \otimes \mathbf{Y}(f)) \circ \mathrm{contr}_{W,\mathbf{1},\mathbf{1}}$$

Proof. The second axiom is the definition of $\mathbf{Y}(f)$. The first axiom can be proved by induction on n for all the $\mathcal{F}_f^n \left(\bot_{W,!(A^\perp \otimes B)} \right)$, and using continuity of all the operations (Proposition 8.4.8), it is true for $\mathbf{Y}(f)$. □

It follows that $\sim$-**QARel** is a pre-model of LQΛ! as defined in Section 7.2, and in particular satisfies the invariance lemma for terms.

8.4.3 A Sound and Adequate Semantics for LQΛ!

To prove that **QARel** forms a sound and adequate model for LQΛ!, we first need to define the semantics of quantum closures. The definition is the same as for LQΛ.

Definition 8.4.11. *If* $\vdash_{\mathbb{L}} [q, \ell, t] : A$*, we define* $[\![[q, \ell, t]]\!]^{\vdash_{\mathbb{L}} A} \in \sim\text{-}\mathbf{QARel}(\mathbf{1}, [\![A]\!])$ *as follows:*

- *we know that we have* $\Delta \vdash_{\mathbb{L}} t : A$ *with* $\Delta = x_1 : \mathbf{qubit}, \ldots, x_n : \mathbf{qubit}$.
- *we recall that* $(\!|q|\!) \in \mathbf{CPM}(\mathbf{1}, \mathfrak{Q}^{\otimes n})$ *is defined in Section 2.3.3.*
- $[\![[q, \ell, t]]\!]\,(\{\star\}, \mathbf{x_A}) := [\![t]\!]\,(\{q\}, \mathbf{x_A}) \circ (\!|q|\!)$

and we then define $[\![\sum_i p_i [q_i, \ell_i, t_i]]\!]\,(\{\star\}, \mathbf{x_A}))$ *as* $\sum_i p_i\, [\![[q_i, \ell_i, t_i]]\!]\,(\{\star\}, \mathbf{x_A})$*, using the fact that the set* $\overline{\mathbf{CPM}}(\mathbf{1}, \mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}}))$ *is a completed positive convex cone.*

We can now extend the invariance lemma to closures.

Lemma 8.4.12 (Invariance)**.** *For every closures* $\Gamma \vdash_{\mathbb{L}} c : A$ *and* $\Gamma \vdash_{\mathbb{L}} d : A$

$$c \to d \implies [\![c]\!] = [\![d]\!]$$

Proof. We already proved in Lemma 7.2.9 that we had invariance for terms. We consider reduction rules over closures. Using the context factorisation Lemma 7.2.8, we obtain that for $!\Omega, \Gamma \vdash_{\mathbb{L}} t : A$ and $!\Omega, \Gamma, \Delta \vdash_{\mathbb{L}} E[t] : B$ we have

$$[\![E[s]]\!] = [\![E]\!] \circ ([\![s]\!] \otimes [\![!\Omega, \Delta]\!]) \circ \mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],[\![\Delta]\!]}$$

It follows that if we have $t \to s$, then we have:

$$[q, \ell, E[t]] \to [q, \ell, E[s]] \implies [\![[q, \ell, E[t]]]\!] = [\![[q, \ell, E[s]]]\!]$$

For the reduction of **new**, **meas** and **U**, we simply use the fact that the operational semantics uses morphisms of **Hilb** while the denotational semantics uses the corresponding morphisms of $\overline{\mathbf{CPM}}$.

The last two reductions relies on $\overline{\mathbf{CPM}}$ being a completed positive convex cone, and composition being linear for this cone. □

In order to prove adequacy, in the linear case, we used strong normalisation of the reductions. However, in LQΛ!, strong normalisation only holds for terms that do not have any **let rec**. In order to bypass this problem, we prove the following approximation lemma, that shows that every term with **let rec**s can be approximated in the semantics by a term without **let rec**s.

Lemma 8.4.13 (Approximation Lemma)**.** *For $\Gamma \vdash t : A$ a term, if we write t_n for the term where every* **let rec** *has been replaced by* **let rec** [**n**] *(as defined in Table 7.1), then*

$$[\![t]\!] = \lim_n [\![t_n]\!]$$

The same holds for closures.

Proof. We first note that $[\![\textbf{let rec } [0]\ f\ x = t\ \textbf{in}\ s]\!] = \bot_{[\![!\Omega,\Gamma]\!],C}$, so

$$\forall s,t,\forall n \in \mathbb{N}, [\![\textbf{let rec } [0]\ f\ x = t\ \textbf{in}\ s]\!] \leq [\![\textbf{let rec } [n]\ f\ x = t\ \textbf{in}\ s]\!]$$

We also note that

$$\begin{aligned}[\![\textbf{let rec } [n+1]\ f\ x = t\ \textbf{in}\ s]\!] \quad = \quad & [\![s]\!] \circ ([\![!\Omega,\Gamma]\!] \\ & \otimes [\![\lambda x.\textbf{let rec } [n]\ f\ x = t\ \textbf{in}\ t]\!]) \circ \mathrm{contr}_{[\![!\Omega]\!],[\![\Gamma]\!],\mathbf{1}}\end{aligned}$$

Using monotonicity of the structure (Proposition 8.4.8), and by induction on i we have:

$$\forall i \in \mathbb{N}, \forall s,t,\forall n \in \mathbb{N}, [\![\textbf{let rec } [i]\ f\ x = t\ \textbf{in}\ s]\!] \leq [\![\textbf{let rec } [n+i]\ f\ x = t\ \textbf{in}\ s]\!]$$

Using monotonicity of the structure again, we obtain that we always have $[\![t_n]\!] \leq [\![t_m]\!]$ whenever $n \leq m$, so the limit is well defined. We now want to prove that the limit is t. For that, we just check that for $\mathcal{F}_f$ defined in Definition 8.4.9, we have

$$\mathcal{F}_{[\![\lambda x.t]\!]}([\![\lambda x.\textbf{let rec } [n]\ f\ x = t\ \textbf{in}\ t]\!]) = [\![\lambda x.\textbf{let rec } [n+1]\ f\ x = t\ \textbf{in}\ t]\!]$$

(The proof of this equation is very similar to the proof of invariance by the recursion reduction rule). Since $[\![\lambda x.\textbf{let rec } [0]\ f\ x = t\ \textbf{in}\ t]\!] = \bot_{[\![!\Omega]\!],[\![!(A \multimap B)]\!]}$, and by uniqueness of limits, this proves

$$\forall t, \mathbf{Y}([\![\lambda x.t]\!]) = \lim_n [\![\lambda x.\textbf{let rec } [n]\ f\ x = t\ \textbf{in}\ t]\!]$$

Using continuity of the structure (Proposition 8.4.8), it follows that

$$\forall t,s, [\![\textbf{let rec } f\ x = t\ \textbf{in}\ s]\!] = \lim_n [\![\lambda x.\textbf{let rec } [n]\ f\ x = t\ \textbf{in}\ s]\!]$$

Then, by induction over the typing derivation of t, using $\lim_n \lim_m f_{m,n} = \lim_n f_{n,n}$ and using the continuity of the structure, we obtain:

$$[\![t]\!] = \lim_n [\![t_n]\!]$$

Using continuity of the structure, the same holds for closures. □

We now conclude with soundness and adequacy as in the case of LQΛ.

Theorem 8.4.14 (Soundness and Adequacy)**.** *For every term* $\vdash_{\mathbb{L}} t : \mathbf{1}$*, we have*

$$\mathbb{P}(t \Downarrow) = p \iff [\![t]\!] = [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$$

In particular, we have $[\![t]\!](\star, \star) \in \mathbf{CPM}(\mathbf{1}, \mathbf{1})$*, so finitary in* $\overline{\mathbf{CPM}}$*.*

Proof. If t does not contain a **let rec**, we use normalisation (Proposition 7.1.3) and write $t \to^* \sum_i p_i[q_i, \ell_i, ()] + \sum_j p'_j[q'_j, \ell'_j, \bot]$. Writing p for $\sum_i p_i$ we have that

$$\left[\!\!\left[\sum_i p_i[q_i, \ell_i, ()] + \sum_j p'_j[q'_j, \ell'_j, \bot] \right]\!\!\right] = [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$$

Using invariance, we obtain the expected equivalence.

If t does contain a **let rec**. We consider t_n which is t where all the **let rec** have been replaced by some **let rec** [**n**]. From the previous lemma, we know that $[\![t]\!] = \lim_n [\![t_n]\!]$. Using Lemma 7.1.5, we obtain that $\mathbb{P}(t \Downarrow) = \sup_n \mathbb{P}(t_n \Downarrow)$. The result immediately follows. □

Corollary 8.4.15. *For every pair of terms* $\Gamma \vdash_{\mathbb{L}} t : A$ *and* $\Gamma \vdash_{\mathbb{L}} s : A$*, we have*

$$[\![t]\!] = [\![s]\!] \implies t =_{\text{obs}} s$$

The proof is the same as the proof of Corollary 3.2.8. This model is in fact fully abstract, and the proof of full abstraction is a major contribution of this thesis. However, we will not prove it directly and only deduce it from the full abstraction of the game model in Section 9.3.3. Hence, we postpone the full abstraction theorem to Section 10.3. We note that the proof of full abstraction could be written in this model, assuming one first proves that for every term $\Gamma \vdash_{\mathbb{L}} t : A$, we have $[\![t]\!]$ finitary, *i.e.*, $\forall \mathbf{x_\Gamma} \in \mathcal{E}_{\simeq}([\![\Gamma]\!]), \forall \mathbf{x_A} \in \mathcal{E}_{\simeq}([\![A]\!]), [\![t]\!](\mathbf{x_\Gamma}, \mathbf{x_A}) \in \mathbf{CPM}(\mathcal{H}_{[\![\Gamma]\!]}(\underline{\mathbf{x_\Gamma}}), \mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}}))$. We refer to Proposition 43 of [PSV14] for a proof of finitaryness in the case of $\overline{\mathbf{CPMs}}^{\oplus}$.

8.4.4 Affine Case

In Definition 6.4.2, we defined an affine variant of **QARel**. We can do the same for $\sim$-**QARel**, with a definition of thunkability which is identical, relying on the fact that by definition of $\sim$-arenas, the equivalence classes of singleton configurations are trivial, so we always have $\underline{\{\mathbf{a}\}} = \{a\}$.

Definition 8.4.16. *The category* $\sim$-$\mathbf{QARel}^a$ *is the full subcategory of* $\sim$-**QARel** *restricted to only affine*[11] $\sim$*-arenas. A quantum relation on* $\sim$*-arenas* $\sigma \in \sim\text{-}\mathbf{QARel}^a(A, B)$ *is called thunkable if*

[11] *i.e.*, every singleton configuration is an exhaustive configuration.

- *for every* $a \in \min A$*, there exists a unique* $b \in \min B$*, such that* $\sigma(\{\mathbf{a}\},\{\mathbf{b}\}) \neq 0$*, and*
- *for every* $a \in \min A$*,* $b \in \min B$ *with* $\sigma(\{\mathbf{a}\},\{\mathbf{b}\}) \neq 0$*, we have*

$$\mathbf{Tr}_{\mathcal{H}_B(\{b\})} \circ \sigma(\{\mathbf{a}\},\{\mathbf{b}\}) = \mathbf{Tr}_{\mathcal{H}_A(\{a\})}$$

We write $\mathbf{QARel}_t^a$ *for the subcategory of thunkable maps.*

Proposition 8.4.17. *The category* $(\sim\text{-}\mathbf{QARel}^a, \sim\text{-}\mathbf{QARel}_t^a, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ *is an affine non-trivial distributive CFC with a bottom. It also forms a categorical model for AQ*Λ!*.*

The categorical model for AQΛ! uses the same lists, same quantum primitives, the category $\mathcal{F}$ is also the subcategory with only functional $\sim$-arenas, and it has the same ! and almost the same recursor $\mathbf{Y}$. The main difference is that $\bot_{A,B} \notin \sim\text{-}\mathbf{QARel}_t^a$, so we cannot take

$$\mathbf{Y}(f) := \lim_n \mathcal{F}_f^n \left(\bot_{W,!(A^\bot \otimes B)}\right)$$

Even though we do not have $\bot$ in $\sim\text{-}\mathbf{QARel}_t^a$, we still have a minimum, which is the morphism corresponding to the term $\lambda x.\bot$, so we can take:

$$\mathbf{Y}(f) := \lim_n \mathcal{F}_f^n \left([\![\lambda x.\bot]\!]\right)$$

The proofs are the same as in the case of $\sim$-**QARel**, except that we need to check that the structural morphisms of ! satisfy the thunkability condition, and that the recursor preserves thunkability. Using the same proofs as in the linear case, we obtain soundness and adequacy of $\sim$-**QARel** for AQΛ!. And similarly to the linear case, this model is also fully abstract, and the full abstraction will arise from the full abstraction of the game model as stated in Theorem 10.3.9.

Chapter 9

Game Semantics for the Quantum λ-calculus

9.1 Polarisation of the Symmetry

9.1.1 Uniformity of Strategies

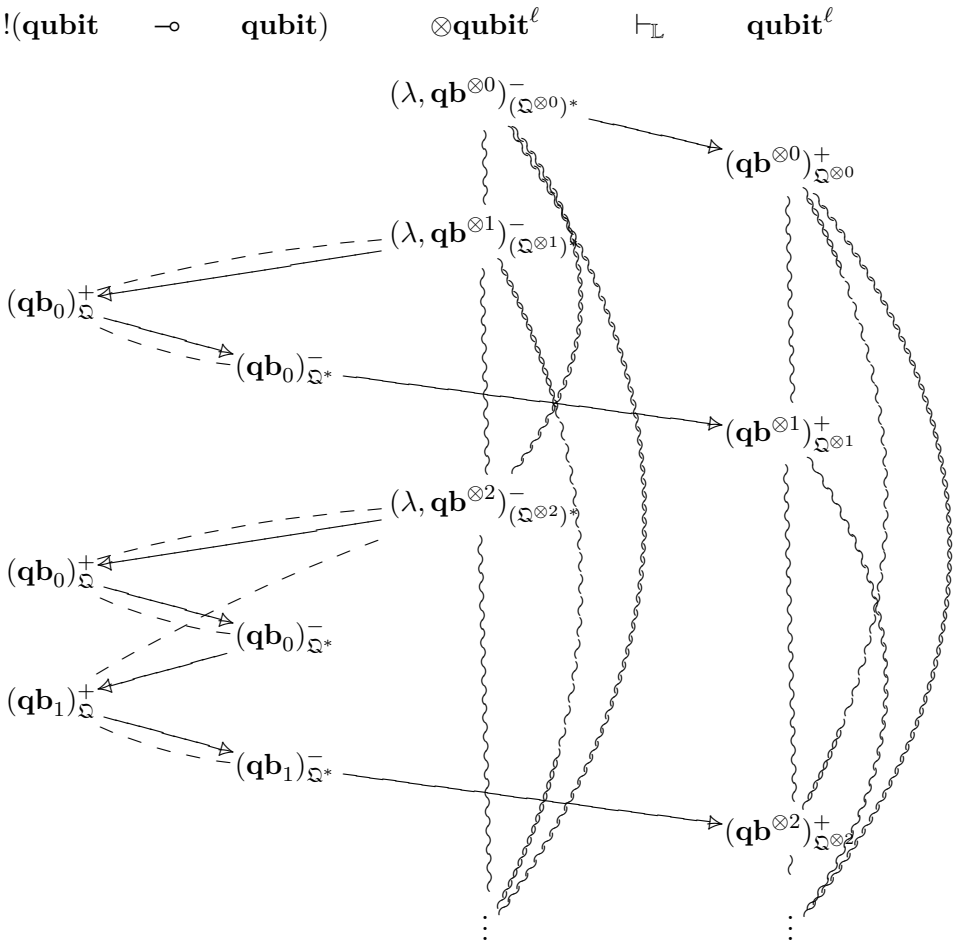
Ensuring uniformity of $\sim$-strategies is more subtle than ensuring uniformity of relations. This comes from a major difference in design philosophy between the two: in the relational case, we put quantum valuations on equivalence classes of configurations of the arena, meaning we put the annotation "after" having taken into account the symmetry; in the case of strategies, we will put quantum valuations on configurations of the strategy, meaning we put the annotation "before" having taken into account the symmetry. Accordingly, before defining quantum $\sim$-strategies, we start by simply considering quantum strategies over $\sim$-arenas, temporarily ignoring the symmetry of the arena.

9.1.2 The Map(-,-) Example

We look back at the $\mathrm{Map}(f, \ell)$ example, and we describe in Fig. 9.1 the esp of its strategy. We remind the reader that dashed lines represent causal dependencies from the game.

We choose to describe its quantum valuation through $\mathcal{Q}^{-,+}$ rather than $\mathcal{Q}$, as it is easier to explain. For x a configuration, the quantum valuation $\mathcal{Q}^{-,+}(x)$ is a composition and tensor of braiding, associator, trace and identity morphisms: the quantum information received through the event $(\lambda, \mathbf{qb}^{\otimes n})^-_{(\mathfrak{Q}^{\otimes n})^*}$ is fed to the events $(\mathbf{qb}_i)^+_{\mathfrak{Q}}$ as much as possible, the remainder being traced away, and the information received through the events $(\mathbf{qb}_i)^-_{\mathfrak{Q}^*}$ is fed to the event $(\lambda, \mathbf{qb}^{\otimes n})^+_{\mathfrak{Q}^{\otimes n}}$ if this event is in the configuration, or traced away if it is not.

But what about uniformity? The strategy we defined here arbitrarily chooses some copy indices for each function call: first the copy index 0, then 1, *etc.*, but it could have chosen



Term Represented:

$$\begin{array}{lll} \mathrm{Map}(f,\ell) & := & \textbf{let rec } m\ \ell' = \\ & & \quad \textbf{match } \ell' \textbf{ with } \begin{array}{|lcl} \mid [] & \mapsto & [] \\ \mid x :: \ell'' & \mapsto & (f\ x) :: (m\ \ell'') \end{array} \\ & & \textbf{in } m\ \ell \end{array}$$

$$f : !(\mathbf{qubit} \multimap \mathbf{qubit}), \ell : \mathbf{qubit}^\ell \vdash_{\mathbb{L}} \mathrm{Map}(f,\ell) : \mathbf{qubit}^\ell$$

Figure 9.1: Strategy for $\mathrm{Map}(f,\ell)$

to only use odd copy indices, or any other arbitrary choice. All the strategies that are the same as $[\![\mathrm{Map}(f,\ell)]\!]$ up to changing those copy indices are said to be *weakly isomorphic* to $[\![\mathrm{Map}(f,\ell)]\!]$. Already in the classical case [CCW19], it was a challenge to find conditions on strategies ensuring that weak isomorphism is a congruence. That is the purpose of "thin concurrent games" and in particular of the condition "thin". A lot of properties of the game model will only be satisfied up to weak isomorphism.

9.1.3 Another Example

The example $[\![\mathrm{Map}(f,\ell)]\!]$ had a replicable function in a contravariant position. While in the relational model there is no significant difference between replication in contravariant and covariant positions, this is not the case for the game model. So we consider the following example to illustrate the covariant case:

$$\vdash_{\mathbb{L}} \mathrm{Apply}_U() :!(\mathbf{qubit} \multimap \mathbf{qubit})$$

$$\mathrm{Apply}_U() := \lambda q^{\mathbf{qubit}}.\mathbf{U}\ q$$

We provide the esp for the strategy $[\![\mathrm{Apply}_U]\!]$ in Fig. 9.2. As previously, we choose to describe its quantum valuation through $\mathcal{Q}^{-,+}$ rather than $\mathcal{Q}$. For a non-empty $\oplus$-covered configuration x of the strategy, we have $x = \{\star^-, \lambda^+\} \sqcup \bigsqcup_{i\in I}\{(\mathbf{qb}_i)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_i)^+_{\mathfrak{Q}}\}$ for some $I \subseteq_{\mathrm{fin}} \mathbb{N}$, and

$$\mathcal{Q}^{-,+}(x) = \bigotimes_{i\in I} (\!|U|\!) \in \mathbf{CPM}(\mathfrak{Q}^{\otimes|I|}, \mathfrak{Q}^{\otimes|I|})$$

The annotation $\mathcal{Q}^{-,+}$ on a configuration which is not "non-minimal $\oplus$-covered" is given by the normalisation and the obliviousness property on quantum strategies.

In contrast to Fig. 9.1 where the strategy was selecting some copy indices from the game, in Fig. 9.2 the receptivity condition forces the strategy to acknowledge all the copy indices of the game.

What about uniformity? In this example, uniformity appears directly in the strategy: the esp is preserved by permutation of the copy indices, and so is the quantum valuation. In general, we enforce this constraint of uniformity by adding a symmetry to the esp of the strategy, which in this example is

$$\theta : x \simeq y \text{ whenever } \theta \text{ is an order-isomorphism}$$

and expect the quantum valuation to be uniform with respect to this symmetry. This will be made formal in Section 9.3.2.

9.1.4 Polarisation of Symmetry

As illustrated in the above examples, uniformity imposes very different requirement for replicable functions in covariant and contravariant position. This is why $\sim$-arenas come

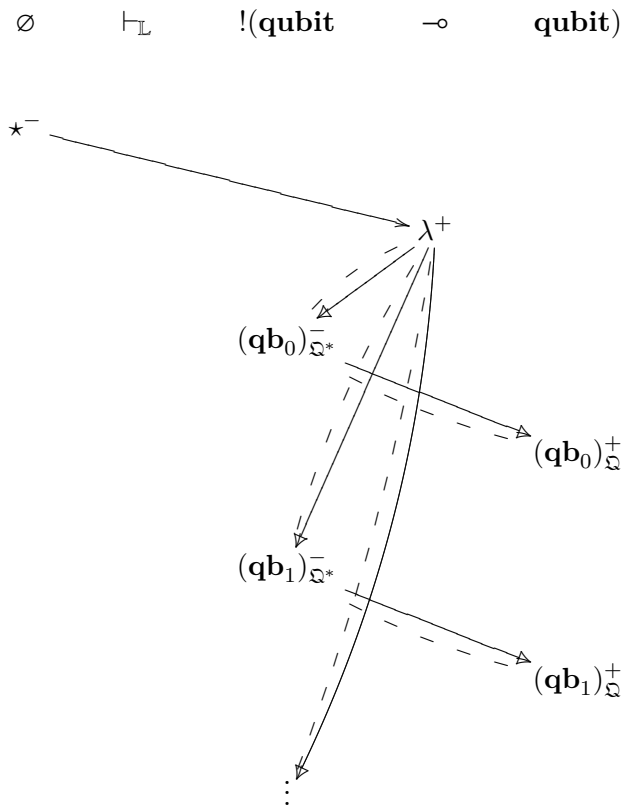


Figure 9.2: Strategy for Apply_U

with three symmetries: the negative symmetry $\simeq^-$ used by covariant replicable functions, the positive symmetry $\simeq^+$ used by contravariant replicable functions, and the non-polarised symmetry $\simeq$ which contains both. As those polarised symmetries are now relevant, it is time to give the full definition of $\sim$-games which we previously postponed in Definition 8.2.9.

Definition 9.1.1. *A $\sim$-game is an esp E together with*

- *three sets $\simeq_E, \simeq_E^+, \simeq_E^-$ called symmetry, positive symmetry and negative symmetry, such that $(E, \simeq_E)$, $(E, \simeq_E^+)$ and $(E, \simeq_E^-)$ are $\sim$-esps, and $\simeq_E^+$ and $\simeq_E^-$ are subsets of $\simeq_E$;*
- *a selection of canonical configurations $\underline{(-)} : \mathcal{C}_\simeq(E) \to \mathcal{C}(E)$ such that $\underline{\mathbf{x}} \in \mathbf{x}$ for every $\mathbf{x} \in \mathcal{C}_\simeq(E)$;*
- *an implicit total order on $\mathcal{C}_\simeq(E)$;*

satisfying the following properties:

Polarised *If $\theta \in \simeq^+$ and $\theta \in \simeq^-$, then θ is an identity symmetry.*

Positive extension *If $\theta \in \simeq^-$ and $\theta \subseteq^- \phi \in \simeq$ then $\phi \in \simeq^-$.*

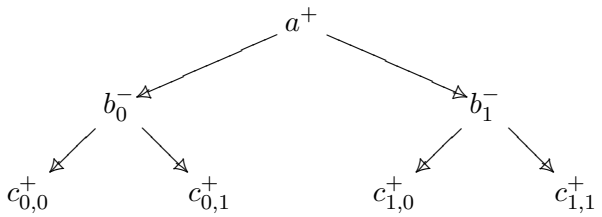
Negative extension *If $\theta \in \simeq^+$ and $\theta \subseteq^+ \phi \in \simeq$ then $\phi \in \simeq^+$.*

Representable *For every $\mathbf{x} \in \mathcal{C}_\simeq(E)$, we have*

$$\forall \phi : \underline{\mathbf{x}} \simeq_E \underline{\mathbf{x}}, \exists! \phi^+ : \underline{\mathbf{x}} \simeq_E^+ \underline{\mathbf{x}}, \exists! \phi^- : \underline{\mathbf{x}} \simeq_E^- \underline{\mathbf{x}}, \phi = \phi^+ \circ \phi^-$$

We recall that we write $\mathcal{A}_E(x)$, $\mathcal{A}_E^+(x)$ and $\mathcal{A}_E^-(x)$ for the sets of auto-symmetries, auto-positive-symmetries and respectively auto-negative-symmetries over x. Following the intuition of a symmetry as a valid change of copy indices, the conditions on $\sim$-games mean the following:

- "Polarised" is the central property of $\sim$-games: no change of copy indices can be made by both Player and Opponent alone, either the change can be made by Player alone (and not by Opponent alone), or the change can be made by Opponent alone (and not by Player alone), or the change requires an action of both Player and Opponent (and none can do it alone).
- "Positive extension" ensures that a valid change of copy indices that only changes copy indices of Player moves is a change that can be made by Player alone.
- "Negative extension" is the symmetric condition for Opponent.
- "Representable" is a technical condition that was added to ensure that $|\mathcal{A}_E(\underline{\mathbf{x}})| = |\mathcal{A}_E^+(\underline{\mathbf{x}})| \cdot |\mathcal{A}_E^-(\underline{\mathbf{x}})|$ (Lemma 9.1.3), which is a central property needed for the collapse of quantum $\sim$-strategies into quantum relations on $\sim$-arenas.



Symmetries:

$$\theta^- \in \simeq^- \quad : \quad \forall (e,e') \in \theta^-, e = e' \text{ or } \begin{cases} \exists i \in \{0,1\}, & (e,e') = (b_i^-, b_{1-i}^-), \text{ or} \\ \exists i,j \in \{0,1\}, & (e,e') = (c_{i,j}^+, c_{1-i,j}^+) \end{cases}$$

$$\theta^+ \in \simeq^+ \quad : \quad \forall (e,e') \in \theta^+, e = e' \text{ or } \quad \exists i,j \in \{0,1\}, (e,e') = (c_{i,j}^+, c_{i,1-j}^+)$$

$\theta \in \simeq$ whenever θ is an order-isomorphism

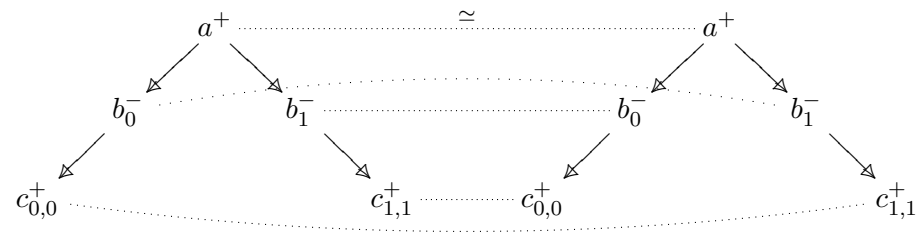
Figure 9.3: Example of $\sim$-game.

Indeed, while Lemma 8.2.4 ensures that $\forall x' \in \mathbf{x}, |\mathcal{A}_E(x')| = |\mathcal{A}_E(\underline{\mathbf{x}})|$, this is not always the case of $\mathcal{A}_E^+$ and $\mathcal{A}_E^-$, meaning that the choice of the canonical representative is important. We refer to [Cla20] for more technical motivation behind the representable condition.

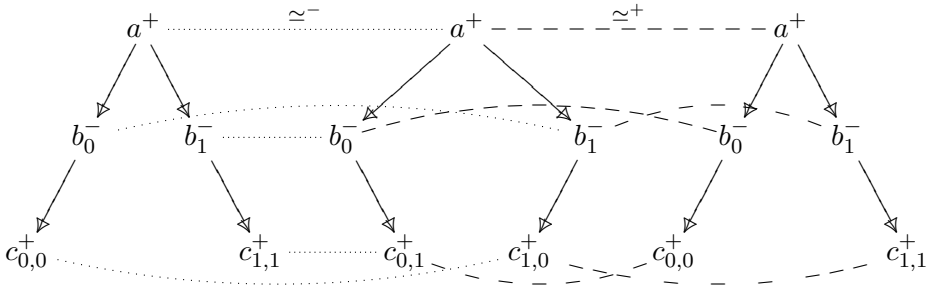
In the following, we try to give an intuition on the difference between a configuration which satisfies the representable condition (hence is a valid choice for a canonical configuration), and a configuration which does not. For that, we consider the example in Fig. 9.3. In this game, Opponent has control over the first copy index, while Player has control over the second. We can look at the following equivalence class for $\simeq$:

$$\{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,0}^+\} \simeq \{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,1}^+\} \simeq \{a, b_0^-, c_{0,1}^+, b_1^-, c_{1,0}^+\} \simeq \{a, b_0^-, c_{0,1}^+, b_1^-, c_{1,1}^+\}$$

This equivalence class corresponds to "taking a^+, two copies of b^-, and then one copy of c^+ for every copy of b^-". In this equivalence class, the first and the last representatives stand out as more "uniform", as we made consistent choices: whenever we had to take "one copy of c^+ for every copy of b^-", we always choose the first copy, or always the second copy. This "uniformity" is what the representability condition enforces. Let us consider a "non-uniform" configuration of this class, for example $\{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,1}^+\}$. This configuration has the following auto-symmetry:



There is only one way to decompose[1] this auto-symmetry into a negative symmetry followed by a positive symmetry, and it is the following:



Note that this auto-symmetry did not decompose into a negative auto-symmetry and a positive auto-symmetry, as the middle configuration is not the same as the left and right hand side configurations. So $\{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,1}^+\}$ does not satisfy the representability condition. On the cardinality side, we have

x	$\{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,0}^+\}$	$\{a, b_0^-, c_{0,0}^+, b_1^-, c_{1,1}^+\}$
$\lvert\mathcal{A}(x)\rvert$	2	2
$\lvert\mathcal{A}^+(x)\rvert$	1	1
$\lvert\mathcal{A}^-(x)\rvert$	2	1
$\lvert\mathcal{A}(x)\rvert = \lvert\mathcal{A}^-(x)\rvert \cdot \lvert\mathcal{A}^+(x)\rvert$	Yes	No

9.1.5 Miscellaneous Lemmas on Games with Symmetry

In this section, we mention two important lemmas in relation with the decomposition of the symmetry $\simeq$ into the positive symmetry $\simeq^+$ and the negative symmetry $\simeq^-$.

Lemma 9.1.2. *If $(E, \simeq, \simeq^+, \simeq^-, _)$ is a $\sim$-game, the function $(\theta^+, \theta^-) \mapsto \theta^+ \circ \theta^-$ forms an order-isomorphism between $\{(\theta^+, \theta^-) \in (\simeq^+ \times \simeq^-) \mid \mathrm{dom}(\theta^+) = \mathrm{codom}(\theta^-)\}$ and $\simeq$, with for order $\subseteq \times \subseteq$ on the left hand side, and $\subseteq$ on the right hand side.*

[1] In fact, Lemma 9.1.2 shows that every symmetry can be decomposed in a unique way into a negative symmetry followed by a positive symmetry.

We refer to lemma 3.19 of [CCW19] for a proof. This lemma ensures that every symmetry can be uniquely decomposed into a positive symmetry and a negative one. In particular, the morphisms ϕ^+ and ϕ^- of the representable condition in Definition 9.1.1 are necessarily unique if they exist. We note that the opposite lemma, which decomposes a symmetry uniquely into a negative symmetry and a positive one, is a direct corollary[2].

Lemma 9.1.3. *If $(E, \simeq, \simeq^+, \simeq^-, _)$ is a $\sim$-game, then for every $\mathbf{x} \in \mathcal{C}_{\simeq}(E)$, $|\mathcal{A}_E(\underline{\mathbf{x}})| = |\mathcal{A}^+_E(\underline{\mathbf{x}})| \times |A^-_E(\underline{\mathbf{x}})|$.*

Proof. The representable condition on $\sim$-games provides us with a bijection between $\mathcal{A}_E(\underline{\mathbf{x}})$ and $\mathcal{A}^+_E(\underline{\mathbf{x}}) \times A^-_E(\underline{\mathbf{x}})$. □

9.2 Strategies with Symmetry

In this section, we go through the definitions and some the basic properties of thin concurrent strategies as defined in [CCW19, Cla20], which we will refer to for proofs. We call them $\sim$-strategies. All the technical choices are motivated by obtaining a notion of *weak isomorphism* between $\sim$-strategies which is a congruence, and up to which ! behaves as an exponential.

9.2.1 Interactive Composition with Symmetry

In order to define $\sim$-strategies, we have to define the symmetry on the interaction $G \circledast F$ and on the interactive composition $G \odot F$ from the symmetries on G and F.

Definition 9.2.1. *For $f : F \to A^\perp \parallel B$ and $g : G \to B^\perp \parallel C$ two maps of $\sim$-esps, we define on the event structure $G \circledast F$ the following family of order-isomorphisms $\simeq_{G \circledast F}$:*

$$\theta : y \circledast x \simeq_{G \circledast F} y' \circledast x' \text{ whenever } \pi_G\, \theta : y \simeq_G y' \text{ and } \pi_F\, \theta : x \simeq_F x'$$

Similarly, we define on the esp $G \odot F$ the following family of order-isomorphisms:

$$\theta : y \odot x \simeq_{G \odot F} y' \odot x' \text{ whenever } \exists \phi : y \odot x \simeq_{G \odot F} y' \odot x' \text{ s.t } \theta \subseteq \phi$$

It is important to note that those families of isomorphisms might not give rise to a $\sim$-es and a $\sim$-esp as the extension property of symmetries might not be satisfied. This is an unfortunate problem, and leads to the fact that the category $\sim$-**ES** does not have all pullbacks (see [CCW19]). The notion of $\sim$-receptivity is a solution to this problem.

Definition 9.2.2. *A map of $\sim$-esps $f : A \to B$ is said to be*

[2] Using the fact that $\theta = \theta^- \circ \theta^+$ if and only if $\theta^{-1} = (\theta^+)^{-1} \circ (\theta^-)^{-1}$.

$$a^- \quad b^- \qquad \xrightarrow{\quad f \quad} \qquad a^- \quad b^-$$

	Symmetries:	
$\forall x, \mathbf{id}_x \in \simeq$		$\forall x, \mathbf{id}_x \in \simeq$
		$\{(a^-, b^-)\} \in \simeq$
		$\{(b^-, a^-)\} \in \simeq$
		$\{(a^-, b^-), (b^-, a^-)\} \in \simeq$

Figure 9.4: Map of $\sim$-esps which is not $\sim$-receptive.

$$a^- \quad b^- \qquad \xrightarrow{\quad f \quad} \qquad a^- \quad b^-$$

	Symmetries:	
$\forall x, \mathbf{id}_x \in \simeq$		$\forall x, \mathbf{id}_x \in \simeq$
$\{(a^-, b^-)\} \in \simeq$		$\{(a^-, b^-)\} \in \simeq$
$\{(b^-, a^-)\} \in \simeq$		$\{(b^-, a^-)\} \in \simeq$
$\{(a^-, b^-), (b^-, a^-)\} \in \simeq$		$\{(a^-, b^-), (b^-, a^-)\} \in \simeq$

Figure 9.5: Map of $\sim$-esps which is $\sim$-receptive.

$\sim$-Receptive *if whenever we have the first and the second following diagrams, there exists a unique* $a'^- \in |A|$ *such that we have the third diagram:*

$$\begin{array}{ccc} x & \xrightarrow[\simeq_A]{\theta} & x' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\ y & \xrightarrow[\simeq_B]{f\,\theta} & y' \end{array} \qquad \begin{array}{c} x \cup \{a^-\} \\ {\scriptstyle f}\downarrow \\ y \cup \{b^-\} \end{array} \xrightarrow[\simeq_B]{f\,\theta \cup \{(b,b')\}} y' \cup \{b'^-\} \qquad \begin{array}{ccc} x \cup \{a^-\} & \xrightarrow[\simeq_A]{\theta \cup \{(a,a')\}} & x' \cup \{a'^-\} \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\ y \cup \{b^-\} & \xrightarrow[\simeq_B]{f\,\theta \cup \{(b,b')\}} & y' \cup \{b'^-\} \end{array}$$

Similarly to how receptivity prevents a strategy from "forgetting" some Opponent moves, $\sim$-receptivity prevents a strategy from "forgetting" a symmetry between some Opponent moves. We provide in Fig. 9.4 an example of a map which is not $\sim$-receptive. The map f in this example, which is the identity on events, is a map of $\sim$-esps as it preserves the symmetry. However, it does not reflect the symmetry, and in particular it does not reflect the symmetry between a^- and b^-, which breaks the $\sim$-receptivity condition. The identity map is always $\sim$-receptive, as such the map in Fig. 9.5 is $\sim$-receptive.

One might find strange that in order to obtain a pullback in $\sim$-**ES**, we consider a condition which refers to polarities, but this is a core feature of the solution: we leverage the fact that during the interaction of $f : F \to A^\perp \parallel B$, and $g : G \to B^\perp \parallel C$, the map f has B in its codomain while the map g has $B^\perp$ in its codomain, so every event of B is negative on one side or the other.

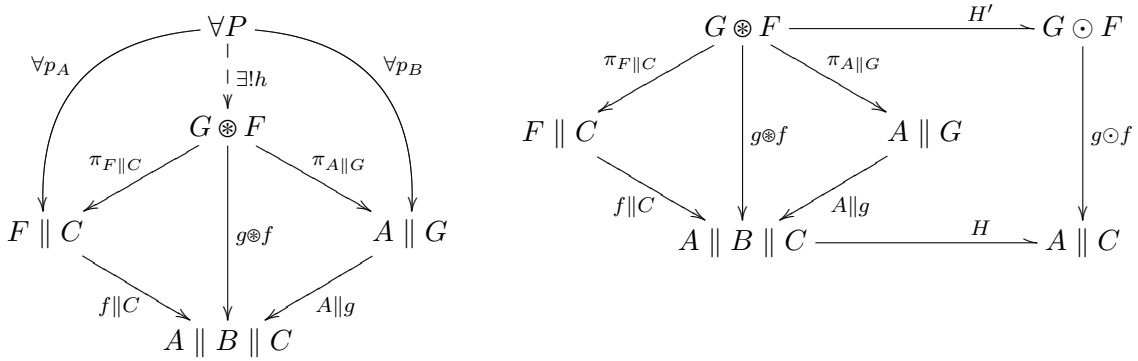


Figure 9.6: Diagrams for the interaction (left) and the interactive composition (right).

Lemma 9.2.3. *If both $f : F \to A^\perp \parallel B$ and $g : G \to B^\perp \parallel C$ are $\sim$-receptive maps of $\sim$-esp, then $G \circledast F$ together with $\simeq_{G\circledast F}$ as defined in Definition 9.2.1 is a $\sim$-es and forms a pullback in $\sim$-**ES**, which we sum up the diagram at the left of Fig. 9.6. It follows that the interaction is associative up to isomorphism.*

This lemma is a consequence of Lemma 3.12 of [CCW19]. Now that we have the interaction, we just need to extend the notion of hiding to $\sim$-es and we will have the interactive composition.

Definition 9.2.4. *A partial map of $\sim$-es $H : A \rightharpoonup B$ is said to be a hiding if it is a hiding map of event structures (see Definition 4.2.13) such that $\theta : x \simeq_B y$ if and only if there exists $\theta' : x' \simeq_A y'$ such that $H\,\theta' = \theta$.*

Lemma 9.2.5. *For $f : A \to B$ a map of $\sim$-es and $H : B \rightharpoonup B'$ a hiding, there exists some unique $\sim$-es A', hiding $H' : A \rightharpoonup A'$ and map $f' : A' \to B'$ such that:*

$$H \circ f = f' \circ H'$$

This lemma follows from Proposition 4.2.14.

Lemma 9.2.6. *If both $f : F \to A^\perp \parallel B$ and $g : G \to B^\perp \parallel C$ are $\sim$-receptive maps of $\sim$-esp, then $G \odot F$ together with $\simeq_{G\odot F}$ as defined in Definition 9.2.1 is a $\sim$-esp and the diagram at the right of Fig. 9.6 commutes with H, H' hiding maps. It follows that the interactive composition is associative up to isomorphism.*

This lemma is a consequence of Lemmas 9.2.3 and 9.2.5.

9.2.2 Strategies with Symmetry

We can now define the category of $\sim$-arenas and $\sim$-strategies. As in the case without symmetry, $\sim$-strategies will be receptive and courteous, but we also expect them to be $\sim$-receptive and thin. We refer to [CCW19] for the exact technical motivations behind the thin condition, and just mention that its main use is to ensure that the weak isomorphism defined in Definition 9.2.11 is a congruence, and it does so by making positive extensions "unique".

Definition 9.2.7. *A $\sim$-strategy from a $\sim$-game A to a $\sim$-game B is a map of $\sim$-esps $\sigma : S \to A^{\perp} \parallel B$ which is courteous, receptive, $\sim$-receptive and*

Thin *whenever $\theta : x \simeq_S y \multimap^{+} (y \cup \{s'^{+}\})$ there exists a unique s^{+} such that $\theta \cup \{(s^{+}, s'^{+})\} : (x \cup \{s^{+}\}) \simeq_S (y \cup \{s'^{+}\})$.*

Note that while S is a $\sim$-esps, A and B are $\sim$-games, *i.e.*, S has only one symmetry while A and B have a symmetry, a positive symmetry and a negative symmetry. As hinted before, when a part of the game is duplicated with a negative symmetry, all the copies will be present in the $\simeq$-strategy, while when a part of the game is duplicated with a positive symmetry, only the copies actively used are in the $\simeq$-strategy. We provide in Figs. 9.7 and 9.8 an example of a thin map, and an example of a non-thin map. The thin condition can be seen as a dual of $\sim$-receptivity:

- In Fig. 9.5, we see in the strategy two events a^{-} and b^{-} that are two copies of the same "action" (their corresponding move in the game are symmetric). Those are Opponent actions, which means that the strategy is bound to react uniformly to those two actions, which is why they are symmetric in the game.
- In Fig. 9.7, we see in the strategy two events a^{+} and b^{+}, which are also two copies of the same "action". Those two copies are not symmetric to each other, as there is nothing that bind Player to behave the same way on two different function calls it initialised. This is alike to how in a programming language, it is obviously possible to use the same function twice, in different contexts, and still behave differently afterwards.

Definition 9.2.8. *We define the copy-cat $\sim$-strategy $\mathrel{-\!\!\circ}_A : A \nrightarrow A$ as the copy-cat strategy together with the following symmetry on $\mathrel{-\!\!\circ}_A$:*

$$\theta : x \parallel y \simeq_{\mathrel{-\!\!\circ}_A} x' \parallel y' \text{ whenever } \theta = \theta' \parallel \theta'' \text{ with } \theta' : x \simeq_{A^{\perp}} x' \text{ and } \theta'' : y \simeq_A y'$$

Similarly to the case without symmetries, copy-cat will be the identity for the interactive composition up to "renaming of the events", which we call strong isomorphism. The interactive composition will also be associative up to strong isomorphism.

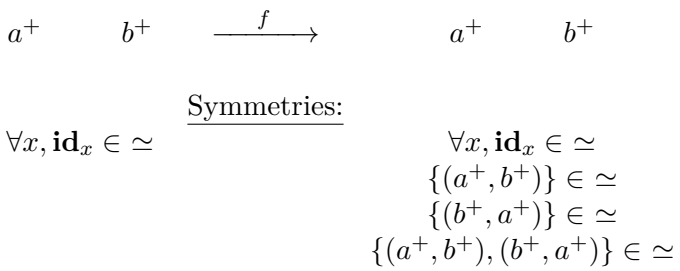


Figure 9.7: Map of $\sim$-esps which is thin.

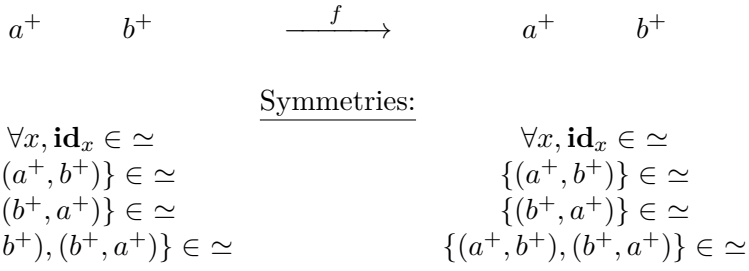


Figure 9.8: Map of $\sim$-esps which is not thin.

Definition 9.2.9. *We say that two $\sim$-strategies $\sigma : S \to A^\perp \parallel B$ and $\sigma' : S' \to A^\perp \parallel B$ are strongly isomorphic, and write $\sigma \cong \sigma'$, whenever there exists an isomorphism of $\sim$-esps $f : S \to S'$ which commutes with the strategies, i.e., $\sigma' \circ f = \sigma$.*

Similarly to the case without symmetry, $\sim$-strategies form a category up to strong isomorphism, more precisely:

Theorem 9.2.10. *For σ a $\sim$-strategy from A to B, and τ a $\sim$-strategy from B to C, $\tau \odot \sigma$ is a $\sim$-strategy from A to C. Moreover $\text{æ}_B \odot \sigma \cong \sigma \cong \sigma \odot \text{æ}_A$.*

We refer to [CCW19] for a proof. We will often use a weaker notion of isomorphism, which represents the fact that the two strategies are the same "up to symmetry", or in other words "up to renaming the events and changing the copy indices". In Fig. 9.9 we present three $\sim$-strategies $\sigma_0, \sigma_1, \sigma_2$ (we put in the same column the events and their image by the map of $\sim$-esps), all representing the term

$$f : !(\mathbf{1} \multimap \mathbf{bit}) \vdash_{\mathbb{L}} f\,() : \mathbf{bit}$$

We have σ_0 and σ_1 strongly isomorphic, as their only difference is the "name" of the events. We have σ_1 and σ_2 weakly isomorphic, as σ_0 uses the "first copy" of f while σ_1 uses the "second copy" of f. Formally, weak isomorphism is defined as follows.

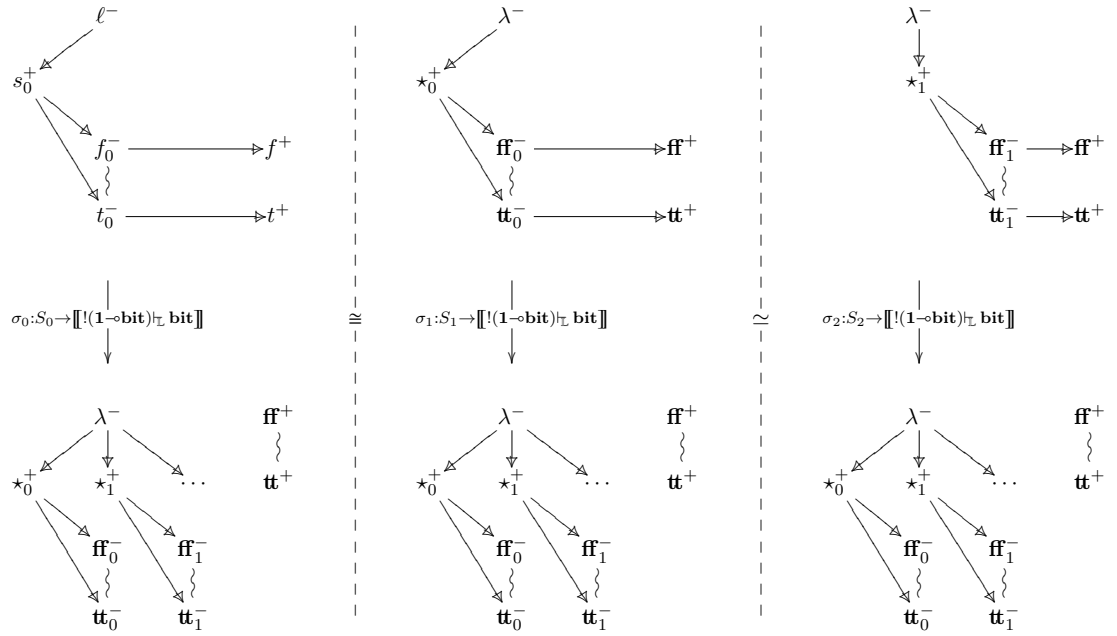


Figure 9.9: Examples of strong and weak isomorphisms

Definition 9.2.11. *Two $\sim$-strategies $\sigma : S \to A^\perp \parallel B$ and $\sigma' : S' \to A^\perp \parallel B$ are said weakly isomorphic, and we write $\sigma \simeq \sigma'$, if S and S' are isomorphic $\sim$-esps and the isomorphism commutes with the strategy up to $\simeq^+_{A^\perp \parallel B}$; i.e., there exists an invertible map of $\sim$-esps $f : S \to S'$:*

$$\forall x \in \mathcal{C}(S), \{(\sigma' \circ f)(e), \sigma(e) \mid e \in x\} : (\sigma' \circ f)\, x \simeq^+_{A^\perp \parallel B} \sigma\, x$$

In other words, two strategies are weakly isomorphic whenever they only differ by the copy indices that Player chooses. We note that if σ and σ' are strongly isomorphic, then they are weakly isomorphic too.

Proposition 9.2.12. *Two $\sim$-strategies $\sigma : S \to A^\perp \parallel B$ and $\sigma' : S' \to A^\perp \parallel B$ are weakly isomorphic, if and only if:*

Weak Equivalence *There exist two maps of $\sim$-esps $f : S \to S'$ and $g : S' \to S$ such that, for $\simeq$ the congruence on $\sim$-**ESP** we have:*

$$\sigma \circ g \simeq \sigma' \qquad g \circ f \simeq \mathbf{id}_S$$
$$\sigma' \circ f \simeq \sigma \qquad f \circ g \simeq \mathbf{id}_{S'}$$

This is a consequence of Corollary 3.30 of [CCW19]. The direction "weakly isomorphic $\Rightarrow$ weakly equivalent" is trivial as we can keep the same isomorphism f between S and

S' and take $g = f^{-1}$. The direction "weakly equivalent $\Rightarrow$ weakly isomorphic" requires more work as we have to build an isomorphism from f and g that are only inverse up to symmetry.

Lemma 9.2.13. *Weak isomorphism is a congruence, i.e., for σ, σ' two $\sim$-strategies from A to B, and τ, τ' two $\sim$-strategies from B to C, we have*

$$\sigma \simeq \sigma' \text{ and } \tau \simeq \tau' \implies \tau \odot \sigma \simeq \tau' \odot \sigma'$$

This is a consequence of proposition 3.40 of [CCW19], and note that the proof of this proposition relies on the thin condition of $\sim$-strategies.

Proposition 9.2.14. *$\sim$-games and $\sim$-strategies, up to strong isomorphism, form a CpCC $(\sim\text{-}\mathbf{Strat}, \|, \varnothing, (_)^\perp)$. The weak isomorphism is a congruence for this additional structure.*

The bifunctor $\|$ on $\sim$-strategies is deduced from the bifunctor $\|$ on maps of $\sim$-esps as in the case without symmetry. We refer to theorem 3.42 of [CCW19] for a proof of the compact closure.

9.3 Quantum Strategies with Symmetry

9.3.1 Example

We previously considered the example $\mathrm{Map}(f, \ell)$ where the replicable function is in contravariant position, and the example $\mathrm{Apply}_U()$ where the replicable function is in covariant position. We now consider $\mathrm{Square}(f) := \lambda x^{\mathbf{qubit}}.f(f(x))$ where we have both.

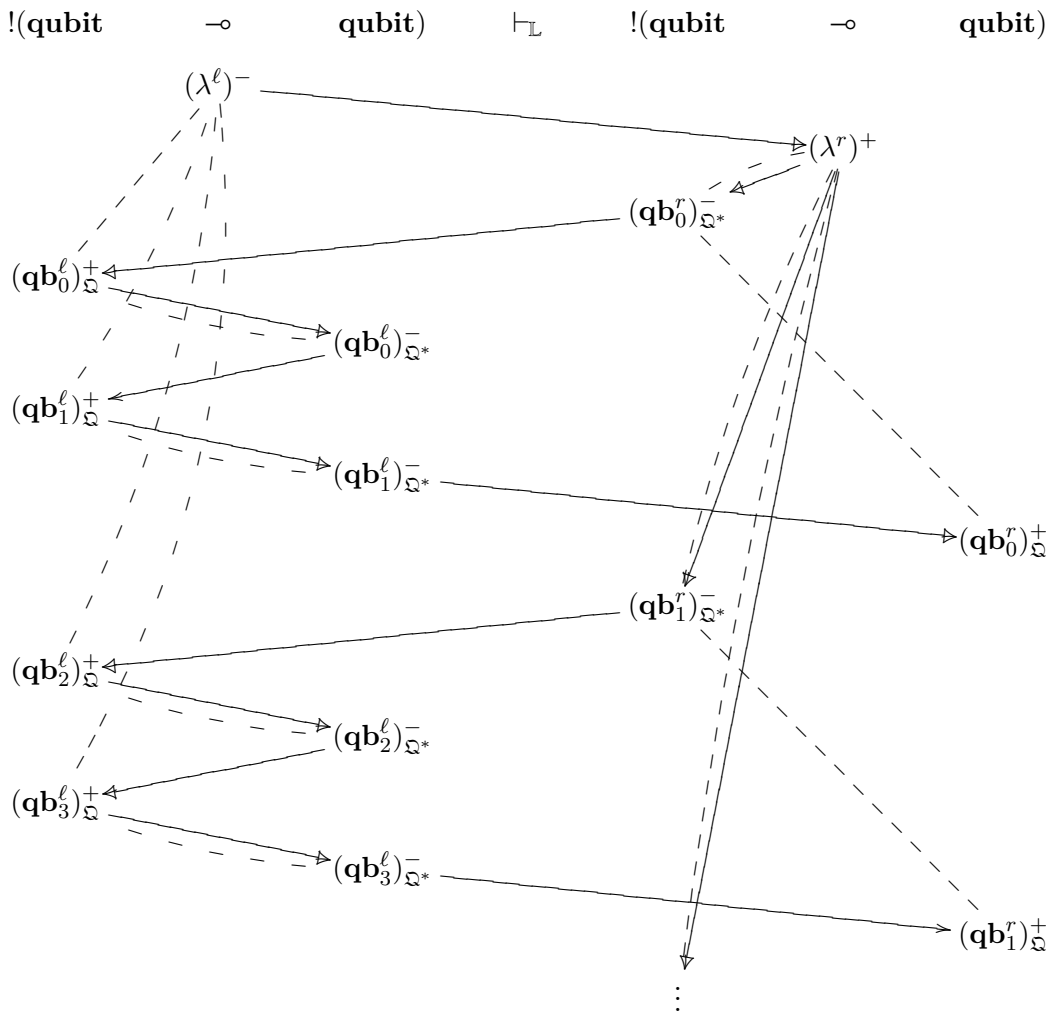
We describe its associated esp in Fig. 9.10. For each function call on the right hand side, there are two function calls on the left hand side. The symmetry of this $\sim$-strategy simply contains all the order-isomorphisms. The quantum valuation $\mathcal{Q}^{-,+}$ on $\oplus$-covered configurations is simply the identity, as the quantum data from $(\mathbf{qb}^r_n)^-_{\mathfrak{Q}^*}$ is fed into $(\mathbf{qb}^\ell_{2n})^+_{\mathfrak{Q}}$, the quantum data from $(\mathbf{qb}^\ell_{2n})^-_{\mathfrak{Q}^*}$ is fed into $(\mathbf{qb}^\ell_{2n+1})^+_{\mathfrak{Q}}$, and the quantum data of $(\mathbf{qb}^\ell_{2n+1})^-_{\mathfrak{Q}^*}$ is fed into $(\mathbf{qb}^r_n)^+_{\mathfrak{Q}}$.

In this example, we note once again that the function in covariant position has all its copies symmetric in the strategy, because the strategy must answer uniformly to Opponent calling replicable functions. However, the function in contravariant position has no uniformity restriction, and the strategy arbitrarily chooses which copy indices to use. Different choices of copy indices lead to weakly isomorphic strategies.

9.3.2 Quantum Strategies with Symmetry

Definition 9.3.1. *A quantum $\sim$-strategy $\sigma : A \nrightarrow B$ is a $\sim$-strategy between quantum payoff $\sim$-games together with a quantum valuation $\mathcal{Q}_\sigma$ on configurations $x \in \mathcal{C}(S)$ such that:*

$$\sigma\, x = x_A \parallel x_B \implies \mathcal{Q}_\sigma(x) \in \mathbf{CPM}(\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$$



Term Represented:

$f :!(\mathbf{qubit} \multimap \mathbf{qubit}) \vdash_{\mathbb{L}} \mathrm{Square}(f) :!(\mathbf{qubit} \multimap \mathbf{qubit})$

$\mathrm{Square}(f) := \lambda x^{\mathbf{qubit}}.f(f(x))$

Figure 9.10: Strategy for Square(f)

Quantum strategy *The normalisation, obliviousness and drop conditions of Definition 5.3.3 are satisfied.*

Winning x *$\oplus$-covered* $\implies \kappa_{A^\perp \parr B}(\sigma\, x) \geq 0$.

Uniform $\theta : x \simeq_S x' \implies \mathcal{Q}_\sigma(x') = \mathcal{H}_B(\theta_B^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\theta_A)$, *where* $\sigma\, \theta = \theta_A \parallel \theta_B$.

An important observation is that the "Uniform" condition is applied to the symmetry in the strategy, not to the symmetry in the $\sim$-game. In the example of Square(f) Fig. 9.10, if we consider the configuration $\{(\lambda^\ell)^-, (\lambda^r)^+, (\mathbf{qb}_0^r)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0^\ell)^+_{\mathfrak{Q}}, (\mathbf{qb}_1^r)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_2^\ell)^+_{\mathfrak{Q}}\}$ there is an auto-symmetry in the strategy that exchanges $(\mathbf{qb}_0^r)^-_{\mathfrak{Q}^*}$ with $(\mathbf{qb}_1^r)^-_{\mathfrak{Q}^*}$ and $(\mathbf{qb}_0^\ell)^+_{\mathfrak{Q}}$ with $(\mathbf{qb}_2^\ell)^+_{\mathfrak{Q}}$, so the quantum valuation must be preserved by this auto-symmetry. However, if we consider $\{(\lambda^\ell)^-, (\lambda^r)^+, (\mathbf{qb}_0^r)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0^\ell)^+_{\mathfrak{Q}}, (\mathbf{qb}_0^\ell)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_1^\ell)^+_{\mathfrak{Q}}, (\mathbf{qb}_1^\ell)^-_{\mathfrak{Q}^*}, (\mathbf{qb}_0^r)^+_{\mathfrak{Q}}\}$, there is no non-trivial auto-symmetry in the strategy, and in particular the auto-symmetry of the game that exchanges $(\mathbf{qb}_0^\ell)^+_{\mathfrak{Q}}$ with $(\mathbf{qb}_1^\ell)^+_{\mathfrak{Q}}$ and $(\mathbf{qb}_0^\ell)^-_{\mathfrak{Q}^*}$ with $(\mathbf{qb}_1^\ell)^-_{\mathfrak{Q}}$ is not reflected in the strategy, hence the uniformity condition does not apply here. We mentioned earlier that the choice of copy indices by Player was arbitrary, and that different choices lead to weakly isomorphic strategies. The weak and strong isomorphisms are defined as follows:

Lemma 9.3.2. *For $\sigma, \sigma' : A \nrightarrow B$ two $\sim$-strategies and $f : S \to S'$ a map of $\sim$-esps such that $\sigma \simeq \sigma' \circ f$ as maps of $\sim$-esps, then for every $x \in \mathcal{C}(S)$ we have an induced symmetry $\Psi_A^f(x) \parallel \Psi_B^f(x) : \sigma\, x \simeq_{A^\perp \parallel B} \sigma'\,(f\, x)$ such that the following diagram commutes:*

$$\begin{array}{ccc} x & \xrightarrow{\quad f \quad} & f\, x \\ {\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle\sigma'} \\ \sigma\, x & \xrightarrow[\Psi_A^f(x) \parallel \Psi_B^f(x)]{} & \sigma\,(f\, x) \end{array}$$

Definition 9.3.3. *For two quantum $\sim$-strategies $\sigma, \sigma' : A \nrightarrow B$, we say that a map of $\sim$-esp $f : S \to S'$ preserves the quantum valuation whenever:*

$$\mathcal{Q}_{\sigma'}(f\, x) = \mathcal{H}_B((\Psi_B^f(x))^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\Psi_A^f(x))$$

The quantum $\sim$-strategies σ and σ' are said

Strongly Isomorphic *Whenever they are strongly isomorphic as $\sim$-strategies with associated isomorphism f, and f preserves quantum valuations.*

Weakly Isomorphic *Whenever they are weakly isomorphic as $\sim$-strategies with associated isomorphism f, and f preserves quantum valuations.*

Weakly Equivalent *Whenever they are are weakly equivalent as $\sim$-strategies with associated maps $f : S \to S'$ and $g : S' \to S$, and both f and g preserves quantum valuation.*

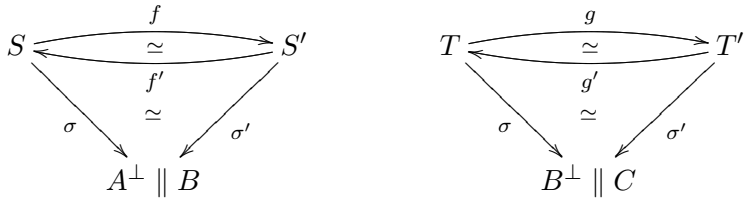
Lemma 9.3.4. *Strong isomorphism implies weak isomorphism. Weak isomorphism and weak equivalence are equivalent.*

This follows from Proposition 9.2.12 and Lemma 3.29 of [CCW19].

We take the same quantum valuation for copy-cat $\mathfrak{C}_A$, for the interactive composition $\odot$, for the two parallel composition $\boxtimes$ and $\parr$, and for the negation $(_)^\perp$ as in the case without symmetry. We obtain the following result:

Proposition 9.3.5. *Quantum payoff $\sim$-games and quantum $\sim$-strategies, up to strong isomorphism, form two SMCs $(\sim\text{-}\mathbf{QCG}, \parr, \varnothing)$ and $(\sim\text{-}\mathbf{QCG}, \boxtimes, \varnothing)$. In fact, $(\sim\text{-}\mathbf{QCG}, \| , \boxtimes, \varnothing, (_)^\perp)$ forms a linearly distributive category with negation, so a $\star$-autonomous category in light of their equivalence. Weak isomorphism is a congruence in $\sim$-$\mathbf{QCG}$.*

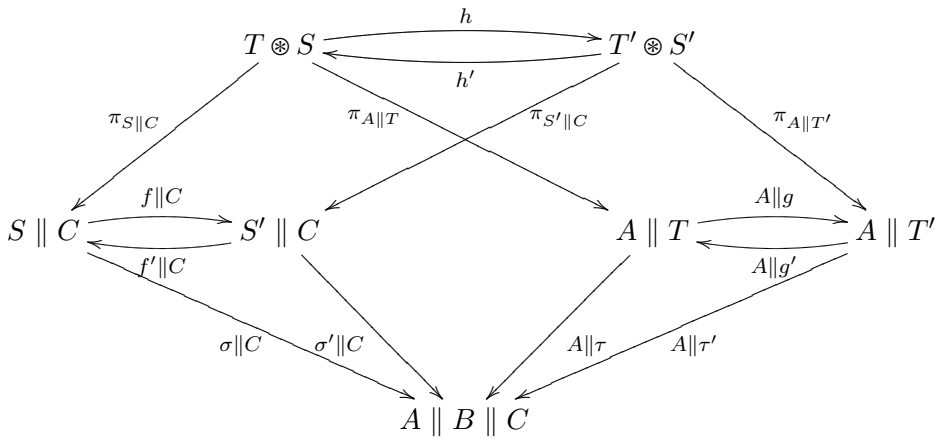
Proof. Most of those claims can be deduced from the case with quantum valuation but without symmetry together with the case with symmetry but without quantum valuations. We prove the congruence of weak isomorphism. We consider $\sigma \simeq \sigma' \in \sim\text{-}\mathbf{QCG}(A, B)$ and $\tau \simeq \tau' \in \sim\text{-}\mathbf{QCG}(B, C)$. By definition of weak equivalence, it means we have $f : S \to S'$, $f' : S' \to S$, $g : T \to T'$ and $g' : T' \to T$ maps of $\sim$-esps such that



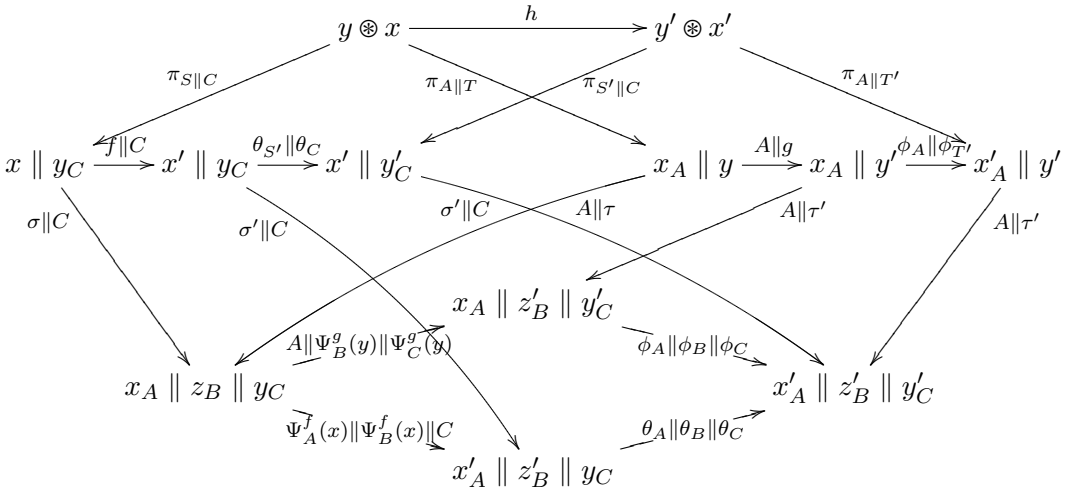
$$\mathcal{Q}_{\sigma'}(f\, x) = \mathcal{H}_B((\Psi_B^f(x))^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\Psi_A^f(x))$$
$$\mathcal{Q}_\sigma(f'\, x') = \mathcal{H}_B((\Psi_B^{f'}(x'))^{-1}) \circ \mathcal{Q}_{\sigma'}(x') \circ \mathcal{H}_A(\Psi_A^{f'}(x'))$$
$$\mathcal{Q}_{\tau'}(g\, y) = \mathcal{H}_C((\Psi_C^g(y))^{-1}) \circ \mathcal{Q}_\tau(y) \circ \mathcal{H}_B(\Psi_B^g(y))$$
$$\mathcal{Q}_\tau(g'\, y') = \mathcal{H}_C((\Psi_C^{g'}(y'))^{-1}) \circ \mathcal{Q}_{\tau'}(y') \circ \mathcal{H}_B(\Psi_B^{g'}(y'))$$

We want to prove that $\tau \odot \sigma \simeq \tau' \odot \sigma'$. Using Lemma 3.21 from [CCW19], We obtain two maps of $\sim$-es $h : T \circledast S \to T' \circledast S'$ and $h' : T' \circledast S' \to T \circledast S$ such that the following

diagram commutes up to $\simeq$:



In particular for $y \circledast x \in \mathcal{C}(T \circledast S)$, the following diagram commutes



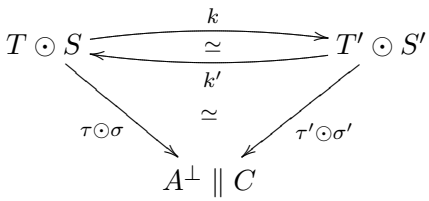
From that diagram it follows that

$$\begin{array}{rcl}
\mathcal{Q}_{\tau'\circledast\sigma'}(y' \circledast x') & = & \mathcal{Q}_{\tau'}(y') \circ \mathcal{Q}_{\sigma'}(x') \\
& = & \mathcal{H}_C((\phi_C \circ \Psi^g_C(y))^{-1}) \circ \mathcal{Q}_\tau(y) \circ \mathcal{H}_B(\phi_B \circ \Psi^g_B(y)) \\
\circ & & \mathcal{H}_B((\theta_B \circ \Psi^f_B(x))^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\theta_A \circ \Psi^f_A(x)) \\
& = & \mathcal{H}_C(\theta_C^{-1}) \circ \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\phi_A) \\
& = & \mathcal{H}_C(\theta_C^{-1}) \circ \mathcal{Q}_(\tau \circledast \sigma)(y \circledast x) \circ \mathcal{H}_A(\phi_A)
\end{array}$$

Symmetrically, for $y' \circledast x' \in \mathcal{C}(T' \circledast S')$ and $y \circledast x = h'\,(y' \circledast x')$ we obtain

$$\mathcal{Q}_{\tau\circledast\sigma}(y \circledast x) = \mathcal{H}_C((\theta'_C)^{-1}) \circ \mathcal{Q}_{\tau\circledast\sigma}(y' \circledast x') \circ \mathcal{H}_A(\phi'_A)$$

Using hiding (Lemma 9.2.5), we obtain two maps of $\sim$-esps $k : T \odot S \to T' \odot S'$ and $k' : T' \odot S' \to T \odot S$ such that



$$\mathcal{Q}_{\tau'\odot\sigma'}(k\,z) = \mathcal{H}_C((\Psi^k_B(z))^{-1}) \circ \mathcal{Q}_{\tau\odot\sigma}(z) \circ \mathcal{H}_A(\Psi^k_A(z))$$
$$\mathcal{Q}_{\tau\odot\sigma}(k'\,z') = \mathcal{H}_C((\Psi^{k'}_B(z'))^{-1}) \circ \mathcal{Q}_{\tau\odot\sigma}(x') \circ \mathcal{H}_A(\Psi^{k'}_A(z'))$$

So $\tau \odot \sigma \simeq \tau' \odot \sigma'$. □

9.3.3 Categorical Model for LQ$\Lambda_!$

If A and B are quantum $\sim$-arenas, we define negative and thunkable quantum $\sim$-strategies as in Definition 5.5.1, and visible quantum strategies as in Definition 6.2.2. We write $\sim$-**QA** and $\sim$-$\mathbf{QA}_t$ the categories (up to strong isomorphism) of quantum $\sim$-arenas and respectively negative visible quantum $\sim$-strategies and negative thunkable visible quantum $\sim$-strategies. In Section 8.3.2, we already extended the operations $\multimap, \oplus, (_)^\ell$ to quantum $\sim$-arenas.

Proposition 9.3.6. $(\sim\text{-}\mathbf{QA}, \sim\text{-}\mathbf{QA}_t, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$*, up to strong isomorphism, a non-trivial CFC with a bottom. Weak isomorphism is a congruence.*

The proof is the same as in the case without symmetry. To prove that it is a pre-model of LQ$\Lambda_!$ as defined in Section 7.2, we recall that we need lists, quantum primitives, a functional sub-SMC, a linear exponential comonad and a recursor.

The Lists

For every $\sim$-arena A, we have a $\sim$-arena A^ℓ such that $A^\ell = \mathbf{1} \oplus (A \otimes A^\ell)$, so we have the first item.

The Quantum Primitives

We also have a $\sim$-arena $\mathbf{qubit} = \downarrow_{\mathbf{qb}:\mathfrak{Q}} \varnothing$, and some quantum $\sim$-strategies $\mathbf{meas}^{\sim\text{-}\mathbf{QA}}$, $\mathbf{new}^{\sim\text{-}\mathbf{QA}}$, and $\mathbf{U}^{\sim\text{-}\mathbf{QA}}$ which are simply the strategies $\mathbf{meas}^{\mathbf{QA}}$, $\mathbf{new}^{\mathbf{QA}}$, and $\mathbf{U}^{\mathbf{QA}}$.

The Functional Sub-SMC

We take $\sim$-$\mathbf{QA}_f$ the full sub-SMC of $\sim$-$\mathbf{QA}_t$ with all the functional $\sim$-arenas (see Definition 8.3.12), and note that $!A$ is always defined in this subcategory.

The Linear Exponential Comonad

We lift the linear exponential comonad from $\sim$-$\mathbf{QArena}^{\perp}_{\mathrm{fun}}$ into a linear exponential comonad for $\sim$-$\mathbf{QA}_f$.

Definition 9.3.7. *For $f \in \sim\text{-}\mathbf{QArena}^{\perp}_{\mathrm{fun}}(A,B)$, if f is receptive, courteous and $\sim$-receptive, then we define its lifting $\widehat{f} \in \sim\text{-}\mathbf{QA}_f(A,B)$ as follows:*

$$\begin{array}{ccc} !C_B & \xrightarrow{\alpha_B} & B^{\perp} \parr B \\ & \searrow^{\widehat{f}} & \downarrow{\scriptstyle f \parr B} \\ & & A^{\perp} \parr B \end{array}$$

This notion of lifting is called co-lifting in [CCW19].

Lemma 9.3.8 (Lifting lemma)**.** *The lifting $\widehat{-}$ is a symmetric monoidal functor from the subcategory of $(\sim\text{-}\mathbf{QArena}^{\perp}_{\mathrm{fun}}, \otimes, \mathbf{1})$ containing all the receptive courteous and $\sim$-receptive maps to $(\sim\text{-}\mathbf{QA}_f, \otimes, \mathbf{1})$. Moreover,*

$$f \simeq f' \implies \widehat{f} \simeq \widehat{f'}$$

and the $\oplus$-covered configurations of $\widehat{f}$ are necessarily of the form $f(x) \parallel x$.

Proposition 9.3.9. *The modality $!$ forms a linear exponential comonad $(!, \epsilon, \delta, \mathrm{w}, \mathrm{c}, \mathrm{m}, \mathrm{m}_{\mathbf{1}})$, up to weak isomorphism, on $\sim$-$\mathbf{QA}_f$.*

Proof. We first check that all the morphisms from the linear exponential comonad of $\sim$-$\mathbf{QArena}^{\perp}_{\mathrm{fun}}$ are receptive courteous and $\sim$-receptive. We then lift all of them to $\sim$-$\mathbf{QA}_f$, and just need to provide a definition for the functor $!$.

We take $\sigma \in \sim\text{-}\mathbf{QA}_f(A,B) \subseteq \sim\text{-}\mathbf{QA}_t(A,B)$. We have $A = \downarrow_{a:\mathbf{1}} A'$, $B = \downarrow_{b:\mathbf{1}} B'$ with A', B' some negative quantum payoff $\sim$-games . Using thunkability $S = \uparrow_{a':\mathbf{1}} \downarrow_{b':\mathbf{1}} S'$, with S' a negative $\sim$-esp and $\sigma(a') = (0,a)$, $\sigma(b') = (1,b)$. We define the map of $\sim$-esp $!\sigma$ as follows:

$$\begin{array}{rccccl} !\sigma : & \uparrow_{a':\mathbf{1}} \downarrow_{b':\mathbf{1}} !S' & \to & (\downarrow_{a:\mathbf{1}} !A')^{\perp} \parr (\downarrow_{b:\mathbf{1}} !B') & \\ & a' & \mapsto & (0,a) & \\ & b' & \mapsto & (1,b) & \\ & (n,s) & \mapsto & (i,(n,e)) & \text{whenever} \quad \sigma(s) = (i,e) \end{array}$$

And we take the valuation:

$$\mathcal{Q}_{!\sigma}(\{a', b'\} \sqcup (x_0 \parallel \ldots \parallel x_n)) = \bigotimes_{i=1}^{n} \mathcal{Q}_\sigma(\{a', b'\} \sqcup x_i)$$

All the diagrams but the naturality follow from the functoriality of the lifting, and their commutation up to $\simeq$ become commutations up to weak isomorphism thanks to the lifting lemma. We then check the naturality diagrams by using Lemmas A.2.3 and A.2.4. The naturality will be satisfied up to strong isomorphism. As an example, we treat the case of the dereliction $\widehat{\mathrm{w}_A} \in \sim\text{-}\mathbf{QA}(!A, A)$. We want to prove the commutation of the following diagram:

$$\begin{array}{ccc} !A & \xrightarrow{\widehat{\mathrm{w}_A}} & A \\ \downarrow{\scriptstyle !\sigma} & & \downarrow{\scriptstyle \sigma} \\ !B & \xrightarrow{\widehat{\mathrm{w}_B}} & B \end{array}$$

The $\oplus$-covered configurations of $\sigma \odot \widehat{\mathrm{w}_A}$ are of the form $x \odot (\mathrm{w}_A(x_A) \parallel x_A)$ with $\sigma\, x = x_A \parallel x_B$. The $\oplus$-covered configurations of $\widehat{\mathrm{w}_B} \odot !\sigma$ are of the form $(\mathrm{w}_B(x_B) \parallel x_B) \odot y$ with $(!\sigma)\, y = y_A \parallel \mathrm{w}_B(x_B)$. Since $\mathrm{w}_B(x_B)$ only uses the copy of B of index 0, then necessarily y only uses the copy of S of index 0. This means that $y = \{0\} \times x$ for some $x \in \mathcal{C}(S)$ $\oplus$-covered with $\sigma\, x = x_A \parallel x_B$. This allows us to build a bijection between $\oplus$-covered configurations of $\sigma \odot \widehat{\mathrm{w}_A}$ and the ones of $\widehat{\mathrm{w}_B} \odot !\sigma$:

$$\{x \odot (\mathrm{w}_A(x_A) \parallel x_A) \mid \sigma\, x = x_A \parallel x_B\} \rightleftarrows \{(\mathrm{w}_B(x_B) \parallel x_B) \odot (\{0\} \times x) \mid \sigma\, x = x_A \parallel x_B\}$$

This bijection is an order-isomorphism, so using Lemma A.2.4, $\sigma \odot \widehat{\mathrm{w}_A} \cong \widehat{\mathrm{w}_B} \odot !\sigma$ when seen as strategies. Since this isomorphism preserves and reflects the symmetry and the quantum valuation, they are strongly isomorphic as quantum $\sim$-strategies. □

The Recursor

For the last item required for the pre-model, we define the recursor $\mathbf{Y}$ through a supremum. As explained in Section 4.4.2 for the case without symmetry, we took an in-between stance with respect to strong isomorphism: the morphisms of $\sim$-$\mathbf{QA}$ are actual $\sim$-strategies, not equivalence classes of such, but all the categorical laws are only satisfied up to strong isomorphism. This ad hoc stance avoids the technical overload of defining a bicategory, while allowing us to still talk about concrete strategies instead of equivalence classes.

This choice is motivated by the fact that in order to build the recursor, we will use a supremum for a certain order, but the substructure order on esps (see Definition 4.4.1) is no longer a partial order when we work up to isomorphisms of esps[3].

[3]The antisymmetry fails in some infinite cases.

While $\sim$-**QA** is only a "non-trivial distributive CFC with a bottom" up to strong isomorphism, it has the same data (objects and morphisms) as a regular "non-trivial distributive CFC with a bottom". The associativity (and other axioms from a "non-trivial distributive CFC with a bottom") are valid only up to strong isomorphism, but that does not prevent us from building concrete $\sim$-strategies for the terms of QΛ!, postponing the strong isomorphism to the lemmas (value substitution, *etc.*)

Since the interactive composition is only associative up to strong isomorphism, we make its bracketing explicit in the remaining of this section.

Definition 9.3.10. *For $\sigma, \tau : A \nrightarrow B$ two $\sim$-strategies, we say that $\sigma \leq \tau$ if there is a map of $\sim$-esp $f : S \to T$ which*

- *is a substructure map of esp,*
- *preserves and reflects the symmetry,*
- *commutes with the $\sim$-strategies: $\sigma = \tau \circ f$ as maps of $\sim$-esps,*
- *preserves the quantum valuation: $\mathcal{Q}_\sigma(x) = \mathcal{Q}_\tau(f\,x)$.*

Proposition 9.3.11. *The poset $(\sim\text{-}\mathbf{QA}(A,B), \leq)$ is a dcpo for any quantum $\sim$-arenas A and B. This dcpo is an enrichment of $\sim$-**QA**, i.e., all the operations of $\sim$-**QA** are monotone and continuous for this dcpo. Moreover $(\{\sigma \odot \mathrm{lu}_A^{-1} \mid \sigma \in \sim\text{-}\mathbf{QA}(\mathbf{1} \otimes A, B)\}, \leq)$ has a minimal element*

$$\bot_{A,B} = ((\mathbf{0}_B \odot \mathbf{0}^{-1}_{\mathbf{0}\otimes A}) \odot (\bot \otimes A)) \odot \mathrm{lu}_A^{-1}$$

The fact that $\bot_{A,B}$ is minimal comes from the fact that $\bot_{A,B}$ is the $\sim$-strategy with only minimal events and no other event in its $\sim$-esp, and from Lemma A.1.5 which ensures that $\bot_{A,B}$ and $\sigma \odot \mathrm{lu}_A^{-1}$ have the same minimal events. Unfortunately, $\bot_{A,B}$ is not thunkable, which means that to use its minimality, we need some back and forth between $\sim$-$\mathbf{QA}_t$ and $\sim$-**QA**.

Definition 9.3.12. *For $\sigma \in \sim\text{-}\mathbf{QA}_f(W \otimes !(A \multimap B), A \multimap B)$ with W of the form $\bigotimes_i !F_i$, we define the operations*

$$\begin{array}{rlll}
\Lambda_! : & \sim\text{-}\mathbf{QA}(W \otimes A, B) & \to & \sim\text{-}\mathbf{QA}_f(W, !(A \multimap B)) \\
& \tau & \mapsto & !\left((A \multimap \tau) \odot \mathrm{fun}_{W,A}\right) \odot \mathrm{dig}_W \\
\Lambda_!^{-1} : & \sim\text{-}\mathbf{QA}_f(\mathbf{1} \otimes W, !(A \multimap B)) & \to & \sim\text{-}\mathbf{QA}((\mathbf{1} \otimes W) \otimes A, B) \\
& \upsilon & \mapsto & \mathrm{eval}_{B,A} \odot ((\epsilon_{A\multimap B} \odot \upsilon) \otimes A)
\end{array}$$

$$\begin{array}{rlll}
\mathcal{F}_\sigma : & \sim\text{-}\mathbf{QA}(W \otimes A, B) & \to & \sim\text{-}\mathbf{QA}(W \otimes A, B) \\
& \tau & \mapsto & \Lambda_!^{-1}\Big(\Big((!\sigma \odot \mathrm{dig}_{W\otimes !(A\multimap B)}) \odot (W \otimes \Lambda_!(\tau))\Big) \\
& & & \odot(\mathrm{contr}_{W,\mathbf{1},\mathbf{1}} \odot \mathrm{br}_{\mathbf{1},W})\Big) \odot \mathrm{lu}^{-1}_{W\otimes A}
\end{array}$$

And the recursor $\mathbf{Y}(\sigma) := \Lambda_!\left(\lim_n \mathcal{F}_\sigma^n\left(\bot_{W\otimes A,B}\right)\right)$

This definition relies on Proposition 9.3.11, which ensures that whenever $n \leq m$ we have $\mathcal{F}_\sigma^n(\bot_{W\otimes A,B}) \leq \mathcal{F}_\sigma^m(\bot_{W\otimes A,B})$.

Lemma 9.3.13. *The recursor* $\mathbf{Y}$ *satisfies up to weak isomorphism the two axioms required in Section 7.2, i.e.,*

$$\mathbf{Y}(\sigma) \odot \sigma' \simeq \mathbf{Y}(\sigma \odot (\sigma' \otimes !(A \multimap B)))$$

$$\mathbf{Y}(\sigma) \simeq \Big(\Big((!\sigma \odot \mathrm{dig}_{W\otimes !(A\multimap B)}) \odot (W \otimes \upsilon)\Big) \odot \mathrm{contr}_{W,\mathbf{1},\mathbf{1}}\Big) \odot \mathrm{ru}_W^{-1}$$

Proof. The second axiom is the definition of $\mathbf{Y}(\sigma)$, with $\Lambda_!^{-1}$ and $\Lambda_!$ cancelling each other up to weak isomorphism. To prove the first axiom, we proceed by induction on n and prove that it is satisfied by all the $\Lambda_!(\mathcal{F}_\sigma^n(\bot_{W\otimes A,B}))$, relying on the properties of the linear exponential comonad up to weak isomorphism. We then use continuity of all the operations (Proposition 9.3.11) to show that $\mathbf{Y}(\sigma)$ satisfies this property. □

It follows that $\sim$-**QA**, up to weak isomorphism, is a pre-model of LQ$\Lambda_!$, and in particular satisfies the invariance lemma for terms, up to weak isomorphism:

$$t \to s \implies [\![t]\!] \simeq [\![s]\!]$$

9.3.4 Pre-Model for AQ$\Lambda_!$

Similarly, $(\sim\text{-}\mathbf{QA}^a, \sim\text{-}\mathbf{QA}_t^a, \mathbf{id}, \otimes, \mathbf{1}, \multimap, \oplus, \mathbf{0})$ is a non-trivial affine CFC with a bottom, has lists, quantum primitives, and a functional subcategory (with only functional affine $\sim$-arenas), has a linear exponential comonad up to weak isomorphism, and has a recursor up to weak isomorphism, so is a pre-model for AQ$\Lambda_!$ in the sense of Section 7.2, up to weak isomorphism.

9.4 The Exhaustive Equivalence

In this section, we extend the notion of exhaustive equivalence to $\sim$-strategies, and the collapse of $\sim$-strategies into relations on $\sim$-arenas. This exhaustive equivalence is central to the proof of full abstraction, but proving that this exhaustive equivalence is a congruence is very technical. We start by laying down multiple definitions and lemmas on $\sim$-strategies useful for this proof.

9.4.1 Motivation

As seen in Section 6.2.2, the core notions behind the exhaustive equivalence of quantum strategies are the notion of witnesses of a quantum strategy $\sigma : A \nrightarrow B$:

$$\mathrm{wit}_\sigma(x_A, x_B) := \left\{ x \in \mathcal{C}(S) \;\middle|\; \begin{array}{c} x \oplus\text{-covered} \\ \sigma\, x \in x_A \parallel x_B \end{array} \right\}$$

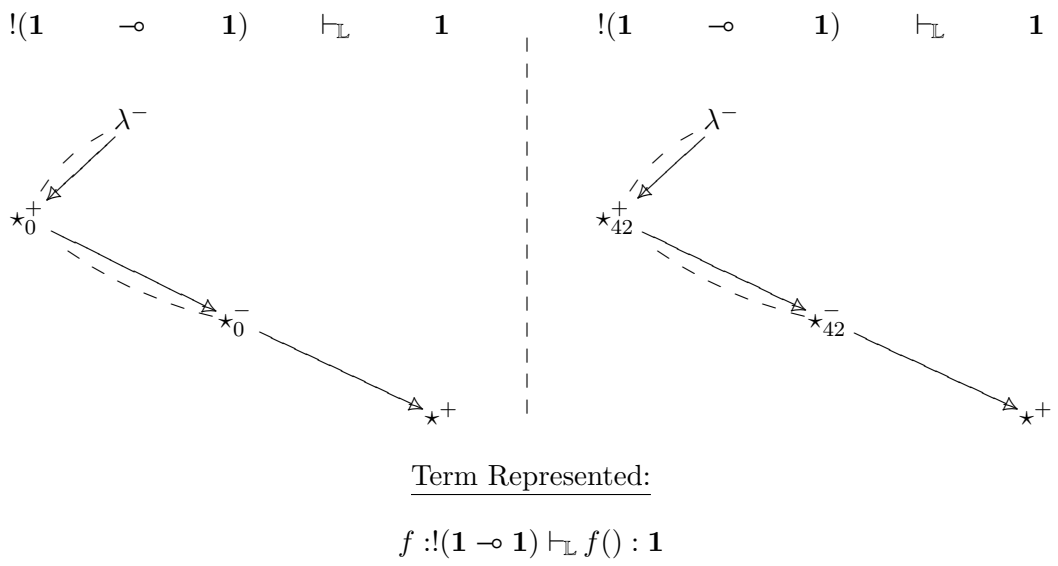


Figure 9.11: Two weakly isomorphic ∼-strategies.

And the notion of collapsed quantum valuation:

$$\mathcal{Q}_\sigma(x_A, x_B) := \sum_{x \in \mathrm{wit}_\sigma(x_A, x_B)} \mathcal{Q}_\sigma(x)$$

Two strategies are exhaustively equivalent if and only they have the same collapsed quantum valuations. Unfortunately, this notion of exhaustive equivalence is not directly compatible with symmetry.

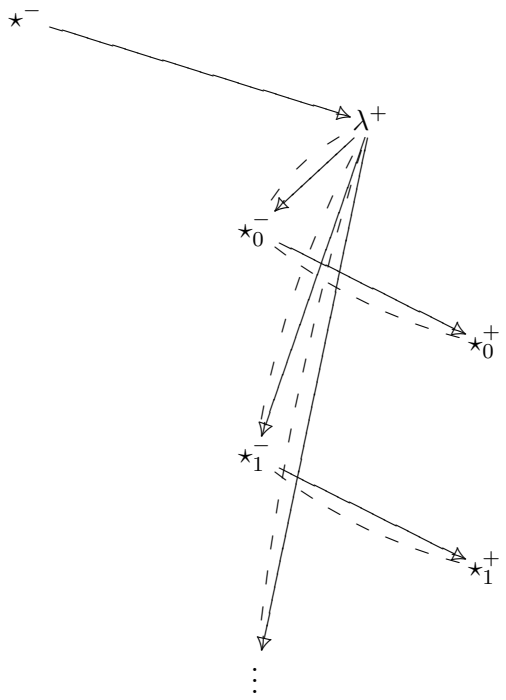
For example, in Fig. 9.11, we provide two ∼-strategies (with trivial symmetry and trivial quantum valuation), each playing exactly one copy of the replicated function, one playing the copy of index 0 and one of index 42; they are weakly isomorphic but not exhaustively equivalent as strategies as

	Left hand side	Right hand side
$\mathcal{Q}(\{\lambda^-, \star_0^+, \star_0^-\}, \{\star^+\})$	$\mathbf{id_1}$	0
$\mathcal{Q}(\{\lambda^-, \star_{42}^+, \star_{42}^-\}, \{\star^+\})$	0	$\mathbf{id_1}$

The main issue is that $\mathcal{Q}(-,-)$ only sums over witnesses, and not "witnesses up to symmetry". In fact, instead of defining $\mathcal{Q}(-,-)$ on configurations of the game, we would like to define it on equivalence classes of configurations:

$$\mathcal{Q}_\sigma : (\mathbf{x_A}, \mathbf{x_B}) \in \mathcal{C}_\simeq(A) \times \mathcal{C}_\simeq(B) \mapsto \sum_{\substack{x \text{ witness of} \\ \mathbf{x_A} \| \mathbf{x_B}}} \mathcal{Q}_\sigma(x)$$

$$\varnothing \qquad \vdash_{\mathbb{L}} \qquad !(\mathbf{1} \qquad \multimap \qquad \mathbf{1})$$



Term Represented:

$$\vdash_{\mathbb{L}} \lambda().() :!(\mathbf{1} \multimap \mathbf{1})$$

Figure 9.12: The $\sim$-strategy for the identity function.

Note that this is not a proper definition, as (1) we did not define what it means to be a witness of an equivalence class and (2) the sum does not type-check as multiple $\mathcal{Q}_\sigma(x)$ might have different domain and codomain Hilbert spaces.

One might think that we could simply define the witnesses of an equivalence class $\mathbf{x_A} \parallel \mathbf{x_B}$ as the set of all $x \in \mathcal{C}(S)$ such that $\sigma\, x \in \mathbf{x_A} \parallel \mathbf{x_B}$. This is however not a sensible definition, as it leads to infinite sums even in simple examples like $\vdash_{\mathbb{L}} \lambda().() :!(\mathbf{1} \multimap \mathbf{1})$ described in Fig. 9.12 (with for symmetry every order-isomorphism, and a trivial quantum valuation). With this definition of witnesses, we would indeed have:

$$\mathcal{Q}(\{\star^-\}, \{\lambda^+, \star_{\mathbf{0}}^-, \star_{\mathbf{0}}^+\}) = \infty \cdot \mathbf{id_1}$$

As explained in more detail in [Cla20], a better solution is to only consider the $x \in \mathcal{C}(S)$

such that there exists $\theta^+ : \sigma\, x \simeq^+ \underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}}$. With this new definition of witnesses, where we only consider witnesses positively symmetric to canonical configurations, we would obtain

$$\mathcal{Q}(\{\star^-\}, \{\lambda^+, \star_{\mathbf{0}}^-, \star_{\mathbf{0}}^+\}) = \mathcal{Q}(\{\star^-, \lambda^+, \star_0^-, \star_0^+\}) = \mathbf{id_1}$$

As shown in Lemma 9.4.2, we do not miss any information by counting only those, as every witness according to the old definition is symmetric in the $\sim$-strategy to a witness according to the new definition.

To answer the second issue of "the sum does not type-check as multiple $\mathcal{Q}_\sigma(x)$ might have different domain and codomain Hilbert spaces", we consider configurations $x \in \mathcal{C}(S)$ together with a positive symmetry $\theta_A \parallel \theta_B : \sigma\, x \simeq^+_{A^\perp \parr B} \underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}}$, which will allow us to use $\mathcal{H}_B(\theta_B^{-1}) \circ \mathcal{Q}(x) \circ \mathcal{H}_A(\theta_A^{-1})$ in the sum. Now that we have explained the basics, we give formal definitions.

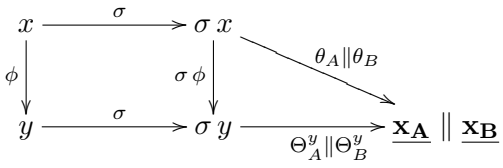
9.4.2 Configurations with Symmetry

Definition 9.4.1. *For $\sigma \in \sim$-**Strat**(A, B), a $\overset{+}{\sim}$-configuration $\lceil x \rceil = (\Theta_A^x, x, \Theta_B^x) \in \overset{+}{\sim}$-$\mathcal{C}(S)$ is a configuration $x \in \mathcal{C}(S)$ together with two symmetries Θ_A^x and Θ_B^x:*

$$\Theta_A^x : x_A \simeq_A^- \underline{\mathbf{x_A}} \qquad \Theta_B^x : x_B \simeq_B^+ \underline{\mathbf{x_B}} \qquad \text{where } \sigma\, x = x_A \parallel x_B$$

We note that we only coerce through positive symmetries in $A^\perp \parr B$, *i.e.*, negative symmetries in A and positive symmetries in B. This is related to counting problems as hinted before, and backed up by the following lemma, which states that we can always put ourselves in a situation where coercion through positive symmetry is enough:

Lemma 9.4.2. *For $\sigma \in \sim$-**Strat**(A, B), if $x \in \mathcal{C}(S)$ and $\theta_A \parallel \theta_B : \sigma\, x \simeq_{A^\perp \parr B} \underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}}$, then there exist $\lceil y \rceil \in \overset{+}{\sim}$-$\mathcal{C}(S)$ and $\phi : x \simeq_S y$ such that the following diagram commutes:*



This lemma is a consequence of lemma B.4 of [CCW19]. We can extend the notion of matching and compatibility from Definition 4.3.1 to $\overset{+}{\sim}$-configurations. While we could also extend the notion of minimally matching compatible to $\overset{+}{\sim}$-configurations, we will not use this notion.

Definition 9.4.3. *For $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$, and two $\overset{+}{\sim}$-configurations $\lceil x \rceil \in \overset{+}{\sim}$-$\mathcal{C}(S)$ and $\lceil y \rceil \in \overset{+}{\sim}$-$\mathcal{C}(T)$, we write $\sigma\, x = x_A \parallel x_B$ and $\tau\, y = y_B \parallel y_C$. Those two $\sim$-configurations are said:*

Matching *if* $\mathbf{x_B} = \mathbf{y_B}$

Matching Compatible *if moreover the induced pre-order over* $\underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}} \parallel \underline{\mathbf{y_C}}$ *is acyclic, i.e., an order. This pre-order is obtained as follows: we note that* $(x \parallel \underline{\mathbf{y_C}}, \leq_{F \parallel C})$ *and* $(\underline{\mathbf{x_A}} \parallel y, \leq_{A \parallel G})$ *are two posets, take their image by* $(\Theta^x_A \parallel \Theta^x_B) \circ (\sigma \parallel C)$ *and* $(\Theta^y_B \parallel \Theta^y_C) \circ (A \parallel \tau)$*, and then the transitive closure of the union of both. The transitive closure of the union of two posets might not be a poset.*

Lemma 9.4.4 (Deadlock-Freeness)**.** *If* $\sigma \in \sim\text{-}\mathbf{Strat}(A, B)$ *and* $\tau \in \sim\text{-}\mathbf{Strat}(B, C)$ *are two* $\sim$*-strategies that satisfy the visibility condition (see Definition 6.2.2), and* A, B, C *are* $\sim$*-arenas, then the interactive composition is deadlock-free, i.e., matching pairs of configurations are compatible. Moreover, it is* $\overset{+}{\sim}$*-deadlock-free, i.e., matching pairs of* $\overset{+}{\sim}$*-configurations are compatible.*

Proof. The first part follows directly from Theorem 6.2.5 which proves deadlock-freeness on strategies without symmetry. For the second part, we can adapt the proof of Theorem 6.2.5 to matching $\overset{+}{\sim}$-pairs instead of matching pairs. □

As in the case without replication, this deadlock-freeness lemma is necessary to prove that the exhaustive equivalence is a congruence.

9.4.3 Witnesses up to Symmetry

In Definition 6.2.6, we defined $\mathrm{wit}_\sigma(x_A, x_B)$ as the set of $\oplus$-covered configurations that project to $x_A \parallel x_B$, and this set plays a major role in the definition of both the exhaustive equivalence and the relational collapse. We extend this definition with symmetry as hinted earlier, and prove two groups of lemmas: the first on how to apply a symmetry to a $\overset{+}{\sim}$-configuration (which is central to the proof that the collapse of a quantum $\sim$-strategy is a quantum relation on $\sim$-arenas), and the second on witnesses of the interaction of two $\sim$-strategies (which is central to the proof that the exhaustive equivalence is a congruence). We refer to [Cla20] for the proofs of those lemmas.

Definition 9.4.5. *For* $\sigma \in \sim\text{-}\mathbf{Strat}(A, B)$ *a* $\sim$*-strategy and* $(\mathbf{x_A}, \mathbf{x_B}) \in \mathcal{C}_\simeq(A) \times \mathcal{C}_\simeq(B)$*, we define*

$$\overset{+}{\sim}\text{-}\mathrm{wit}_\sigma(\mathbf{x_A}, \mathbf{x_B}) := \left\{ \lceil x \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S) \;\middle|\; \begin{array}{c} x \text{ } \oplus\text{-covered} \\ \sigma\, x \in \mathbf{x_A} \parallel \mathbf{x_B} \end{array} \right\}$$

We start by a property that states that we can "apply" a negative symmetry of the $\sim$-game to a $\overset{+}{\sim}$-configuration of a $\sim$-strategy and obtain another $\overset{+}{\sim}$-configuration in a bijective way.

Lemma 9.4.6. *For $\sigma \in \sim\text{-}\mathbf{Strat}(A,B)$ a $\sim$-strategy and $(\mathbf{x_A},\mathbf{x_B}) \in \mathcal{C}_{\simeq}(A)\times\mathcal{C}_{\simeq}(B)$, there is a group action $(_ \curvearrowright _) : \mathcal{A}^{-}_{A^\perp \parr B}(\underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}}) \times \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B}) \to \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})$. such that for all $\lceil y \rceil = \varphi^- \curvearrowright \lceil z \rceil$ there is $\phi : z \simeq_S y$ such that the following diagram commutes:*

$$\begin{array}{ccc} \sigma z & \xrightarrow[\simeq^+_{A^\perp\parr B}]{\Theta^z_A \parr \Theta^z_B} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \\ {\scriptstyle \sigma\,\phi}\downarrow{\scriptstyle \simeq_{A^\perp\parr B}} & & {\scriptstyle \simeq^-_{A^\perp\parr B}}\downarrow{\scriptstyle \varphi^-} \\ \sigma y & \xrightarrow[\Theta^y_A \parr \Theta^y_B]{\simeq^+_{A^\perp\parr B}} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \end{array}$$

This lemma is exactly Proposition 17 of [Cla20]. Its dual with positive symmetry is also true, though the proof is much simpler:

Lemma 9.4.7. *For $\sigma \in \sim\text{-}\mathbf{Strat}(A,B)$ a $\sim$-strategy and $(\mathbf{x_A},\mathbf{x_B}) \in \mathcal{C}_{\simeq}(A)\times\mathcal{C}_{\simeq}(B)$, there is a group action $(_ \curvearrowright _) : \mathcal{A}^{+}_{A^\perp \parr B}(\underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}}) \times \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B}) \to \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})$ such that for all $\lceil y \rceil = \varphi^+ \curvearrowright \lceil z \rceil$ there is $\phi : z \simeq_S y$ such that the following diagram commutes:*

$$\begin{array}{ccc} \sigma z & \xrightarrow[\simeq^+_{A^\perp\parr B}]{\Theta^z_A \parr \Theta^z_B} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \\ {\scriptstyle \sigma\,\phi}\downarrow{\scriptstyle \simeq_{A^\perp\parr B}} & & {\scriptstyle \simeq^+_{A^\perp\parr B}}\downarrow{\scriptstyle \varphi^+} \\ \sigma y & \xrightarrow[\Theta^y_A \parr \Theta^y_B]{\simeq^+_{A^\perp\parr B}} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \end{array}$$

Proof. We consider $\lceil z \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})$ and $\varphi^+ \in \mathcal{A}^{+}_{A^\perp \parr B}(\underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}})$. We write $\varphi^+ = \varphi^+_A \parr \varphi^+_B$. We note that can simply post-compose by the symmetry φ^+ and obtain:

$$\begin{array}{ccc} \sigma z & \xrightarrow[\simeq^+_{A^\perp\parr B}]{\Theta^z_A \parr \Theta^z_B} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \\ \| & & {\scriptstyle \simeq^+_{A^\perp\parr B}}\downarrow{\scriptstyle \varphi^+} \\ \sigma z & \xrightarrow[((\phi^+_A)^{-1}\circ\Theta^z_A)\parr((\phi^+_B)^{-1}\circ\Theta^z_B)]{\simeq^+_{A^\perp\parr B}} & \underline{\mathbf{x_A}} \parr \underline{\mathbf{x_B}} \end{array}$$

So we take $\varphi^+ \curvearrowright \lceil z \rceil := (((\phi^+_A)^{-1}\circ\Theta^z_A), z, ((\phi^+_B)^{-1}\circ\Theta^z_B))$. This is a group action. □

We now want to prove properties about the witnesses of the interaction of two $\sim$-strategies. The end goal of those properties is Proposition 9.4.17 which shows that the exhaustive equivalence is a congruence, and that the relational collapse is a functor. We start by defining what is a witness of the interaction.

Definition 9.4.8. *For $\sigma \in \sim\text{-}\mathbf{Strat}(A,B)$,$\tau \in \sim\text{-}\mathbf{Strat}(B,C)$ two $\sim$-strategies, we define the $\overset{+}{\sim}$-configurations of the interaction $\lceil y \circledast x \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(T \circledast S)$ as the configuration $y \circledast x \in \mathcal{C}(T \circledast S)$ together with two symmetries $\Theta^{y\circledast x}_A$ and $\Theta^{y\circledast x}_C$:*

$$\Theta^{y\circledast x}_A : x_A \simeq^-_A \underline{\mathbf{x_A}} \qquad \Theta^{y\circledast x}_C : y_C \simeq^+_C \underline{\mathbf{y_C}} \qquad \text{where } \sigma\, x = x_A \parr x_B \text{ and } \tau\, y = x_B \parr y_C$$

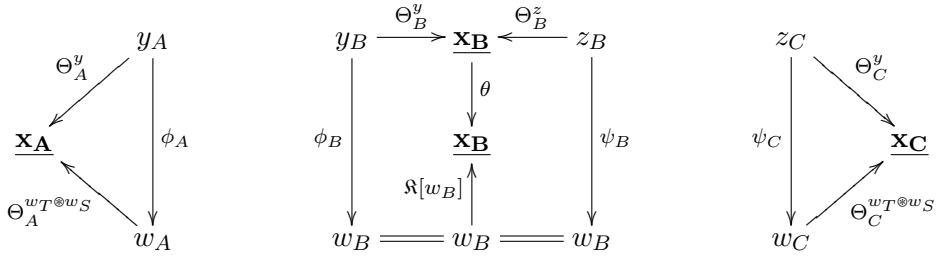
Definition 9.4.9. *For $\sigma \in \sim\text{-}\mathbf{Strat}(A,B)$,$\tau \in \sim\text{-}\mathbf{Strat}(B,C)$ two $\sim$-strategies and for $(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C}) \in \mathcal{C}_{\simeq}(A)\times\mathcal{C}_{\simeq}(B)\times\mathcal{C}_{\simeq}(C)$, we define*

$$\overset{+}{\sim}\text{-wit}_{\tau\circledast\sigma}(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C}) := \left\{ \lceil y \circledast x \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(T\circledast S) \;\middle|\; \begin{array}{c} y,x \text{ } \oplus\text{-covered} \\ (\tau\circledast\sigma)\,(y\circledast x) \in \mathbf{x_A} \parallel \mathbf{x_B} \parallel \mathbf{x_C} \end{array} \right\}$$

Proposition 9.4.10. *For $\sigma \in \sim\text{-}\mathbf{Strat}(A,B)$,$\tau \in \sim\text{-}\mathbf{Strat}(B,C)$ two $\sim$-strategies and $(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C}) \in \mathcal{C}_{\simeq}(A)\times\mathcal{C}_{\simeq}(B)\times\mathcal{C}_{\simeq}(C)$, there is a bijection*

$$\Upsilon : \underset{\text{matching compatible}}{\overset{+}{\sim}\text{-wit}_{\sigma}(\mathbf{x_A},\mathbf{x_B}) \times \overset{+}{\sim}\text{-wit}_{\tau}(\mathbf{x_B},\mathbf{x_C})} \quad\to\quad \overset{+}{\sim}\text{-wit}_{\tau\circledast\sigma}(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C}) \times \mathcal{A}_B(\underline{\mathbf{x_B}})$$

such that if we write $\Upsilon(\lceil y\rceil,\lceil z\rceil) = (\lceil w_T \circledast w_S\rceil,\theta)$, and $y_A,y_B,z_B,z_C,w_A,w_B,w_C$ for their respective projections, then there exists $\phi_S : y \simeq_S w_S$ projecting to ϕ_A and ϕ_B and $\psi_T : z \simeq_T w_T$ projecting to ψ_B and ψ_C such that the following diagrams commute:



Where we $\Re[w_B] : w_B \simeq_B \underline{\mathbf{w_B}} = \underline{\mathbf{x_B}}$ is an arbitrarily chosen symmetry of B (fixed once and for all). A different choice for $\Re[w_B]$ only affects θ, and has no later consequence. This proposition is exactly corollary 23 from [Cla20].

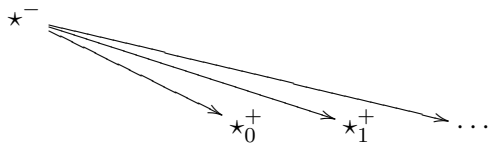
9.4.4 Quantum Valuations and Symmetry

Now that we have defined witnesses up to symmetry, and proved some powerful lemmas on them, we can define the collapsed quantum valuation of a quantum $\sim$-strategy, and leverage the previous lemmas. As hinted before, we will sum over the configurations x of the strategy σ such that $\sigma\, x$ is positively symmetric to a canonical configuration $\underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}}$, and we will coerce the quantum annotation according to how they are positively symmetric. However, what do we do when there are multiple ways of being positively symmetric? The correct answer is average over all the possibilities. How many possibility are there? The answer is the following.

Lemma 9.4.11. *For $\sigma \in \sim\text{-}\mathbf{QA}(A,B)$ and $\lceil x\rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S)$ with $\sigma\, x = x_A \parallel x_B$.*

$$|\{(\theta_B, x, \theta_A) \in \overset{+}{\sim}\text{-}\mathcal{C}(S)\}| = |\mathcal{A}^{+}_{\mathcal{A}^{\perp} \parr B}(\underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}})|$$

$$\mathbf{1} \qquad \twoheadrightarrow \qquad ?\mathbf{1}$$



Symmetry for the ∼-Game:

$\theta \in \simeq^{+}_{\mathbf{1}^\perp \parr ?\mathbf{1}}$ whenever θ is a polarity-preserving isomorphism
$\theta \in \simeq^{-}_{\mathbf{1}^\perp \parr ?\mathbf{1}}$ whenever θ is the identity
$\theta \in \simeq_{\mathbf{1}^\perp \parr ?\mathbf{1}}$ whenever θ is a polarity-preserving isomorphism

Symmetry for the ∼-Strategy:

$\theta \in \simeq$ whenever θ is the identity

Quantum Valuation for the ∼-Strategy:

$$\mathcal{Q}(x) = \mathbf{id_1}$$
$$\mathcal{Q}(\{\star\}, \{\star_\mathbf{0}\}) = \infty \cdot \mathbf{id_1}$$

Figure 9.13: Example of infinitary ∼-strategy.

Proof. We have a bijection between the two, given by

$$(\theta_B, x, \theta_A) \mapsto (\theta_B || \theta_A) \circ (\Theta^x_B \parallel \Theta^x_A)^{-1}$$ □

This allows us to take the following definitions.

Definition 9.4.12. *For $\sigma \in \sim\text{-}\mathbf{QA}(A, B)$ and $\lceil x \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S)$, we define*

$$\mathcal{Q}_\sigma(\lceil x \rceil) := \mathcal{H}_B((\Theta^x_B)^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\Theta^x_A)$$

Then for $\mathbf{x_A} \in \mathcal{C}_\simeq(A)$ and $\mathbf{x_B} \in \mathcal{C}_\simeq(B)$, we define

$$\begin{aligned} \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) &:= \sum_{\lceil x \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{Q}_\sigma(\lceil x \rceil)}{|\mathcal{A}^+_{A^\perp \parr B}(\mathbf{x_A} \| \mathbf{x_B})|} \qquad \in \overline{\mathbf{CPM}}(\mathcal{H}_A(\underline{\mathbf{x_A}}), \mathcal{H}_B(\underline{\mathbf{x_B}})) \\ &= \sum_{\lceil x \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{H}_B((\Theta^x_B)^{-1})}{|\mathcal{A}^+_B(\underline{\mathbf{x_B}})|} \circ \mathcal{Q}_\sigma(x) \circ \frac{\mathcal{H}_A(\Theta^x_A)}{|\mathcal{A}^-_A(\underline{\mathbf{x_A}})|} \end{aligned}$$

While the sum is potentially infinite, meaning a priori $\mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$ can be in the infinitary fragment of $\overline{\mathbf{CPM}}$, as proven in Corollary 10.3.4, this is never the case for ∼-strategies between arenas that come from types of QΛ!. Outside of QΛ! types, we provide in Fig. 9.13 an example of a ∼-strategy from $\mathbf{1}$ to $?\mathbf{1}$ which reaches the infinitary fragment: the strategy here "uses $?\mathbf{1}$" infinitely often, in parallel, with probability one every time. There is however one simple situation where $\mathcal{Q}(-,-)$ is finitary:

Lemma 9.4.13. *For $\sigma \in \sim\text{-}\mathbf{QA}(\mathbf{1},\mathbf{1})$, we have $\mathcal{Q}_\sigma(\{\star\},\{\star\}) \sqsubseteq \mathbf{id}_\mathbf{1}^{\mathbf{CPM}}$.*

Proof. We write m for the minimal event of S. We consider the witnesses $x_1,\dots,x_n \in \overset{+}{\sim}\text{-wit}_\sigma(\{\star\},\{\star\})$. We necessarily have $\{m\} \subset^+ x_1,\dots,x_n$. Using the drop condition we obtain

$$\mathbf{id_1} - \mathcal{Q}_\sigma(x_1) - \cdots - \mathcal{Q}_\sigma(x_n) \sqsupseteq 0$$

So $\sum_{i=1}^n \mathcal{Q}_\sigma(x_i) \sqsubseteq \mathbf{id_1}$. If $\overset{+}{\sim}\text{-wit}_\sigma(\{\star\},\{\star\})$ is finite, then we immediately have

$$\mathcal{Q}_\sigma(\{\star\},\{\star\}) \sqsubseteq \mathbf{id}_\mathbf{1}^{\mathbf{CPM}}$$

Otherwise, we need to take a supremum and obtain $\mathcal{Q}_\sigma(\{\star\},\{\star\}) \sqsubseteq \mathbf{id}_\mathbf{1}^{\mathbf{CPM}}$. □

In order to translate the diagrams from Lemmas 9.4.6 and 9.4.7 into properties on the quantum valuation, we use the following lemma:

Lemma 9.4.14. *We assume $\lceil x\rceil, \lceil y\rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S)$, and write $\sigma\, x = x_A \parallel x_B$ and $\sigma\, y = y_A \parallel y_B$. If $\phi : x \simeq_S y$ and $\varphi_A \parallel \varphi_B : \underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}} \simeq_{A^\perp \parr B} \underline{\mathbf{y_A}} \parallel \underline{\mathbf{y_B}}$ are such that the following diagram commutes:*

$$\begin{array}{ccc} \sigma x & \xrightarrow[\simeq^+_{A^\perp \parr B}]{\Theta^x_A \parr \Theta^x_B} & \underline{\mathbf{x_A}} \parallel \underline{\mathbf{x_B}} \\ {\scriptstyle \sigma\,\phi}\downarrow{\scriptstyle \simeq_{A^\perp \parr B}} & & {\scriptstyle \simeq_{A^\perp \parr B}}\downarrow{\scriptstyle \varphi_A \parallel \varphi_B} \\ \sigma y & \xrightarrow[\Theta^y_A \parallel \Theta^y_B]{\simeq^+_{A^\perp \parr B}} & \underline{\mathbf{y_A}} \parallel \underline{\mathbf{y_B}} \end{array}$$

Then $\mathcal{Q}_\sigma(\lceil y\rceil) = \mathcal{H}_B(\varphi_B^{-1}) \circ \mathcal{Q}_\sigma(\lceil x\rceil) \circ \mathcal{H}_A(\varphi_A)$.

Proof. By definition of quantum $\sim$-strategies, since $\phi : x \simeq_S y$, if we write ϕ_A and ϕ_B for its projections then

$$\mathcal{Q}_\sigma(y) = \mathcal{H}_B(\phi_B^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\phi_A)$$

It follows that:

$$\mathcal{Q}_\sigma(\lceil y\rceil) = \mathcal{H}_B((\Theta^y_B)^{-1}) \circ \mathcal{H}_B(\phi_B^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\phi_A) \circ \mathcal{H}_A(\Theta^y_A)$$

Using the commutation of the diagram, we obtain the expected result. □

This allow us to proves that $\mathcal{Q}_\sigma(-,-)$ is uniform for the symmetry of the game:

Proposition 9.4.15. *For $\sigma \in \sim\text{-}\mathbf{QA}(A,B)$, $\mathbf{x_A} \in \mathcal{C}_\simeq(A)$, and $\mathbf{x_B} \in \mathcal{C}_\simeq(B)$:*

$$\mathcal{H}_B\left(\mathcal{A}_B(\mathbf{x_B})\right) \circ \mathcal{Q}_\sigma(\mathbf{x_A},\mathbf{x_B}) \circ \mathcal{H}_A\left(\mathcal{A}_A(\mathbf{x_A})\right) = \mathcal{Q}_\sigma(\mathbf{x_A},\mathbf{x_B})$$

Proof. We will prove the following property which is equivalent[a]:

$$\forall \theta_A \in \mathcal{A}_A(\mathbf{x_A}), \theta_B \in \mathcal{A}_B(\mathbf{x_B}), \mathcal{H}_B(\theta_B^{-1}) \circ \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) \circ \mathcal{H}_A(\theta_A) = \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$$

We use the definition of $\mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$ and obtain that the left hand side is equal to:

$$\sum_{\lceil z \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \mathcal{H}_B(\theta_B^{-1}) \circ \frac{\mathcal{Q}_\sigma(\lceil z \rceil)}{|\mathcal{A}^+_{A^\perp \parr B}(\mathbf{x_A} \parallel \mathbf{x_B})|} \circ \mathcal{H}_A(\theta_A)$$

If $\theta_A \parallel \theta_B \in \mathcal{A}^-_{A^\perp \parr B}(\mathbf{A}, \mathbf{B})$, then using Lemma 9.4.6 and Lemma 9.4.14, the left hand side is equal to:

$$\sum_{\lceil z \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{Q}_\sigma((\theta_B \parallel \theta_A) \curvearrowright \lceil z \rceil)}{|\mathcal{A}^+_{A^\perp \parr B}(\mathbf{x_A} \parallel \mathbf{x_B})|}$$

Since $\curvearrowright$ is a group action, $(\theta_B \parallel \theta_A) \curvearrowright _$ forms a bijection, the left hand side is equal to:

$$\sum_{\lceil y \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{Q}_\sigma(\lceil y \rceil)}{|\mathcal{A}^+_{A^\perp \parr B}(\mathbf{x_A} \parallel \mathbf{x_B})|}$$

Which is by definition the right hand side. If $\theta_A \parallel \theta_B \in \mathcal{A}^+_{A^\perp \parr B}(\mathbf{A}, \mathbf{B})$, then we proceed similarly using Lemma 9.4.6 and Lemma 9.4.14. In the general case, we decompose $\theta_A \parallel \theta_B$ as a positive and negative auto-symmetry, using the fact that $A^\perp \parr B$ is representable, and we apply successively the reasoning for the negative and positive case. □

[a] For the direct implication, we use that $\mathcal{A}_B(\mathbf{x_B})$ and $\mathcal{A}_A(\mathbf{x_A})$ are groups. For the reverse implication, we simply use the linearity of the sum.

As an immediate corollary, we have the following:

Corollary 9.4.16. *For $\sigma \in \sim$-$\mathbf{QA}(A, B)$, we have $\mathcal{Q}_\sigma(-,-) \in \sim$-$\mathbf{QARel}(A, B)$.*

Note that this does not mean that the collapse $\mathcal{Q}(-,-)$ is a functor from $\sim$-**QA** to $\sim$-**QARel**, as we still need to prove its functoriality. We can now collect all the different lemmas proven up until now, and combine them to obtain the following proposition, which will be central in the proof of congruence of the exhaustive equivalence, and the proof of functoriality of the collapse.

Proposition 9.4.17. *For $\sigma \in \sim$-$\mathbf{QA}(A, B)$ and $\tau \in \sim$-$\mathbf{QA}(B, C)$, which we recall are visible $\sim$-strategies, and for $\mathbf{x_A} \in \mathcal{C}_\simeq(A), \mathbf{x_C} \in \mathcal{C}_\simeq(C)$ we have*

$$\mathcal{Q}_{\tau \odot \sigma}(\mathbf{x_A}, \mathbf{x_C}) = \sum_{\mathbf{x_B} \in \mathcal{C}_\simeq(B)} \mathcal{Q}_\tau(\mathbf{x_B}, \mathbf{x_C}) \circ \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$$

Proof.

$$
\begin{aligned}
\mathcal{Q}_{\tau\odot\sigma}(\mathbf{x_A},\mathbf{x_C}) &= \sum_{\lceil w_T\odot w_S\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})} \frac{\mathcal{H}_C((\Theta_C^{w_T\odot w_S})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau\odot\sigma}(w_T\odot w_S)\circ\frac{\mathcal{H}_A(\Theta_A^{w_T\odot w_S})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (1)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)}\ \sum_{\lceil w_T\circledast w_S\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C})} \frac{\mathcal{H}_C((\Theta_C^{w_T\circledast w_S})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau\circledast\sigma}(w_T\circledast w_S)\circ\frac{\mathcal{H}_A(\Theta_A^{w_T\circledast w_S})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (2)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)}\ \sum_{\lceil w_T\circledast w_S\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B},\mathbf{x_C})} \frac{\mathcal{H}_C((\Theta_C^{w_T\circledast w_S})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau}(w_T)\circ\mathcal{Q}_{\sigma}(w_S)\circ\frac{\mathcal{H}_A(\Theta_A^{w_T\circledast w_S})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (3)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)}\ \sum_{\substack{\lceil z\rceil\in\overset{+}{\sim}\text{-wit}_\tau(\mathbf{x_B},\mathbf{x_C})\\ \lceil y\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})\\ \Upsilon(\lceil y\rceil,\lceil z\rceil)=(\lceil w_T\circledast w_S\rceil,\theta)}} \frac{\mathcal{H}_C((\Theta_C^{w_T\circledast w_S})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau}(w_T)\circ\frac{\mathbf{id}}{|\mathcal{A}_B(\underline{\mathbf{x_B}})|}\circ\mathcal{Q}_{\sigma}(w_S)\circ\frac{\mathcal{H}_A(\Theta_A^{w_T\circledast w_S})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (4)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)}\ \sum_{\substack{\lceil z\rceil\in\overset{+}{\sim}\text{-wit}_\tau(\mathbf{x_B},\mathbf{x_C})\\ \lceil y\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})}} \frac{\mathcal{H}_C((\Theta_C^{z})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau}(z)\circ\frac{\mathcal{H}_B(\Theta_B^{z}\circ(\Theta_B^{y})^{-1})}{|\mathcal{A}_B(\underline{\mathbf{x_B}})|}\circ\mathcal{Q}_{\sigma}(y)\circ\frac{\mathcal{H}_A(\Theta_A^{y})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (5)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)} \sum_{\lceil z\rceil\in\overset{+}{\sim}\text{-wit}_\tau(\mathbf{x_B},\mathbf{x_C})} \frac{\mathcal{H}_C((\Theta_C^{z})^{-1})}{|\mathcal{A}_C^+(\underline{\mathbf{x_C}})|}\circ\mathcal{Q}_{\tau}(z)\circ\frac{\mathcal{H}_B(\Theta_B^{z})}{|\mathcal{A}_B^-(\underline{\mathbf{x_B}})|} \\
&\quad \circ \sum_{\lceil y\rceil\in\overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A},\mathbf{x_B})} \frac{\mathcal{H}_B((\Theta_B^{y})^{-1})}{|\mathcal{A}_B^+(\underline{\mathbf{x_B}})|}\circ\mathcal{Q}_{\sigma}(y)\circ\frac{\mathcal{H}_A(\Theta_A^{y})}{|\mathcal{A}_A^-(\underline{\mathbf{x_A}})|} \\
&\quad (6)\\
&= \sum_{\mathbf{x_B}\in\mathcal{C}_\simeq(B)} \mathcal{Q}_\tau(\mathbf{x_B},\mathbf{x_C})\circ\mathcal{Q}_\sigma(\mathbf{x_A},\mathbf{x_B})
\end{aligned}
$$

For (1), we use Lemma A.2.2, to obtain that matching compatible pairs of $\oplus$-covered configurations are necessarily minimal matching compatible. For (2), we simply use the definition of the valuation on the interaction. For (3), we use Proposition 9.4.10 and

Lemma 9.4.4. For (4), we use the definition of quantum $\sim$-strategies, more precisely the stability under symmetry of the valuation:

$$\mathcal{Q}_\tau(w_T) = \mathcal{H}_C(\psi_C^{-1}) \circ \mathcal{Q}_\tau(z) \circ \mathcal{H}_B(\psi_B) \qquad \mathcal{Q}_\sigma(w_S) = \mathcal{H}_B(\phi_B^{-1}) \circ \mathcal{Q}_\sigma(z) \circ \mathcal{H}_A(\psi_A)$$

And then we use the commuting diagrams of Proposition 9.4.10. For (5), we use Lemma 9.1.3 and linearity of the sum. For (6), we simply use the definitions. □

Similarly to the case without replication, we can replace the sum over $\mathcal{C}_\simeq(B)$ by a sum over $\mathcal{E}_\simeq(B)$ using the fact that strategies are winning.

Corollary 9.4.18. *For $\sigma, \tau \in \sim$-$\mathbf{QA}(A, B)$, and for $\mathbf{x_A} \in \mathcal{E}_\simeq(A), \mathbf{x_C} \in \mathcal{E}_\simeq(C)$ we have*

$$\mathcal{Q}_{\tau\odot\sigma}(\mathbf{x_A}, \mathbf{x_C}) = \sum_{\mathbf{x_B} \in \mathcal{E}_\simeq(B)} \mathcal{Q}_\tau(\mathbf{x_B}, \mathbf{x_C}) \circ \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$$

Proof. Since both σ and τ are winning, and since the symmetry of the game preserves the payoff, it means that whenever $\kappa_{B^\perp}(\underline{\mathbf{x_B}}) < 0$, we have $\mathcal{Q}_\tau(\mathbf{x_B}, \mathbf{x_C}) = 0$, and whenever $\kappa_B(\underline{\mathbf{x_B}}) < 0$, we have $\mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) = 0$. Since $\kappa_{B^\perp}(\underline{\mathbf{x_B}}) = -\kappa_B(\underline{\mathbf{x_B}})$, this means that we can eliminate from the sum every term where $\kappa_B(\underline{\mathbf{x_B}}) \neq 0$. □

9.4.5 Exhaustive Equivalence of Strategies with Symmetry

We can now define the exhaustive equivalence and prove it is a congruence. Similarly to the case without symmetry, the exhaustive equivalence "forgets branching points of the program", *i.e.*, makes equivalent the strategies for **if** $\mathrm{Coin}_{1/2}$ **then** () **else** () and (), and "forgets the evaluation order", *i.e.*, makes equivalent the strategies for $f_0(); f_1()$ and $f_1(); f_0()$. Additionally, the exhaustive equivalence will "forget copy indices", *i.e.*, strategies that are weakly isomorphic will be exhaustively equivalent.

Definition 9.4.19. *We say that two $\sim$-strategies $\sigma, \tau \in \sim$-$\mathbf{QA}(A, B)$ are exhaustively equivalent and write $\sigma \equiv \tau$ whenever*

$$\forall \mathbf{x_A} \in \mathcal{E}_\simeq(A), \forall \mathbf{x_B} \in \mathcal{E}_\simeq(B), \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) = \mathcal{Q}_\tau(\mathbf{x_A}, \mathbf{x_B})$$

Theorem 9.4.20. *The relation $\equiv$ is a congruence in $\sim$-$\mathbf{QA}$ and is compatible with all the additional structure. In particular, if $\sigma \simeq \tau$ then $\sigma \equiv \tau$.*

Proof. The congruence with respect to $\odot$ comes from Corollary 9.4.18. The fact that if $\sigma \simeq \tau$ then $\sigma \equiv \tau$ comes from the "Uniform" property of quantum $\sim$-strategies. The congruence with respect to the others constructs of $\sim$-$\mathbf{QA}$ is a direct verification. □

9.4.6 Collapse of Strategies into Relations on Arenas

Similarly to the case without symmetry, the exhaustive equivalence is tightly related to a collapse of $\sim$-**QA** into $\sim$-**QARel**.

Theorem 9.4.21 (Factorisation)**.** *We have a functor from* $\sim$-**QA** *into* $\sim$-**QARel***:*

$$\sigma \mapsto \mathcal{Q}_\sigma(-,-)$$

This functor preserves all the structure and is $\equiv$*-faithful: two* $\sim$*-strategies correspond to the same relations on* $\sim$*-arenas if and only if they are exhaustively equivalent. We have:*

The proof of this theorem is direct: the functoriality follows from Corollary 9.4.18, and Corollary 9.4.16 ensures that the image of the functor is indeed in $\sim$-**QARel**. The category $\sim$-**QARel** is significantly bigger than the image of $\sim$-**QA** by the functor, containing a lot of weighted relations that have no computational meaning, describing "systems" that execute some behaviours with "probabilities" greater than one, or even infinite when the annotations are in the infinitary fragment of $\overline{\mathbf{CPM}}$. While not all strategies of $\sim$-**QA** come from $\mathrm{Q}\Lambda_!$, we can extend to the case with symmetry the Theorem 5.3.15 which ensure that if we consider $\mathcal{Q}^{-,+}$ is always a superoperator, in other words the quantum valuations always correspond to physically realisable operations.

9.5 A Sound and ≡-Adequate Game Model

To prove that $\sim$-**QA** forms a sound and adequate model for $\mathrm{Q}\Lambda_!$, we first need to define the semantics of quantum closures of $\mathrm{Q}\Lambda_!$. The definition is essentially the same as for $\mathrm{Q}\Lambda$.

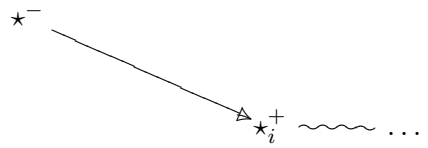
Definition 9.5.1. *For* $p_i \in [0,1]$ *for all* $i \in I$*, with* $\sum_{i\in I} p_i \leq 1$*, we define the quantum* $\sim$*-strategy* $\mathrm{choice}_{\{p_i \mid i\in I\}} : \mathbf{1} \to \oplus_{i\in I}\mathbf{1}$ *as described in Fig. 9.14. For* $\sigma_i : A \nrightarrow B$ *(*$i \in I$*) negative quantum strategies, we define* $\boxplus_{i\in I} p_i \cdot \sigma_i : A \nrightarrow B$ *with the copairing as follows:*

$$\boxplus_{i\in I} p_i \cdot \sigma_i := [\sigma_i \mid i \in I] \odot \left(\mathrm{choice}_{\{p_i \mid i\in I\}} \otimes A\right)$$

Lemma 9.5.2. *The operation* $\boxplus$ *is, up to strong isomorphism, associative, commutative, left-linear and semi-right-linear, i.e.,*

$$\begin{array}{rcll}
\boxplus_{i\in I} p_i \cdot \boxplus_{j\in J} q_j \cdot \sigma_{i,j} & \cong & \boxplus_{(i,j)\in I\times J} (p_i q_j) \cdot \sigma_{i,j} & \\
p\sigma \boxplus q\tau & \cong & q\tau \boxplus p\sigma & \\
\tau \odot \left(\boxplus_{i\in I} p_i \cdot \sigma_i\right) & \cong & \boxplus_{i\in I} p_i \cdot (\tau \odot \sigma_i) & \\
\left(\boxplus_{i\in I} p_i \cdot \sigma_i\right) \odot \tau & \cong & \boxplus_{i\in I} p_i \cdot (\sigma_i \odot \tau) & \text{whenever } \tau \text{ thunkable}
\end{array}$$

$$\mathbf{1} \qquad \nrightarrow \qquad \oplus_{i\in I}\mathbf{1}$$



Quantum valuation:

$$\mathcal{Q}(\varnothing) = \mathcal{Q}(\{\star^-\}) = \mathbf{id_1}$$
$$\forall i \in I, \mathcal{Q}(\{\star^-, \star_i^+\}) = p_i \cdot \mathbf{id_1}$$

Symmetry:

$$\theta : x \simeq y \text{ whenever } \theta = \mathbf{id}_x$$

Figure 9.14: Strategy $\text{choice}_{\{p_i \mid i\in I\}} : \mathbf{1} \to \oplus_{i\in I}\mathbf{1}$

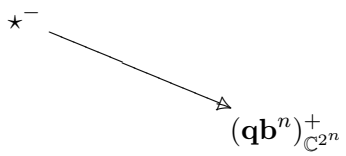
Moreover, it is right-linear and idempotent up to exhaustive equivalence:

$$\begin{array}{ccc} \left(\boxplus_{i\in I}\, p_i \cdot \sigma_i\right) \odot \tau & \cong & \boxplus_{i\in I}\, p_i \cdot (\sigma_i \odot \tau) \\ p \cdot \sigma \boxplus (1-p) \cdot \sigma & \equiv & \sigma \end{array}$$

Definition 9.5.3. *If $\vdash [q, \ell, t] : A$, we define $[\![[q, \ell, t]]\!]^{\vdash A} \in \sim\text{-}\mathbf{QA}(\mathbf{1}, [\![A]\!])$ as follows:*

- *we know we have $\Delta \vdash t : A$ with $\Delta = x_1 : \mathbf{qubit}, \ldots, x_n : \mathbf{qubit}$.*
- *we recall that $(\!|q|\!) \in \mathbf{CPM}(\mathbf{1}, \mathfrak{Q}^{\otimes n})$ is defined in Section 2.3.3, and define $q^{\mathbf{QA}}$ as in Fig. 9.15*

$$q^{\mathbf{QA}} : \qquad \mathbf{1} \qquad \nrightarrow \qquad \mathbf{qubit}^{\otimes n}$$



$$\begin{array}{rcl} \mathcal{Q}_{q^{\mathbf{QA}}}(\varnothing) & = & \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}} \\ \mathcal{Q}_{q^{\mathbf{QA}}}(\{\star\}) & = & \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}} \\ \mathcal{Q}_{q^{\mathbf{QA}}}(\{\star, (\mathbf{qb}^n)\}) & = & (\!|q|\!) \end{array}$$

Figure 9.15: The Strategy $q^{\mathbf{QA}} : \mathbf{1} \nrightarrow \mathbf{qubit}^{\otimes n}$ for q quantum state

- $[\![[q, \ell, t]]\!] := [\![t]\!] \odot q^{\mathbf{QA}}$

and we then define $[\![\sum_i p_i[q_i, \ell_i, t_i]]\!]$ *as* $\boxplus_i p_i [\![[q_i, \ell_i, t_i]]\!]$.

We can now extend the invariance lemma to closures.

Lemma 9.5.4 (≡-Invariance)**.** *For every closures* $\Gamma \vdash c : A$ *and* $\Gamma \vdash d : A$

$$c \to d \implies [\![c]\!] \equiv [\![d]\!]$$

This lemma can be either proven directly through a proof almost identical to the one for Lemma 8.4.12, or deduced from Lemma 8.4.12 through Theorem 9.4.21. In fact the same can be said for the soundness and ≡-adequacy, by following the same method as for ∼-**QARel** or by using the Theorem 9.4.21, we obtain the following:

Lemma 9.5.5 (Approximation Lemma)**.** *For* $\Gamma \vdash t : A$ *a term, if we write* t_n *for the term where every* **let rec** *has been replaced by* **let rec** [**n**] *(as defined in Table 7.1), then*

$$[\![t]\!] \simeq \lim_n [\![t_n]\!]$$

The same holds for closures.

Theorem 9.5.6 (Soundness and ≡-Adequacy)**.** *For every term* $\vdash t : \mathbf{1}$*, we have*

$$\mathbb{P}(t \Downarrow) = p \iff [\![t]\!] \equiv [\![p[\varnothing, \varnothing, ()] + (1-p)[\varnothing, \varnothing, \bot]]\!]$$

In particular, we have $[\![t]\!](\{\star\}, \{\star\}) \in \mathbf{CPM}(\mathbf{1}, \mathbf{1})$*, so finitary in* $\overline{\mathbf{CPM}}$.

Corollary 9.5.7. *For every pair of terms* $\Gamma \vdash t : A$ *and* $\Gamma \vdash s : A$*, we have*

$$[\![t]\!] \equiv [\![s]\!] \implies t =_{\text{obs}} s$$

This model is in fact fully abstract (both for LQΛ! and AQΛ!), and the proof of full abstraction is the subject of the next and last chapter (before the conclusion).

Chapter 10

Full Abstraction for the Quantum λ-calculus

10.1 Adding Formal Parameters

10.1.1 Motivation

The core of our proof of full abstraction for $\mathrm{Q}\Lambda$ was to build for every type A a set of test terms $\Downarrow_i^A$ (and generator terms) of type $A \multimap \mathbf{1}$ such that for every $\vdash t : A$ and $\vdash t' : A$, there exists one of those test terms $\Downarrow_i^A$ such that $\Downarrow_i^A \ t$ and $\Downarrow_i^A \ t'$ converge with distinct probabilities. In order to generalise the proof to $\mathrm{Q}\Lambda_!$, we "just" need to find such test terms (and generator terms) for the type $!(A \multimap B)$.

As an example, let us focus on the typing context $f :!(\mathbf{1} \multimap \mathbf{bit}) \vdash \mathbf{1}$. Here, we would like to find some generator terms $\vdash \Uparrow_i^{!(\mathbf{1} \multimap \mathbf{bit})} :!(\mathbf{1} \multimap \mathbf{bit})$ such that for every $f :!(\mathbf{1} \multimap \mathbf{bit}) \vdash t : \mathbf{1}$ and $f :!(\mathbf{1} \multimap \mathbf{bit}) \vdash t' : \mathbf{1}$ not observationally equivalents to one another, one of those generator terms $\Uparrow_i^{!(\mathbf{1} \multimap \mathbf{bit})}$ is such that $t\{x \leftarrow \Uparrow_i^{!(\mathbf{1} \multimap \mathbf{bit})}\}$ and $t'\{x \leftarrow \Uparrow_i^{!(\mathbf{1} \multimap \mathbf{bit})}\}$ converge with different probabilities. We consider the case

$$t = \mathrm{Ignore}(f()); \mathrm{Ignore}(f()); \mathbf{tt} \qquad\qquad t' = \mathbf{if} \ f() \ \mathbf{then} \ f() \ \mathbf{else} \ \mathrm{Not}(f())$$

where $\mathrm{Ignore}(b) = \mathbf{if} \ b \ \mathbf{then} \ () \ \mathbf{else} \ ()$ and $\mathrm{Not}(b) = \mathbf{if} \ b \ \mathbf{then} \ \mathbf{ff} \ \mathbf{else} \ \mathbf{tt}$. Both t and t' always call f twice, but to distinguish them, we need a generator term that can behave differently on two different calls, like for example

$$\vdash \lambda().\mathrm{Coin}_{1/2}() :!(\mathbf{1} \multimap \mathbf{bit})$$

However, the choice of probabilities here was arbitrary, and other terms might require other choices, for example if

$$t = \mathbf{if} \ f() \ \mathbf{then} \ f() \ \mathbf{else} \ \mathrm{Ignore}(f()); \mathrm{Coin}_{1/2}()$$

$$t' = \textbf{if } f() \textbf{ then } f() \textbf{ else } \mathrm{Not}(f())$$

then the previous generator term no longer distinguishes the two, and we have to take a different probability than 1/2. In general, to distinguish terms using ! in their type, we will need to use terms with probabilistic choices inside them, and we will be faced with the problem of proving that "at least one of the possible assignment of probability works".

Following the proof method of [ETP14], we do so by considering terms with formal parameters $X, Y, \ldots$ standing for probabilities, and we will postpone as much as possible the moment where we need to replace the formal parameters by a probability. In the previous example, the generator term would be:

$$\lambda(). \left(\frac{X}{2} \mathbf{ff} + \frac{Y}{2} \mathbf{tt} \right)$$

10.1.2 Q$\Lambda_!$ with Formal Parameters

We define the languages $\mathrm{LQ}\Lambda_!^{\mathrm{param}}$ and $\mathrm{AQ}\Lambda_!^{\mathrm{param}}$ as respectively LQ$\Lambda_!$ and AQ$\Lambda_!$ with the following additional primitive for terms. We use $\mathrm{Q}\Lambda_!^{\mathrm{param}}$ in statements that apply to both $\mathrm{LQ}\Lambda_!^{\mathrm{param}}$ and $\mathrm{AQ}\Lambda_!^{\mathrm{param}}$ indifferently.

$$t, s ::= \ldots \mid X \cdot t$$

where X is taken from a countably infinite set $\mathbb{F}$ of formal parameters, and will eventually be substituted with a probability in $[0, 1]$. We recall that we defined the syntactic sugar $p \cdot t$ with $p \in [0, 1]$ in Table 7.1. We write $\vdash_{\mathbb{L}}^{\mathrm{param}}, \vdash_{\mathbb{A}}^{\mathrm{param}}, \vdash^{\mathrm{param}}$ for the typing sequents of $\mathrm{LQ}\Lambda_!^{\mathrm{param}}$, $\mathrm{AQ}\Lambda_!^{\mathrm{param}}$ and $\mathrm{Q}\Lambda_!^{\mathrm{param}}$ respectively. The typing rules are the same as for LQ$\Lambda_!$, AQ$\Lambda_!$ and Q$\Lambda_!$ respectively, with the additional following rule:

Typing Rules:
$$\frac{\Gamma \vdash^{\mathrm{param}} t : A}{\Gamma \vdash^{\mathrm{param}} X \cdot t : A}$$

Syntactic Sugar:
$$pX \cdot t := p \cdot (X \cdot t)$$

We choose not to define an operational semantics to $\mathrm{Q}\Lambda_!^{\mathrm{param}}$, as it is only an intermediate language for the sake of defining the test and generator terms. We will focus on defining its game semantics, and then use the properties of this game semantics for $\mathrm{Q}\Lambda_!^{\mathrm{param}}$ to prove the game semantics for Q$\Lambda_!$ is fully abstract.

10.1.3 Formal Power Series

We start by defining formal power series, which are a generalisation of polynomials with a countably infinite number of terms. One of the core issues when considering formal power series is the convergence: which instantiations of the parameters give a convergent

infinite sum. We will consider formal power series with coefficient in $\mathbb{R}_{\geq 0}$, $\mathbf{CPM}(H,K)$ and $\overline{\mathbf{CPM}}(H,K)$ for any Hilbert spaces H and K. In the latter, the convergence is trivial, as the existence of suprema ensures that every infinite sum is converges. In particular, infinite sums of elements of $\overline{\mathbf{CPM}}(H,K)$ give the same result independently of the order of summing (Lemma 2.2.5). Since $\mathbf{CPM}(H,K)$ is a fragment of $\overline{\mathbf{CPM}}(H,K)$, it follows that while not every infinite sum might converges, when it does the result is independent from the order of summing. This also applies to $\mathbb{R}_{\geq 0}$, as it is isomorphic to $\mathbf{CPM}(\mathbf{1},\mathbf{1})$. In more general contexts, a notion of "absolute convergence" would be required to obtain this independence. We take $(C,+,\cdot,0)$ to be either $(\mathbb{R}_{\geq},+,\cdot,0)$, $(\mathbf{CPM}(H,K),+,\cdot,0)$, or $(\overline{\mathbf{CPM}}(H,K),+,\cdot,0)$ and write $\overline{C}$ for its D-completion for the induced order (or $\overline{\overline{\mathbf{CPM}}(H,K)} = \overline{\mathbf{CPM}}(H,K)$ in the latter case). We take $\mathbb{F}$ a countably infinite set of formal parameters, ranged over by $X, Y, \ldots$.

Definition 10.1.1. *A formal power series s with parameters $X_1,\ldots,X_n \in \mathbb{F}$ on C is a function from $\mathbb{N}^n$ to C. We write*

$$s = \sum_{(k_1,\ldots,k_n)\in\mathbb{N}^n} s(k_1,\ldots,k_n)\cdot X_1^{k_1}\ldots X_n^{k_n}$$

For $p_1,\ldots,p_n \in \overline{\mathbb{R}_{\geq 0}}$, we write

$$s[p_1,\ldots,p_n] := \sum_{(k_1,\ldots,k_n)\in\mathbb{N}^n} p_1^{k_1}\cdot\cdots\cdot p_n^{k_n}\cdot s(k_1,\ldots,k_n) \in \overline{C}$$

We say that s is $[0,1]$-convergent if

$$\forall p_1,\ldots,p_n \in [0,1], s[p_1,\ldots,p_n] \in C$$

We write $C[X_1,\ldots,X_n]$ for the positive convex cone (or completed positive convex cone) of $[0,1]$-convergent formal power series over C with parameters $X_1,\ldots,X_n$. The skeleton of a formal power series s is $s(1,\ldots,1)$, i.e., the coefficient of $X_1\ldots X_n$.

In the notation with sums, we will keep the terms of coefficient 0 and the parameters of exponent 0 implicit. So for example $1\cdot X_1 + 2\cdot X_2$ stands for the formal power series $s : (k_1,k_2) \mapsto 1$ if $(k_1,k_2) = (1,0)$ or $(k_1,k_2) = (0,1)$ and 0 otherwise. Using those notations, for every $\mathcal{X} \subseteq \mathcal{Y} \subseteq_{\text{fin}} \mathbb{F}$, we have a canonical embedding of $C[\mathcal{X}]$ into $C[\mathcal{Y}]$ which is simply the "identity".

$$\sum_{k\in\mathbb{N}^n} s_k\cdot X_1^{k_1}\ldots X_n^{k_n} \mapsto \sum_{k\in\mathbb{N}^n} s_k\cdot X_1^{k_1}\ldots X_n^{k_n}$$

In particular, we have a canonical embedding of $C = C[]$ in $C[\mathcal{X}]$. We will use very few results on formal power series, but one of them will be a central piece in the proof of full abstraction, as it will allow us to extract an instantiation of the formal parameters:

Theorem 10.1.2. *Let s, s' be two formal power series over $\mathbb{R}_{\geq 0}$, with the same finitely many parameters $X_1, \dots, X_n$. If s and s' are $[0,1]$-convergent, then*

$$s = s' \iff \forall p_1, \dots, p_n \in [0,1], s[p_1, \dots, p_n] = s'[p_1, \dots, p_n]$$

Proof. We consider $s - s'$, which is a power series from $\mathbb{R}^n$ to $\mathbb{R}$ which is absolutely convergent on $[0,1]^n$ and equal to zero on $[0,1]^n$, so using [Gou18], all the coefficients of the power series $s - s'$ are zero. □

10.2 Parametrised Game Semantics

We now define the parametrised game model that will allow us to represent $\mathrm{Q}\Lambda_!^{\mathrm{param}}$. Since $\sim$-**QA** is not a positive convex cone, we cannot simply take formal power series of quantum $\sim$-strategies. Instead, we consider $\sim$-strategies annotated by formal power series of quantum valuations. We note that the game semantics of $\mathrm{Q}\Lambda_!^{\mathrm{param}}$ will only use polynomials as quantum valuations, not infinite formal power series. However, since the collapsed quantum valuation $\mathcal{Q}(-,-)$ is potentially an infinite sum of quantum valuations, it might not always be a polynomial (see Fig. 10.1).

10.2.1 Parametrised Quantum Strategy with Symmetry

We assume a set of formal parameters $\mathbb{F}$, and $\mathcal{X} \subseteq_{\mathrm{fin}} \mathbb{F}$ a finite set of parameters.

Definition 10.2.1. *The category* $\mathbf{CPM}[\mathcal{X}]$ *is the category with the same objects as* **CPM**, *for morphisms* $\mathbf{CPM}[\mathcal{X}](A,B) := \mathbf{CPM}(A,B)[\mathcal{X}]$, *which we recall are $[0,1]$-convergent power series, for identity and composition the following:*

$$\mathbf{id}_A^{\mathbf{CPM}[\mathcal{X}]}(k_1, \dots, k_n) := \begin{cases} \mathbf{id}_A^{\mathbf{CPM}} & \text{whenever } k_1 = \dots = k_n = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\left(\sum_{k \in \mathbb{N}^n} g_k \cdot X_1^{k_1} \dots X_n^{k_n} \right) \circ \left(\sum_{j \in \mathbb{N}^n} f_j \cdot X_1^{j_1} \dots X_n^{j_n} \right) := \sum_{j,k \in \mathbb{N}^n} (g_k \circ f_j) \cdot X_1^{k_1 + j_1} \dots X_n^{k_n + j_n}$$

Or in other words:

$$(s \circ s')(k_1, \dots, k_n) := \sum_{\forall i, 0 \leq \ell_i \leq k_i} s(\ell_1, \dots, \ell_n) \circ s(k_1 - \ell_1, \dots, k_n - \ell_n)$$

The category $\mathbf{CPM}[\mathcal{X}]$ inherits the compact closure of **CPM**.

Definition 10.2.2. *For $s, s' \in \mathbf{CPM}[\mathcal{X}](A,B)$, we say that s is smaller than s' for the Loewner order, and write $s \sqsubseteq s'$, if for every instantiation $p_1, \dots, p_n \in [0,1]$ of the parameters of $\mathcal{X}$, we have*

$$s[p_1, \dots, p_n] \sqsubseteq s'[p_1, \dots, p_n]$$

Using the canonical embedding of $\mathbf{CPM}(A,B)[\mathcal{X}]$ in $\mathbf{CPM}(A,B)[\mathcal{Y}]$ whenever $\mathcal{X} \subset \mathcal{Y}$, we also have the following:

Definition 10.2.3. *The category* $\mathbf{CPM}[-]$ *is the category with the same objects as* $\mathbf{CPM}$*, for morphisms* $\mathbf{CPM}[-](A,B) := \bigcup_{\mathcal{X} \subseteq_{\text{fin}} \mathbb{F}} \mathbf{CPM}(A,B)[\mathcal{X}]$*, for identity and composition the following:* $\mathbf{id}_A^{\mathbf{CPM}[-]} := \mathbf{id}_A^{\mathbf{CPM}[]}$*, and for* $s \in \mathbf{CPM}[\mathcal{X}](A,B)$ *and* $s' \in \mathbf{CPM}[\mathcal{Y}](B,C)$*, we compose them by embedding them in* $\mathbf{CPM}[\mathcal{X} \cup \mathcal{Y}](A,B)$ *and* $\mathbf{CPM}[\mathcal{X} \cup \mathcal{Y}](B,C)$ *respectively, and then composing them as in* $\mathbf{CPM}[\mathcal{X} \cup \mathcal{Y}]$*.*

The category $\mathbf{CPM}[-]$ inherits the compact closure of $\mathbf{CPM}$. And we can see $\mathbf{CPM}$ as a subcategory of $\mathbf{CPM}[-]$, meaning that the functor $\mathcal{H}$ can be seen as a contravariant functor from the $\sim$-Scott category to $\mathbf{CPM}[-]$.

We extend the definition of quantum $\sim$-strategies to the parametrised case.

Definition 10.2.4. *A parametrised quantum* $\sim$*-strategy* $\sigma : A \nrightarrow B$ *is a* $\sim$*-strategy together with a finite set of parameters* $\mathcal{X}_\sigma$*, and a parametrised quantum valuation* $\mathcal{Q}$ *on configurations* $x \in \mathcal{C}(S)$ *satisfying the same properties as the quantum valuations of quantum* $\sim$*-strategies:*

$$\sigma\, x = x_A \parallel x_B \implies \mathcal{Q}_\sigma(x) \in \mathbf{CPM}[\mathcal{X}_\sigma](\mathcal{H}_A(x_A), \mathcal{H}_B(x_B))$$

Normalisation $\mathcal{Q}_\sigma(\varnothing) = \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}[\mathcal{X}_\sigma]}$

Obliviousness $x \subseteq^- x' \implies \mathcal{Q}_\sigma(x') = \mathcal{H}_B(x'^B \ {}^-\!\supseteq x^B) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(x^A \subseteq^+ x'^A)$

Drop condition $x \subseteq^+ x_1, \ldots, x_n \implies d_\sigma^{\text{odd}}(x; x_1, \ldots, x_n) \sqsubseteq d_\sigma^{\text{even}}(x; x_1, \ldots, x_n)$*, where*

$$d_\sigma^{\text{odd}}(x; x_1, \ldots, x_n) := \sum_{\substack{\varnothing \neq I \subseteq \{1,\ldots,n\} \\ |I| \text{ odd} \\ x_I \in \mathcal{C}(S)}} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)$$

$$d_\sigma^{\text{even}}(x; x_1, \ldots, x_n) := \mathcal{Q}_\sigma(x) + \sum_{\substack{\varnothing \neq I \subseteq \{1,\ldots,n\} \\ |I| \text{ even} \\ x_I \in \mathcal{C}(S)}} \mathcal{H}_B(x^B \subseteq^+ x_I^B) \circ \mathcal{Q}_\sigma(x_I) \circ \mathcal{H}_A(x_I^A \ {}^-\!\supseteq x^A)$$

and $x_I := \bigcup_{i \in I} x_i$*.*

Winning x *$\oplus$-covered* $\implies \kappa_{A^\perp \parr B}(\sigma\, x) \geq 0$*.*

Uniform $\theta : x \simeq_S x' \implies \mathcal{Q}_\sigma(y) = \mathcal{H}_B(\theta_B^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\theta_A)$*, where* $\sigma\, \theta = \theta_A \parallel \theta_B$*.*

Equivalently, one could ask that every instantiation of the parameters of $\mathcal{X}_\sigma$ in $[0,1]$ gives rise to a quantum $\sim$-strategy. For the interactive composition, we simply take the interactive composition of $\sim$-strategies together with

$$\mathcal{X}_{\tau \odot \sigma} = \mathcal{X}_\tau \cup \mathcal{X}_\sigma \qquad \mathcal{Q}_{\tau \odot \sigma}(y \odot x) = \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x)$$

We define the copy-cat parametrised quantum $\sim$-strategy as the copy-cat quantum $\sim$-strategy together with $\mathcal{X}_{\mathfrak{a}_A} = \varnothing$. We extend strong isomorphism, weak isomorphism, and weak equivalence by taking the same definition and asking $\mathcal{X}_\sigma$ to be preserved.

Proposition 10.2.5. *The category $\sim$-**QA**$[-]$ of quantum $\sim$-arenas and parametrised negative visible quantum $\sim$-strategies up to weak isomorphism forms a pre-model for $LQ\Lambda_!$, as defined in Section 7.2, and when restricted to affine quantum $\sim$-arenas it forms up to weak isomorphism a pre-model for $AQ\Lambda_!$.*

To represent terms of the form $X \cdot t$ in $\mathrm{Q}\Lambda_!^{\mathrm{param}}$, we use

$$\left[\!\!\left[\dfrac{\begin{array}{c}\mathrm{T}\\ \vdots \\ \Gamma \vdash^{\mathrm{param}} t : A\end{array}}{\Gamma \vdash^{\mathrm{param}} X \cdot t : A} \right]\!\!\right] := X \cdot [\![T]\!]$$

Where $X \cdot \sigma$ is defined as follows.

Definition 10.2.6. *For $\sigma \in \sim$-**QA**$[-](A, B)$, and $X \in \mathbb{F}$, we define $X \cdot \sigma \in \sim$-**QA**$[-](A, B)$ as the same $\sim$-strategy as σ but with*

$$\mathcal{X}_{X\cdot\sigma} = \mathcal{X}_\sigma \cup \{X\} \qquad \mathcal{Q}_{X\cdot\sigma}(x) = \begin{cases} X \cdot \mathcal{Q}(x) & \text{whenever } \exists e \in x, p_S(e) = \oplus \\ \mathcal{Q}(x) & \text{whenever } \forall e \in x, p_S(e) = \ominus \end{cases}$$

Lemma 10.2.7. *The above definition gives a parametrised quantum $\sim$-strategy.*

Proof. All the conditions are trivial to check but the drop condition. We consider $x \subseteq^+ x_1, \ldots, x_n$. Using Lemma 5.3.4, we can assume without loss of generality that $x \subset^+ x_1, \ldots, x_n$, which ensures that every x_i contains at least a positive event. If x contains a positive event, then we have

$$\begin{array}{rcl} d^{\mathrm{odd}}_{X\cdot\sigma}(x; x_1, \ldots, x_n) & = & X \cdot d^{\mathrm{odd}}_\sigma(x; x_1, \ldots, x_n) \\ & \sqsubseteq & X \cdot d^{\mathrm{even}}_\sigma(x; x_1, \ldots, x_n) \\ & = & d^{\mathrm{even}}_{X\cdot\sigma}(x; x_1, \ldots, x_n) \end{array}$$

If x does not contain a positive event, then using $X \cdot \mathcal{Q}_{X\cdot\sigma}(x) \sqsubseteq \mathcal{Q}_{X\cdot\sigma}(x)$ we obtain

$$\begin{aligned}
d^{\mathrm{odd}}_{X\cdot\sigma}(x; x_1, \dots, x_n) &= X \cdot d^{\mathrm{odd}}_{\sigma}(x; x_1, \dots, x_n) \\
&\sqsubseteq X \cdot d^{\mathrm{even}}_{\sigma}(x; x_1, \dots, x_n) \\
&= X \cdot \mathcal{Q}_\sigma(x) + \sum_{\substack{\varnothing \neq I \subseteq \{1,\dots,n\} \\ |I| \text{ even} \\ x_I \in \mathcal{C}(S)}} \mathcal{H}_B(x^B \subseteq^+ x^B_I) \circ (X \cdot \mathcal{Q}_\sigma(x_I)) \circ \mathcal{H}_A(x^A_I \ {}^- \!\supseteq x^A) \\
&= X \cdot \mathcal{Q}_{X\cdot\sigma}(x) + \sum_{\substack{\varnothing \neq I \subseteq \{1,\dots,n\} \\ |I| \text{ even} \\ x_I \in \mathcal{C}(S)}} \mathcal{H}_B(x^B \subseteq^+ x^B_I) \circ \mathcal{Q}_{X\cdot\sigma}(x_I) \circ \mathcal{H}_A(x^A_I \ {}^- \!\supseteq x^A) \\
&\sqsubseteq \mathcal{Q}_{X\cdot\sigma}(x) + \sum_{\substack{\varnothing \neq I \subseteq \{1,\dots,n\} \\ |I| \text{ even} \\ x_I \in \mathcal{C}(S)}} \mathcal{H}_B(x^B \subseteq^+ x^B_I) \circ \mathcal{Q}_{X\cdot\sigma}(x_I) \circ \mathcal{H}_A(x^A_I \ {}^- \!\supseteq x^A) \\
&= d^{\mathrm{even}}_{X\cdot\sigma}(x; x_1, \dots, x_n)
\end{aligned}$$

□

As said earlier, for every parametrised term $\Gamma \vdash^{\mathrm{param}} t : A$, its semantics $[\![t]\!]$ only uses polynomials, *i.e.*, for every configuration x of $[\![t]\!]$, $\mathcal{Q}_{[\![t]\!]}(x)$ is a formal power series with only a finite number of non-zero coefficients. This can be proved by an immediate induction on the typing derivation of t, as all the operations on strategies used in the interpretation obviously preserve this invariant.

10.2.2 The Exhaustive Equivalence

We now extend the exhaustive equivalence to the parametrised case. The definitions are the same as in the non-parametric case.

Definition 10.2.8. *For $\sigma \in \sim\text{-}\mathbf{QA}[-](A, B)$ and $\lceil x \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S)$, we define*

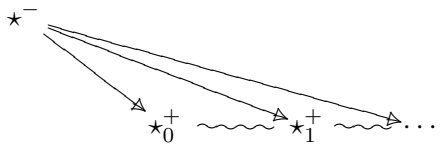
$$\mathcal{Q}_\sigma(\lceil x \rceil) := \mathcal{H}_B((\Theta^x_B)^{-1}) \circ \mathcal{Q}_\sigma(x) \circ \mathcal{H}_A(\Theta^x_A)$$

Then for $\mathbf{x_A} \in \mathcal{C}_\simeq(A)$ and $\mathbf{x_B} \in \mathcal{C}_\simeq(B)$, we define

$$\begin{aligned}
\mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) &:= \sum_{\lceil x \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{Q}_\sigma(\lceil x \rceil)}{|\mathcal{A}^+_{A^\perp \parr B}(\mathbf{x_A} \| \mathbf{x_B})|} \qquad \in \overline{\mathbf{CPM}}[-](\mathcal{H}_A(\underline{\mathbf{x_A}}), \mathcal{H}_B(\underline{\mathbf{x_B}})) \\
&= \sum_{\lceil x \rceil \in \overset{+}{\sim}\text{-wit}_\sigma(\mathbf{x_A}, \mathbf{x_B})} \frac{\mathcal{H}_B((\Theta^x_B)^{-1})}{|\mathcal{A}^+_B(\underline{\mathbf{x_B}})|} \circ \mathcal{Q}_\sigma(x) \circ \frac{\mathcal{H}_A(\Theta^x_A)}{|\mathcal{A}^-_A(\underline{\mathbf{x_A}})|}
\end{aligned}$$

where $\overline{\mathbf{CPM}}[-](H, K)$ is defined similarly to $\mathbf{CPM}[-](H, K)$.

$$\varnothing \qquad \vdash^{\text{param}}_{\mathbb{L}} \qquad \mathbf{1}$$



Quantum valuation:

$$\mathcal{Q}(\varnothing) = \mathcal{Q}(\{\star^-\}) = \mathbf{id_1} \qquad \mathcal{Q}(\{\star^-, \star_n^+\}) = \frac{X^n}{2^{n+1}} \cdot \mathbf{id_1}$$

Term represented:

$$\vdash^{\text{param}}_{\mathbb{L}} \textbf{let rec } f\ () = \textbf{if } \mathrm{Coin}_{1/2} \textbf{ then } X \cdot f() \textbf{ else } () \textbf{ in } f() : \mathbf{1}$$

Figure 10.1: Example of non-polynomial collapsed quantum valuation.

Proposition 10.2.9. *For $\sigma \in \sim\text{-}\mathbf{QA}[-](A,B)$ and $\tau \in \sim\text{-}\mathbf{QA}[-](B,C)$, which we recall are visible $\sim$-strategies, and for $\mathbf{x_A} \in \mathcal{C}_\simeq(A), \mathbf{x_C} \in \mathcal{C}_\simeq(C)$ we have*

$$\mathcal{Q}_{\tau\odot\sigma}(\mathbf{x_A}, \mathbf{x_C}) = \sum_{\mathbf{x_B} \in \mathcal{C}_\simeq(B)} \mathcal{Q}_\tau(\mathbf{x_B}, \mathbf{x_C}) \circ \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B})$$

The proof is the same as in the non-parametric case.

Definition 10.2.10. *We say that two $\sim$-strategies $\sigma, \tau \in \sim\text{-}\mathbf{QA}[-](A,B)$ are exhaustively equivalent and write $\sigma \equiv \tau$ whenever*

$$\forall \mathbf{x_A} \in \mathcal{E}_\simeq(A), \forall \mathbf{x_B} \in \mathcal{E}_\simeq(B), \mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) = \mathcal{Q}_\tau(\mathbf{x_A}, \mathbf{x_B})$$

Corollary 10.2.11. *The relation $\equiv$ is a congruence on $\sim\text{-}\mathbf{QA}[-]$.*

While for every parametrised term $\Gamma \vdash^{\text{param}} t : A$, its semantics $[\![t]\!]$ only uses polynomials, it does not mean that its collapsed valuations $\mathcal{Q}_{[\![t]\!]}(-,-)$ are necessarily polynomials. In fact, in the example Fig. 10.1, we have

$$\mathcal{Q}(\{\star^-\}, \{\star^+\}) = \sum_{n \geq 0} \frac{X^n}{2^{n+1}}$$

10.2.3 The Skeleton of a Strategy

We defined the skeleton of a formal power series as the coefficient of $X_1 \dots X_n$, *i.e.*, the coefficient of the monomial where every formal parameter appears exactly once. We have a similar notion in game semantics.

Definition 10.2.12. *For $\sigma \in \sim\text{-}\mathbf{QA}[-](A,B)$, we define $\mathbf{skl}_\sigma(x)$ (for $x \in \mathcal{C}(S)$), $\mathbf{skl}_\sigma(\lceil y \rceil)$ (for $\lceil y \rceil \in \overset{+}{\sim}\text{-}\mathcal{C}(S)$) and $\mathbf{skl}_\sigma(\mathbf{z_A}, \mathbf{z_B})$ (for $\mathbf{z_A} \in \mathcal{C}_\simeq(A)$, $\mathbf{z_B} \in \mathcal{C}_\simeq(B)$) as the skeleton for the set of parameters $\mathcal{X}_\sigma$ of $\mathcal{Q}_\sigma(x)$, $\mathcal{Q}_\sigma(\lceil y \rceil)$ and $\mathcal{Q}_\sigma(\mathbf{z_A}, \mathbf{z_B})$ respectively.*

In the next section, every formal parameter will be associated to a specific "call" to a non-linear function, so "one of each formal parameter" will correspond to "every subterm decorated with a formal parameter is visited exactly once".

Lemma 10.2.13. *For $\sigma \in \sim\text{-}\mathbf{QA}[-](A,B)$ and $\tau \in \sim\text{-}\mathbf{QA}[-](B,C)$, if $\mathcal{X}_\sigma \cap \mathcal{X}_\tau = \varnothing$ then*

$$\begin{array}{lrcl}
\forall y \odot x \in \mathcal{C}(T \odot S), & \mathbf{skl}_{\tau\odot\sigma}(y \odot x) & = & \mathbf{skl}_\tau(y) \circ \mathbf{skl}_\sigma(x) \\
\forall \mathbf{z_A} \in \mathcal{C}_\simeq(A), \mathbf{z_C} \in \mathcal{C}_\simeq(C) & \mathbf{skl}_{\tau\odot\sigma}(\mathbf{z_A}, \mathbf{z_C}) & = & \sum_{\mathbf{z_B} \in \mathcal{C}_\simeq(B)} \mathbf{skl}_\tau(\mathbf{z_B}, \mathbf{z_C}) \circ \mathbf{skl}_\sigma(\mathbf{z_A}, \mathbf{z_B}) \\
\forall \mathbf{z_A} \in \mathcal{E}_\simeq(A), \mathbf{z_C} \in \mathcal{E}_\simeq(C) & \mathbf{skl}_{\tau\odot\sigma}(\mathbf{z_A}, \mathbf{z_C}) & = & \sum_{\mathbf{z_B} \in \mathcal{E}_\simeq(B)} \mathbf{skl}_\tau(\mathbf{z_B}, \mathbf{z_C}) \circ \mathbf{skl}_\sigma(\mathbf{z_A}, \mathbf{z_B})
\end{array}$$

This lemma follows from the corresponding lemmas on quantum $\sim$-strategies.

10.3 Full Abstraction

Similarly to the case without symmetry, the proof of full abstraction relies on test and generator terms. We start by proving full abstraction for $\mathrm{LQ}\Lambda_!$.

10.3.1 Extended Configurations of an Arena

In order to define test and generator terms in Tables 6.2 and 6.3, we used a notion of extended web, which was the web of a type together with some indices in $\{0,1,2,3\}$, one for each qubit. Their goal was to specify for each qubit which of the four test or generator term we want to use. We lift this notion onto arenas.

Definition 10.3.1. *An extended event of a quantum $\sim$-arena A is (a,i) with $a \in |A|$ and $i \in \{0,1,2,3\}^{\log(\dim(\mathcal{H}_A(a)))}$. We write $|A|^e$ for the set of extended events of A.*

This notion is only well-defined when Hilbert space annotations have dimensions that are a power of 2. It is immediate that it is satisfied for any $\sim$-arena arising from a type of $\mathrm{Q}\Lambda_!^{\mathrm{param}}$. An extended event is simply an event together with one index in $\{0,1,2,3\}$ for each qubit it represents. For $x \in \mathcal{C}(A)$ and $i \in \{0,1,2,3\}^{\log(\dim(\mathcal{H}_A(x)))}$ we write $x|i \subseteq |A|^e$ for the extended configuration where we slitted the index of i along the configuration. Every extended configuration of $|A|^e$ can be uniquely decomposed as $x|i$ for some x and i.

Definition 10.3.2. *An extended configuration of a quantum $\sim$-arena A is $x \in \mathcal{P}_{\mathrm{fin}}(|A|^e)$ such that $x = x'|i$ for $x' \in \mathcal{C}(A)$ and $i \in \{0,1,2,3\}^{\log(\dim(\mathcal{H}_A(x)))}$. We write $\mathcal{C}^e(A)$ for the set of extended configurations of A, and $\mathcal{E}^e(A)$ the set of exhaustive ones (meaning that $x' \in \mathcal{E}(A)$).*

We extend the operations $\otimes, \multimap, \oplus, [\dots]$ from configurations to extended configurations. In practice, we will only use canonical extended exhaustive configurations $\underline{\mathbf{x}}|i$, using the fact that $\underline{\mathbf{x} \multimap \mathbf{y}} = \underline{\mathbf{x}} \multimap \underline{\mathbf{y}}$ (and same for the other constructs) by definition of $\multimap$ (and by definition of the other constructs).

10.3.2 Parametrised Test and Generator Terms

We provide in Tables 10.1 and 10.2 the test and generator terms we will be using. There are a few notable differences with the terms provided in Tables 6.2 and 6.3:

- Rather than indexing parameters by points of a web, we index them by exhaustive extended configurations of a $\sim$-arena: when we write $\Downarrow_x^A$ or $\Uparrow_x^A$, we have $x \in \mathcal{E}^e(\llbracket A \rrbracket)$, as defined above. This difference is purely a presentation choice.
- We give terms for types of the form $!(A \multimap B)$, which allows us to cover all the types of $\mathrm{Q}\Lambda_!^{\mathrm{param}}$. Exhaustive extended configurations of $!(A \multimap B)$ are of the form $(x_1 \otimes \dots \otimes x_n)$ with x_i an exhaustive extended configuration of $A \multimap B$. For clarity, we treat the case $n = 0$ separately.
- Some terms use formal parameters. Parameters used are always fresh, *i.e.*, chosen such that generator and test terms never use the same parameter multiple times in them, and when we consider two generator and/or test terms, they are implicitly assumed to have disjoint sets of parameters.

10.3.3 Example

We come back to the example from earlier in this chapter:

$$t = \mathrm{Ignore}(f()); \mathrm{Ignore}(f()); \mathbf{tt} \qquad t' = \mathbf{if}\ f()\ \mathbf{then}\ f()\ \mathbf{else}\ \mathrm{Not}(f())$$

We describe their semantics in Fig. 10.2. Both $\sim$-strategies have trivial symmetry and quantum valuation, and only differ by the final boolean output. Those two $\sim$-strategies are not exhaustively equivalent, and in particular they differ on $\mathbf{x_\Gamma} \parallel \mathbf{x_A}$ with $\mathbf{x_\Gamma} = \{\lambda, \star_{\mathbf{0}}, \mathbf{ff_0}, \star_{\mathbf{1}}, \mathbf{tt_1}\}$ and $\mathbf{x_A} = \{\mathbf{tt}\}$. For t, we have two configurations of the strategy that match the canonical configuration up to positive symmetry, while we have none for t', so:

$$\begin{array}{lcl} \mathcal{Q}_{\llbracket t \rrbracket}(\mathbf{x_\Gamma}, \mathbf{x_A}) & = & \mathcal{Q}_{\llbracket t \rrbracket}(\{\lambda, \mathbf{ff}_0, \star_1, \mathbf{tt}_1, \mathbf{tt}\}) + \mathcal{Q}_{\llbracket t \rrbracket}(\{\lambda, \mathbf{tt}_0, \star_2, \mathbf{tt}_2, \mathbf{tt}\}) \quad = \quad 2 \cdot \mathbf{id_1} \\ \mathcal{Q}_{\llbracket t' \rrbracket}(\mathbf{x_\Gamma}, \mathbf{x_A}) & = & 0 \end{array}$$

$$
\begin{array}{lcllcl}
\Uparrow^{\mathbf{1}}_{\{\star^+\}} & := & () & & & \\
\Uparrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},0)\}} & := & \mathbf{new}\ \mathbf{ff} & \Uparrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},1)\}} & := & \mathbf{new}\ \mathbf{tt} \\
\Uparrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},2)\}} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} (\mathbf{new}\ \mathbf{ff}) & \Uparrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},3)\}} & := & \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{\mathbf{i}}{\sqrt{2}} \\ \frac{\mathbf{i}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} (\mathbf{new}\ \mathbf{ff}) \\
\\
\Uparrow^{A\otimes B}_{x_A\otimes x_B} & := & \Uparrow^A_{x_A} \otimes \Uparrow^B_{x_B} & \Uparrow^{A\multimap B}_{x_A\multimap x_B} & := & \lambda x.\ \Downarrow^A_{x_A} x; \Uparrow^B_{x_B} \\
\Uparrow^{A\oplus B}_{x_A\oplus\varnothing} & := & \mathbf{inj}_\ell\ \Uparrow^A_{x_A} & \Uparrow^{A\oplus B}_{\varnothing\oplus x_B} & := & \mathbf{inj}_r\ \Uparrow^B_{x_B} \\
\Uparrow^{A^\ell}_{[\,]} & := & [\,] & \Uparrow^{A^\ell}_{[x_1;\ldots;x_n]} & := & \Uparrow^A_{x_1} :: \Uparrow^{A^\ell}_{[x_2;\ldots;x_n]} \\
\\
\Uparrow^{!F}_{\{\lambda^+\}} & := & \lambda x.\bot & \Uparrow^{!F}_{x_1\otimes\ldots\otimes x_n} & := & \lambda v.\ \sum_{i=1}^n \frac{X_i}{n} \cdot \left(\Uparrow^F_{x_i}\ v\right)
\end{array}
$$

Table 10.1: Generator terms $\vdash^{\text{param}}_{\mathbb{L}} \Uparrow^A_a : A$

$$
\begin{array}{lcl}
\Downarrow^{\mathbf{1}}_{\{\star^+\}} & := & \lambda().() \\
\Downarrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},0)\}} & := & \lambda q.\mathbf{if}\ \mathbf{meas}\ q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},1)\}} & := & \lambda q.\mathbf{if}\ \mathbf{meas}\ q\ \mathbf{then}\ ()\ \mathbf{else}\ \bot \\
\Downarrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},2)\}} & := & \lambda q.\mathbf{if}\ \mathbf{meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\Downarrow^{\mathbf{qubit}}_{\{(\mathbf{qb}^+_{\mathbb{Q}},3)\}} & := & \lambda q.\mathbf{if}\ \mathbf{meas} \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{\mathbf{i}}{\sqrt{2}} \\ -\frac{\mathbf{i}}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} q\ \mathbf{then}\ \bot\ \mathbf{else}\ () \\
\\
\Downarrow^{A\otimes B}_{x_A\otimes x_B} & := & \lambda(x\otimes y).\ \Downarrow^A_{x_A} x\ ; \Downarrow^B_{x_B} y \\
\Downarrow^{A\multimap B}_{x_A\multimap x_B} & := & \lambda f.\mathbf{let}\ x = f\ \Uparrow^A_{x_A}\ \mathbf{in}\ \Downarrow^B_{x_B} x \\
\Downarrow^{A\oplus B}_{x_A\oplus\varnothing} & := & \lambda x.\delta\ (x,\ y.\ \Downarrow^A_{x_A} y,\ z.\bot) \\
\Downarrow^{A\oplus B}_{\varnothing\oplus x_B} & := & \lambda x.\delta\ (x,\ y.\bot,\ z.\ \Downarrow^B_{x_B} z) \\
\Downarrow^{A^\ell}_{[\,]} & := & \lambda\ell.\mathbf{match}\ \ell\ \mathbf{with}\ ([\,] \mapsto () \mid x :: y \mapsto \bot) \\
\Downarrow^{A^\ell}_{[x_1;\ldots;x_n]} & := & \lambda\ell.\mathbf{match}\ \ell\ \mathbf{with}\ \left([\,] \mapsto \bot \;\middle|\; x :: y \mapsto \Downarrow^A_{x_1} x\ ; \Downarrow^{A^\ell}_{[x_2;\ldots;x_n]} y\right) \\
\\
\Downarrow^{!F}_{\{\lambda^+\}} & := & \lambda f.() \\
\Downarrow^{!F}_{x_1\otimes\ldots\otimes x_n} & := & \lambda f^{!F}.\ \Downarrow^F_{x_1} f\ ; \ldots ;\ \Downarrow^F_{x_n} f
\end{array}
$$

Table 10.2: Tests term $\vdash^{\text{param}}_{\mathbb{L}} \Downarrow^A_a : A \multimap \mathbf{1}$

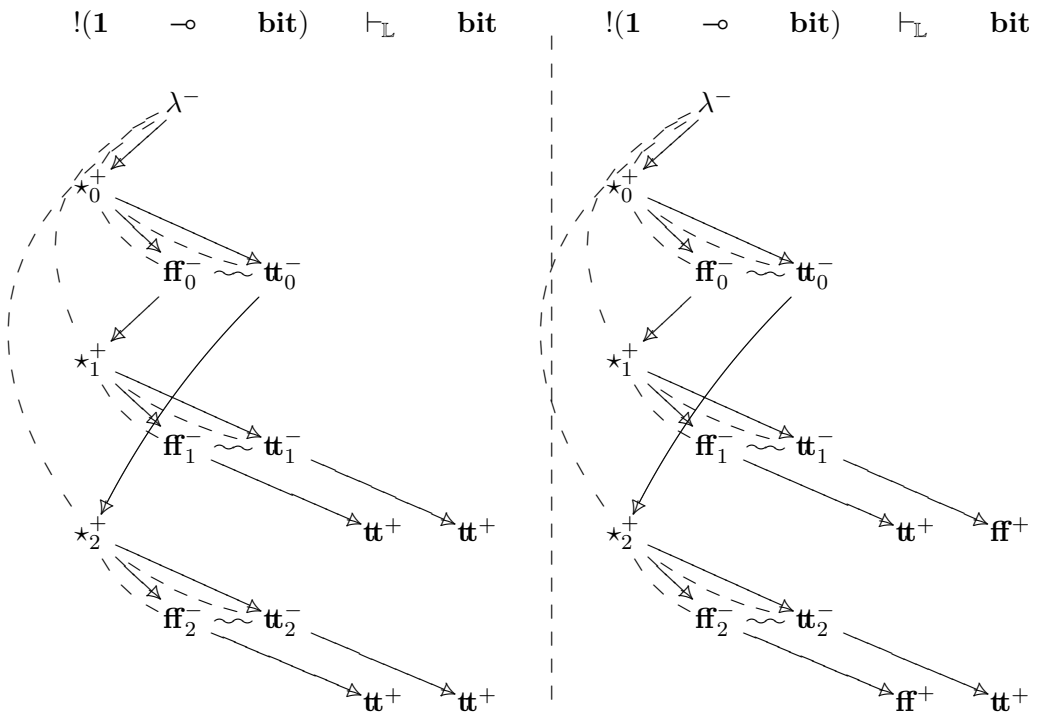


Figure 10.2: The strategies for t and t' respectively.

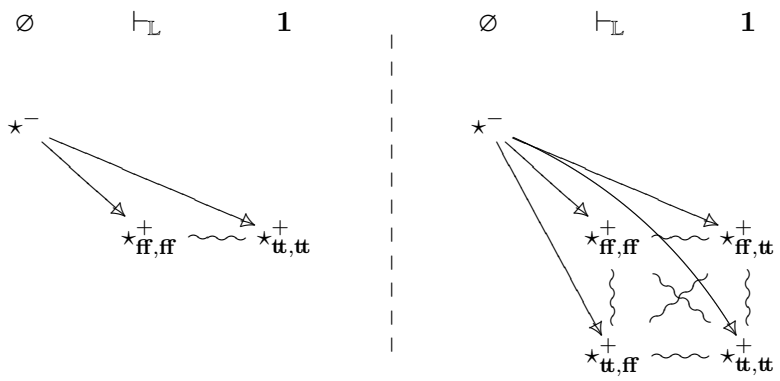


Figure 10.3: The strategies for t_0 and t'_0 respectively.

We now look at the test and generator terms. We have

$$\begin{array}{lcl} \Downarrow^{\mathbf{bit}}_{\mathbf{x_A}} & = & \lambda b.\mathbf{if}\ b\ \mathbf{then}\ ()\ \mathbf{else}\ \bot \\ \Uparrow^{!(\mathbf{1}\multimap\mathbf{bit})}_{\mathbf{x_\Gamma}} & = & \lambda().\frac{X}{2}\mathbf{ff} + \frac{Y}{2}\mathbf{tt} \end{array}$$

If we use those test and generator terms as context for t and t', we obtain

$$\begin{array}{lcl} t_0 & := & \mathbf{let}\ f = \Uparrow^{!(\mathbf{1}\multimap\mathbf{bit})}_{\mathbf{x_\Gamma}}\ \mathbf{in}\ \Downarrow^{\mathbf{bit}}_{\mathbf{x_A}}\ t \\ t'_0 & := & \mathbf{let}\ f = \Uparrow^{!(\mathbf{1}\multimap\mathbf{bit})}_{\mathbf{x_\Gamma}}\ \mathbf{in}\ \Downarrow^{\mathbf{bit}}_{\mathbf{x_A}}\ t' \end{array}$$

If we compute their semantics, we obtain strategies described in Fig. 10.3, where each event $\star_{b,b'}$ corresponds to the branch of the computation where the first call to f returned b and the second call to f returned b'. Those quantum $\sim$-strategies have trivial symmetry and

$$\begin{array}{ll} \mathcal{Q}_{[\![t_0]\!]}(\{\star^-,\star^+_{\mathbf{ff},\mathbf{ff}}\}) = \frac{1}{4}\mathbf{id_1}XX & \mathcal{Q}_{[\![t'_0]\!]}(\{\star^-,\star^+_{\mathbf{ff},\mathbf{ff}}\}) = \frac{1}{4}\mathbf{id_1}XX \\ & \mathcal{Q}_{[\![t'_0]\!]}(\{\star^-,\star^+_{\mathbf{ff},\mathbf{tt}}\}) = \frac{1}{4}\mathbf{id_1}XY \\ & \mathcal{Q}_{[\![t'_0]\!]}(\{\star^-,\star^+_{\mathbf{tt},\mathbf{ff}}\}) = \frac{1}{4}\mathbf{id_1}YX \\ \mathcal{Q}_{[\![t_0]\!]}(\{\star^-,\star^+_{\mathbf{tt},\mathbf{tt}}\}) = \frac{1}{4}\mathbf{id_1}YY & \mathcal{Q}_{[\![t'_0]\!]}(\{\star^-,\star^+_{\mathbf{tt},\mathbf{tt}}\}) = \frac{1}{4}\mathbf{id_1}YY \end{array}$$

So, if we look at the collapsed quantum valuation we have

$$\begin{array}{lcl} \mathcal{Q}_{[\![t_0]\!]}(\{\star\},\{\star\}) & = & \frac{1}{4}\mathbf{id_1}XX + \frac{1}{4}\mathbf{id_1}YY \\ \mathcal{Q}_{[\![t'_0]\!]}(\{\star\},\{\star\}) & = & \frac{1}{4}\mathbf{id_1}XX + \frac{1}{4}\mathbf{id_1}YY + \frac{2}{4}\mathbf{id_1}XY \end{array}$$

Those two formal power series differ on a coefficient: the coefficient of XY, *i.e.*, the skeleton of the formal power series. In fact, we have

$$\begin{array}{lcl} \mathbf{skl}_{[\![t_0]\!]}(\{\star\},\{\star\}) & = & \frac{1}{4}\mathcal{Q}_{[\![t]\!]}(\mathbf{x_\Gamma},\mathbf{x_A}) \\ \mathbf{skl}_{[\![t'_0]\!]}(\{\star\},\{\star\}) & = & \frac{1}{4}\mathcal{Q}_{[\![t']\!]}(\mathbf{x_\Gamma},\mathbf{x_A}) \end{array}$$

We note the importance of the skeleton: in order to "replay" the configuration x_Γ, we have to ensure that f is called exactly once with each of the desired inputs and output, and each of the two formal parameters X and Y corresponds to a "correct call", so we want exactly one of each.

10.3.4 Properties of Test and Generator Terms

As in the case without symmetry, the goal of test terms is to "extract" the coefficient corresponding to a point of the web by "replaying" a given configuration. Formally, we expect that for any non-parametrised term $x : A \vdash_{\mathbb{L}} t : B$, we have:

$$\mathbf{skl}\left[\!\left[\mathbf{let}\ x^A = \Uparrow^A_{\underline{\mathbf{x_A}}|i}\ \mathbf{in}\ \Downarrow^B_{\underline{\mathbf{x_B}}|j}\ t\right]\!\right](\{\star\},\{\star\}) = \mathrm{cst}\cdot T_j^{\mathcal{H}_{[\![B]\!]}(\underline{\mathbf{x_B}})} \circ \mathbf{skl}_{[\![t]\!]}(\mathbf{x_A},\mathbf{x_B}) \circ G_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})}$$

where G and T are the morphisms of Proposition 2.1.18 and $\mathrm{cst} \in \mathbb{R}_{>0}$. We can deduce this property from the following lemma.

Lemma 10.3.3 (Semantics of Tests and Generators)**.** *For A a type and $(\underline{\mathbf{x_A}}|i) \in \mathcal{E}^e([\![A]\!])$:*

$$\underline{\mathbf{x_A}} \neq \underline{\mathbf{y_A}} \implies \mathbf{skl}\left[\!\left[\Uparrow^A_{\underline{\mathbf{x_A}}|i}\right]\!\right](\{\star\},\mathbf{y_A}) = 0 \text{ and } \mathbf{skl}\left[\!\left[\Downarrow^A_{\underline{\mathbf{x_A}}|i}\ x\right]\!\right](\mathbf{y_A},\{\star\}) = 0$$

And for some $\alpha_{(\underline{\mathbf{x_A}}|i)}, \beta_{(\underline{\mathbf{x_A}}|i)} \in \mathbb{N}_{>0}$ we have:

$$\begin{aligned}
\mathbf{skl}\left[\!\left[\Uparrow^A_{\underline{\mathbf{x_A}}|i}\right]\!\right](\{\star\},\mathbf{x_A}) &= \alpha_{(\underline{\mathbf{x_A}}|i)} G_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})} &&\in \mathbf{CPM}(\mathbf{1}, \mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})) \\
\mathbf{skl}\left[\!\left[\Downarrow^A_{\underline{\mathbf{x_A}}|i}\ x\right]\!\right](\mathbf{x_A},\{\star\}) &= \beta_{(\underline{\mathbf{x_A}}|i)} T_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})} &&\in \mathbf{CPM}(\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}}), \mathbf{1})
\end{aligned}$$

Proof. The terms for $A = \mathbf{qubit}$ have been created such that it holds for them. We then simply proceed by induction on the type, using the fact that every formal variable appears at most once in the test/generator term so the skeleton of an interactive composition is the composition of the skeletons (Lemma 10.2.13). For the function case, we use the compact closure of **CPM**. The only difficult case is the ! for the generator term, as it introduces new formal parameters. For $!F = !(A \multimap B)$ a type and $(\underline{\mathbf{x_F}}|i) = (\underline{\mathbf{x_1}}|i_1) \otimes \ldots \otimes (\underline{\mathbf{x_n}}|i_n) \in \mathcal{E}^e([\![!F]\!])$, we have:

$$\Uparrow^{!F}_{\underline{\mathbf{x_F}}|i} := \lambda v. \sum_{k=1}^{n} \frac{X_k}{n} \cdot \left(\Uparrow^F_{\underline{\mathbf{x_k}}|i_k}\ v\right)$$

$$\left[\!\left[\Uparrow^{!F}_{\underline{\mathbf{x_F}}|i}\right]\!\right] := !\left(\Lambda\left(\coprod_{k=1}^{n} \frac{X_k}{n} \Lambda^{-1}\left(\left[\!\left[\Uparrow^F_{\underline{\mathbf{x_k}}|i_k}\right]\!\right]\right)\right)\right) \circ \mathrm{m}_{\mathbf{1}}$$

$$\Lambda(f) := ([\![A]\!] \multimap f) \odot \mathrm{fun}_{[\![B]\!],[\![A]\!]} \qquad \Lambda^{-1}(g) := \mathrm{eval}_{[\![B]\!],[\![A]\!]} \odot (g \otimes [\![A]\!])$$

We recall that $\mathrm{m}_{\mathbf{1}} : \mathbf{1} \nrightarrow !\mathbf{1}$. Since $\boxplus$ is linear up to $\equiv$, we have

$$\left[\!\!\left[\Uparrow^{!F}_{\underline{\mathbf{x_F}}|i} \right]\!\!\right] \equiv ! \left(\Lambda \left(\Lambda^{-1} \left(\coprod_{k=1}^{n} \frac{X_k}{n} \left[\!\!\left[\Uparrow^{F}_{\underline{\mathbf{x_k}}|i_k} \right]\!\!\right] \right) \right) \right) \circ \mathrm{m}_{\mathbf{1}} \cong ! \left(\coprod_{k=1}^{n} \frac{X_k}{n} \left[\!\!\left[\Uparrow^{F}_{\underline{\mathbf{x_k}}|i_k} \right]\!\!\right] \right) \circ \mathrm{m}_{\mathbf{1}}$$

We write σ for $\left[\!\!\left[\Uparrow^{!F}_{\underline{\mathbf{x_F}}|i} \right]\!\!\right]$ and σ_k for $\left[\!\!\left[\Uparrow^{F}_{\underline{\mathbf{x_k}}|i_k} \right]\!\!\right]$. As usual, we write S and S_k for their respective $\sim$-esps. We note that we have

$$S = S_1 \otimes^p \ldots \otimes^p S_n$$

For $\otimes^p$ the parallel tensor as defined in Proposition 5.5.4. If we take a configuration $y \in \mathcal{C}(S)$, it corresponds to a set of configurations $y_j \in \mathcal{C}(S_{k_j})$ for $1 \leq j \leq m$. More precisely,

$$y = y_1 \otimes^p \ldots \otimes^p y_m$$

For $\mathbf{skl}_\sigma(y)$ to be non-zero, we need to have exactly one configuration per S_k, in other words we need $j \mapsto k_j$ to be a bijection between $\{1, \ldots, m\}$ and $\{1, \ldots, n\}$. Moreover, since permuting the different y_j results in a negative symmetry in the game, and the $\mathcal{Q}(-,-)$ only considers configurations up to positive symmetry, not only do we need to have exactly one configuration per S_k, but they have to be in the exact order $S_1, \ldots, S_n$, *i.e.*, $j \mapsto k_j$ is the identity. This means we have

$$\begin{array}{rcl} \mathbf{skl}_{\left[\!\!\left[\Uparrow^{!F}_{\underline{\mathbf{x_F}}|i} \right]\!\!\right]}(\{\star\}, \mathbf{x_F}) & = & \bigotimes_{k=1}^{n} \frac{1}{n} \mathbf{skl}_{\left[\!\!\left[\Uparrow^{F}_{\underline{\mathbf{x_k}}|i_k} \right]\!\!\right]}(\{\star\}, \mathbf{x_k}) \\ & = & \frac{1}{n^n} \prod_{i=1}^{n} \alpha_{(\underline{\mathbf{x_k}}|i_k)} \bigotimes_{k=1}^{n} G_{i_k}^{\mathcal{H}_{[\![F]\!]}(\underline{\mathbf{x_k}})} \\ & = & \frac{1}{n^n} \prod_{i=1}^{n} \alpha_{(\underline{\mathbf{x_k}}|i_k)} G_i^{\mathcal{H}_{[\![!F]\!]}(\underline{\mathbf{x_F}})} \end{array}$$

So for $\alpha_{(\underline{\mathbf{x_F}}|i)} = \frac{1}{n^n} \prod_{i=1}^{n} \alpha_{(\underline{\mathbf{x_k}}|i_k)}$ we obtain the expected result. □

As a corollary of this lemma, we obtain the following finiteness result:

Corollary 10.3.4. *For $\sigma \in \sim$-$\mathbf{QA}[-](\llbracket A \rrbracket, \llbracket B \rrbracket)$ and $\mathbf{x_A} \in \mathcal{E}(\llbracket A \rrbracket), \mathbf{x_B} \in \mathcal{E}(\llbracket B \rrbracket)$, we have $\mathcal{Q}_\sigma(\mathbf{x_A}, \mathbf{x_B}) \in \mathbf{CPM}[-](\mathcal{H}_{\llbracket A \rrbracket}(\underline{\mathbf{x_A}}), \mathcal{H}_{\llbracket B \rrbracket}(\underline{\mathbf{x_B}}))$, i.e., is a $[0,1]$-convergent formal power series of* **CPM** *operators.*

Proof. We start by the non-parametric case $\sigma \in \sim$-$\mathbf{QA}(\llbracket A \rrbracket, \llbracket B \rrbracket)$. We write $I = \{i \mid (\underline{\mathbf{x_A}}|i) \in \mathcal{E}^e(\llbracket A \rrbracket)\}$ and $J = \{j \mid (\underline{\mathbf{x_B}}|j) \in \mathcal{E}^e(\llbracket B \rrbracket)\}$. We consider

$$\tau := \left(\coprod_{j \in J} \frac{1}{|J|} \left[\!\!\left[\Downarrow^{B}_{\underline{\mathbf{x_B}}|j} \quad \mathrm{var} \right]\!\!\right]^{\mathrm{var}:B \vdash_{\mathbb{T}} \mathbf{1}} \right) \odot \sigma \odot \left(\coprod_{i \in I} \frac{1}{|I|} \left[\!\!\left[\Uparrow^{A}_{\underline{\mathbf{x_A}}|i} \right]\!\!\right]^{\vdash_{\mathbb{T}} A} \right)$$

Using Corollary 9.4.18, we obtain

$$\begin{array}{rcl}\mathcal{Q}_\tau(\{\star\},\{\star\}) & = & \left(\sum_{j\in J}\frac{1}{|J|}\sum_{\mathbf{y_B}\in\mathcal{E}(\llbracket B\rrbracket)}\mathcal{Q}_{\left\llbracket\Downarrow^B_{\underline{\mathbf{x_B}}|j}\ x\right\rrbracket}(\mathbf{y_B},\{\star\})\right)\\ & \odot & \mathcal{Q}_\sigma(\underline{\mathbf{x_A}},\underline{\mathbf{x_B}})\\ & \odot & \left(\sum_{i\in I}\frac{1}{|I|}\sum_{\mathbf{y_A}\in\mathcal{E}(\llbracket A\rrbracket)}\mathcal{Q}_{\left\llbracket\Uparrow^A_{\underline{\mathbf{x_A}}|i}\right\rrbracket}(\{\star\},\mathbf{y_A})\right)\end{array}$$

Using Lemma 10.3.3, since σ has no parameters, and that the parameters of $\left\llbracket\Uparrow^A_{\underline{\mathbf{x_A}}|i}\right\rrbracket$ and $\left\llbracket\Downarrow^B_{\underline{\mathbf{x_B}}|j}\right\rrbracket$ are disjoint we have

$$\begin{array}{rcl}\mathbf{skl}_\tau(\{\star\},\{\star\}) & = & \left(\sum_{j\in J}\frac{1}{|J|}\mathbf{skl}_{\left\llbracket\Downarrow^B_{\underline{\mathbf{x_B}}|j}\ x\right\rrbracket}(\mathbf{x_B},\{\star\})\right)\\ & \odot & \mathcal{Q}_\sigma(\underline{\mathbf{x_A}},\underline{\mathbf{x_B}})\\ & \odot & \left(\sum_{i\in I}\frac{1}{|I|}\mathbf{skl}_{\left\llbracket\Uparrow^A_{\underline{\mathbf{x_A}}|i}\right\rrbracket}(\{\star\},\mathbf{x_A})\right)\\ & = & \left(\sum_{j\in J}\frac{\beta_{(\underline{\mathbf{x_B}}|i)}}{|J|}T_j^{\mathcal{H}_{\llbracket B\rrbracket}(\underline{\mathbf{x_B}})}\right)\\ & \odot & \mathcal{Q}_\sigma(\underline{\mathbf{x_A}},\underline{\mathbf{x_B}})\\ & \odot & \left(\sum_{i\in I}\frac{\alpha_{(\underline{\mathbf{x_A}}|i)}}{|I|}G_i^{\mathcal{H}_{\llbracket A\rrbracket}(\underline{\mathbf{x_A}})}\right)\end{array}$$

If we instantiate all the parameters of τ by 1, we obtain a quantum $\sim$-strategy $\tau[1,\ldots,1]$ from $\mathbf{1}$ to $\mathbf{1}$. Using Lemma 9.4.13, we have

$$\mathcal{Q}_{\tau[1,\ldots,1]}(\{\star\},\{\star\})\sqsubseteq\mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$$

Since we consider formal power series over **CPM** operators, all the terms of the formal power series are positive, so

$$\mathbf{skl}_\tau(\{\star\},\{\star\})\sqsubseteq\mathcal{Q}_{\tau[1,\ldots,1]}(\{\star\},\{\star\})\sqsubseteq\mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$$

We also note that

$$\left(\sum_{j\in J}T_j^{\mathcal{H}_{\llbracket B\rrbracket}(\underline{\mathbf{x_B}})}\right)\sqsupseteq\mathbf{Tr}_{\mathcal{H}_{\llbracket B\rrbracket}(\underline{\mathbf{x_B}})}\qquad\left(\sum_{i\in I}G_i^{\mathcal{H}_{\llbracket A\rrbracket}(\underline{\mathbf{x_A}})}\right)\sqsupseteq\mathbf{1}_{\mathcal{H}_{\llbracket A\rrbracket}(\underline{\mathbf{x_A}})}$$

So we obtain, for some $\alpha, \beta \in \mathbb{R}_{>0}$

$$\beta \mathbf{Tr}_{\mathcal{H}_{[\![B]\!]}(\underline{\mathbf{x_B}})} \circ \mathcal{Q}_\sigma(\underline{\mathbf{x_A}}, \underline{\mathbf{x_B}}) \circ \alpha \mathbf{1}_{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})} \sqsubseteq \mathbf{id}_{\mathbf{1}}^{\mathbf{CPM}}$$

It follows that $\mathcal{Q}_\sigma(\underline{\mathbf{x_A}}, \underline{\mathbf{x_B}})$ is finitary. If we now look at the parametrised case $\sigma \in \sim\text{-}\mathbf{QA}[-](\llbracket A \rrbracket, \llbracket B \rrbracket)$, we obtain that all its instantiations with parameters in $[0,1]$ are non-parametric strategies, so all the $\mathcal{Q}(-,-)$ of those instantiations are finitary, so the formal power series $\mathcal{Q}_\sigma(-,-)$ is necessarily $[0,1]$-convergent with parameters in **CPM**. □

10.3.5 Full Abstraction for LQ$\Lambda_!$

We then have all the tools to prove the reverse implication of the full abstraction, and conclude with the full abstraction theorem for LQ$\Lambda_!$.

Definition 10.3.5. *For $\Gamma \vdash^{\text{param}} t : A$ a parametrised term, we say that the term $\Gamma \vdash t' : A$ is an instance of t if there exists some $p_1, \dots, p_n \in [0,1]$ such that by replacing the formal parameters $X_1, \dots, X_n$ of t by $p_1, \dots, p_n$ gives t'.*

Lemma 10.3.6 (Characterisation by Tests and Generators)**.** *We define the set of observers $O_{x:A \vdash_{\mathbb{L}} B}$ as*

$$O_{x:A \vdash_{\mathbb{L}} B} = \left\{ \mathbf{let}\ v^A = \mathrm{Gen}\ \mathbf{in}\ \mathrm{Test}\ _ \;\middle|\; \begin{array}{ll} (\underline{\mathbf{x_A}}|i) \in \mathcal{E}^e(\llbracket A \rrbracket), & (\underline{\mathbf{x_B}}|j) \in \mathcal{E}^e(\llbracket B \rrbracket), \\ \vdash_{\mathbb{L}} \mathrm{Gen} : A & \textit{instance of } \Uparrow^A_{\underline{\mathbf{x_A}}|i}, \\ \vdash_{\mathbb{L}} \mathrm{Test} : A \multimap \mathbf{1} & \textit{instance of } \Downarrow^B_{\underline{\mathbf{x_B}}|j} \end{array} \right\}$$

For every pair of terms $x : A \vdash_{\mathbb{L}} t : B$ and $x : A \vdash_{\mathbb{L}} s : B$, we have:

$$\begin{array}{c} \forall \mathcal{O}[_] \in O_{x:A \vdash_{\mathbb{L}} B}, \mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow) \\ \Longrightarrow \\ \llbracket t \rrbracket \equiv \llbracket s \rrbracket \end{array}$$

Proof. Using Lemma 10.3.3, we have immediately that for any parametrised observer $\mathcal{P}[_] = \mathbf{let}\ v^A = \Uparrow^A_{\underline{\mathbf{x_A}}|i}\ \mathbf{in}\ \Downarrow^B_{\underline{\mathbf{x_B}}|j}\ _$, we have

$$\mathbf{skl}_{\llbracket \mathcal{O}[t] \rrbracket}(\{\star\}, \{\star\}) = \alpha_{(\underline{\mathbf{x_B}}|j)} \beta_{(\underline{\mathbf{x_A}}|i)} T_j^{\mathcal{H}_{[\![B]\!]}(\underline{\mathbf{x_B}})} \circ \mathcal{Q}_{\llbracket t \rrbracket}(\mathbf{x_A}, \mathbf{x_B}) \circ G_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})}$$

If for all observers we have $\mathbb{P}(\mathcal{O}[t] \Downarrow) = \mathbb{P}(\mathcal{O}[s] \Downarrow)$, using soundness and adequacy (Theorem 9.5.6) we have for all $(\underline{\mathbf{x_A}}|i) \in \mathcal{E}^e(\llbracket A \rrbracket)$ and all $(\underline{\mathbf{x_B}}|j) \in \mathcal{E}^e(\llbracket B \rrbracket)$:

$$\llbracket \mathcal{O}[t] \rrbracket \equiv \llbracket \mathcal{O}[s] \rrbracket$$

So for every observer $\mathcal{O}[_]$, we have

$$\mathcal{Q}_{[\![\mathcal{O}[t]]\!]}(\{\star\},\{\star\}) = \mathcal{Q}_{[\![\mathcal{O}[s]]\!]}(\{\star\},\{\star\})$$

Using Theorem 10.1.2, we obtain that that for the parametrised observer $\mathcal{P}[_] = \mathbf{let}\ v^A = \Uparrow^A_{\underline{\mathbf{x_A}}|i}\ \mathbf{in}\ \Downarrow^B_{\underline{\mathbf{x_B}}|j}\ _$, we have

$$\mathcal{Q}_{[\![\mathcal{P}[t]]\!]}(\{\star\},\{\star\}) = \mathcal{Q}_{[\![\mathcal{P}[s]]\!]}(\{\star\},\{\star\})$$

In particular, their skeletons are equal, so:

$$\begin{array}{rcccccc} & \alpha_{(\underline{\mathbf{x_B}}|j)}\beta_{(\underline{\mathbf{x_A}}|i)} & T_j^{\mathcal{H}_{[\![B]\!]}(\underline{\mathbf{x_B}})} & \circ & \mathcal{Q}_{[\![t]\!]}(\mathbf{x_A},\mathbf{x_B}) & \circ & G_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})} \\ = & \alpha_{(\underline{\mathbf{x_B}}|j)}\beta_{(\underline{\mathbf{x_A}}|i)} & T_j^{\mathcal{H}_{[\![B]\!]}(\underline{\mathbf{x_B}})} & \circ & \mathcal{Q}_{[\![s]\!]}(\mathbf{x_A},\mathbf{x_B}) & \circ & G_i^{\mathcal{H}_{[\![A]\!]}(\underline{\mathbf{x_A}})} \end{array}$$

Using Corollary 10.3.4, we know that $\mathbf{skl}_{[\![t]\!]}$ and $\mathbf{skl}_{[\![s]\!]}$ are in **CPM**, so using Proposition 2.1.18 we deduce that for all $\underline{\mathbf{x_A}} \in \mathcal{E}([\![A]\!])$ and $\underline{\mathbf{x_B}} \in \mathcal{E}([\![x_B]\!])$ we have:

$$\mathcal{Q}_{[\![t]\!]}(\underline{\mathbf{x_A}},\underline{\mathbf{x_B}}) = \mathcal{Q}_{[\![s]\!]}(\underline{\mathbf{x_A}},\underline{\mathbf{x_B}}) \qquad \square$$

Theorem 10.3.7 (≡-Full Abstraction)**.** *For every term $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma \vdash_{\mathbb{L}} s : A$, we have*

$$[\![t]\!] \equiv [\![s]\!] \iff t =_{\text{obs}} s$$

Proof. The direct implication is exactly Corollary 9.5.7. We now assume $t =_{\text{obs}} s$. We write $P = \bigotimes_{(v_i:A_i)\in\Gamma} A_i$. We consider

$$t' = \mathbf{let}\ \text{var} = \bigotimes_{(\text{var}_i:A_i)\in\Gamma} \text{var}_i\ \mathbf{in}\ s \text{ and } s' = \mathbf{let}\ \text{var} = \bigotimes_{(\text{var}_i:A_i)\in\Gamma} \text{var}_i\ \mathbf{in}\ s$$

It follows that $t' =_{\text{obs}} s'$. In particular, for every $\mathcal{O}[_] \in O_{y:P|_{\mathbb{L}}\,A}$, we have $\mathbb{P}(\mathcal{O}[t'] \Downarrow) = \mathbb{P}(\mathcal{O}[s'] \Downarrow)$. It follows from the previous lemma that $[\![t']\!] \equiv [\![s']\!]$. From the definition of the semantics, it follows immediately that $[\![t]\!] \equiv [\![s]\!]$. $\square$

10.3.6 Affine Case

In the affine case, we simply take

$$\Downarrow^{A\multimap B}_{\{\lambda^+\}} := \lambda f.()$$
$$\Uparrow^{A\multimap B}_{\{\lambda^+\}} := \lambda x.\bot$$

And with the exact same reasoning we obtain a full abstraction result for AQΛ!.

Theorem 10.3.8 (≡-Full Abstraction)**.** *For every term $\Gamma \vdash_{\mathbb{A}} t : A$ and $\Gamma \vdash_{\mathbb{A}} s : A$, we have*

$$[\![t]\!] \equiv [\![s]\!] \iff t =_{\text{obs}} s$$

10.3.7 Full Abstraction of the Relational Model

We recall that we have the following factorisation result.

Since the functor from $\sim$-**QA** to $\sim$-**QARel** preserves all the structure, and is $\equiv$-faithful, we can deduce the full abstraction of the relational model on $\sim$-arenas from the full abstraction of the game model.

Theorem 10.3.9. *We have the following results of full abstraction for $LQ\Lambda_!$ and $AQ\Lambda_!$:*

- *For every term $\Gamma \vdash_{\mathbb{L}} t : A$ and $\Gamma \vdash_{\mathbb{L}} s : A$, we have*

$$[\![t]\!]_{\sim\text{-}\mathbf{QA}} \equiv [\![s]\!]_{\sim\text{-}\mathbf{QA}} \iff t =_{\mathrm{obs}} s \iff [\![t]\!]_{\sim\text{-}\mathbf{QARel}} = [\![s]\!]_{\sim\text{-}\mathbf{QARel}}$$

- *For every term $\Gamma \vdash_{\mathbb{A}} t : A$ and $\Gamma \vdash_{\mathbb{A}} s : A$, we have*

$$[\![t]\!]_{\sim\text{-}\mathbf{QA}^a} \equiv [\![s]\!]_{\sim\text{-}\mathbf{QA}^a} \iff t =_{\mathrm{obs}} s \iff [\![t]\!]_{\sim\text{-}\mathbf{QARel}^a} = [\![s]\!]_{\sim\text{-}\mathbf{QARel}^a}$$

As shown in Theorem 8.4.2, $\sim$-**QARel** and $\overline{\mathbf{CPMs}}^{\oplus}$ as defined in [PSV14] are essentially the same model, so the full abstraction of $\overline{\mathbf{CPMs}}^{\oplus}$ immediately follows.

Conclusion and Perspectives

This thesis closes the open question of finding a fully abstract model for the quantum λ-calculus. This thesis is part of a line of work about denotational models for languages having both quantitative effects and rich control flows; we expect the methods and tools built along this thesis to have a larger impact than just answering the question of full abstraction for the quantum λ-calculus. While the goal of this thesis has been completed, there are still a number of possible continuations of this work.

- Following the work in [dV19] on event structures supporting two branching structures: (1) a non-quantitative non-deterministic choice and (2) a probabilistic choice, we expect to be able to build a model for a quantum λ-calculus extended with a non-quantitative non-deterministic choice.
- As our model relies on concurrent game semantics, a natural continuation of our work would be to consider a concurrent quantum λ-calculus. Relying on the previous item and the concurrent game model for IPA [CCHW18], we hope to build a model for a quantum λ-calculus extended with concurrency, reference cells, and the non-determinism coming from concurrent access to shared reference cells.
- The quantum λ-calculus is a call-by-value language. It is unclear to us whether we could easily convert our game model into a model for a call-by-name quantum programming language.
- The quantum λ-calculus is a language in which the user directly controls quantum data. This is not the case of every quantum programming language, with for example Quipper [GLR$^+$13] in which the user builds and controls quantum circuits, which are then executed. To our knowledge, there is not yet any game semantics model for a language like this.
- The question of finding a model with good definability properties for the quantum λ-calculus is still open, but we expect that a refinement of our model with a notion of innocence similar to [CCW15a, CCW17, CCPW18] would work.
- The proof used to solve the full-abstraction problem for the quantum λ-calculus bypassed the traditional route to proving a model for a language fully abstract, *i.e.*, via

definability, mentioned above. In trying to achieve definability we have worked on a definition of quantum strategies with quantum valuations on extensions $x \subseteq y$ rather than on configurations x, to make them functorial with respect to the inclusion order on configurations. From this a notion of "quantum Petri net" has arisen, which we plan to investigate.

- For the most part of this thesis, the category **CPM** of quantum computation could be replaced by an arbitrary compact closed category, giving rise to a model for a λ-calculus parametrised by a CpCC similar to [TAO18]. Further work would be required to determine if we can model a λ-calculus parametrised by a category with weaker properties than a CpCC, *e.g.*, an SMC. This would allow us to cover a wide range of effects and language features, *e.g.*, continuous probabilities.

Part IV

Appendix

Appendix A

Miscellaneous Lemmas

A.1 Interaction and Interactive Composition

In this subsection, we list five lemmas about the interaction and interactive composition of strategies. In all those lemmas, we consider $\sigma, \sigma' : A \nrightarrow B$ and $\tau, \tau' : B \nrightarrow C$, writing S, S', T, T' for their respective esps.

- The Lemma A.1.1 formalises the concept of projecting events from $T \odot S$ to T or S.
- The Lemma A.1.2 highlights the fact that the causalities of $T \circledast S$ are a superposition of the causalities of T and the ones of S.
- The Lemma A.1.3 highlights the fact that every event at the "middle" of an interaction $T \circledast S$ is covered by a positive event.
- The Lemma A.1.4 notes that negative extensions on $T \odot S$ correspond to negative extensions on T and S. We will rely on it for the proof of preservation of obliviousness in Proposition 5.3.11.
- The Lemma A.1.5 allows us to tame the isomorphism between strategies by noting that, within some reasonable conditions, we can prove the minimal events of two strategies are the exactly the same.

Lemma A.1.1. *We take $y \odot x \in \mathcal{C}(T \odot S)$. There are two polarity-preserving order-reflecting bijection:*

$$\begin{array}{rccc} \pi_S : & \{e \in y \odot x \mid (\tau \odot \sigma)(e) \in_{\|} |A^{\perp}|\} & \to & \{e \in x \mid \sigma(e) \in_{\|} |A^{\perp}|\} \\ \pi_T : & \{e \in y \odot x \mid (\tau \odot \sigma)(e) \in_{\|} |C|\} & \to & \{e \in y \mid \tau(e) \in_{\|} |C|\} \end{array}$$

They extend to $y \circledast x = [y \odot x]_{\circledast}$ into two order-reflecting bijections:

$$\begin{array}{rccc} \pi_S : & \{e \in y \circledast x \mid (\tau \circledast \sigma)(e) \notin_{\|} |C|\} & \to & x \\ \pi_T : & \{e \in y \circledast x \mid (\tau \circledast \sigma)(e) \notin_{\|} |A^{\perp}|\} & \to & y \end{array}$$

The bijections π_S and π_T are simply obtained from the maps of event structures $\pi_{S\|C}$ and $\pi_{A\|T}$ of the diagram defining $\tau \odot \sigma$.

Lemma A.1.2. *If $e \rightarrowtriangle_{T\circledast S} e'$, then either $\pi_{S\|C}(e) \rightarrowtriangle_{S\|C} \pi_{S\|C}(e')$ or $\pi_{A\|T}(e) \rightarrowtriangle_{A\|T} \pi_{A\|T}(e')$.*

We refer to Lemma 2.10 in [CCRW17] for a proof.

Lemma A.1.3. *For every $y \odot x \in \mathcal{C}(T \odot S)$, for every $e_S \in x$ such that $\sigma(e_S) \in_{\|} |B|$, there exists $s^+ \in x$ such that $e_S \leq_S s$. Symmetrically, for every $e_T \in y$ such that $\tau(e_T) \in_{\|} |B|$, there exists $t^+ \in y$ such that $e_T \leq_T t$.*

Proof. Since configurations are finite, it is equivalent to prove that for every $e_S \in x$ maximal and negative, we do not have $\sigma(e_S) \in_{\|} |B|$. We proceed by contradiction and take $e_S \in x$ maximal and negative with $\sigma(e_S) \in_{\|} |B|$.

Using Lemma A.1.1, we obtain $e \in (y \circledast x)\backslash(y \odot x)$ such that $\pi_S(e) = e_S$. Since $y \circledast x = [y \odot x]_{\circledast}$, it follows we have $e' \in y \odot x$ such that $e <_{T\circledast S} e'$. We take e' minimal, and since e_S was maximal we obtain $e \rightarrowtriangle_{T\circledast S} e'$. Using Lemma A.1.2, we know that either $\pi_{S\|C}(e) \rightarrowtriangle_{S\|C} \pi_{S\|C}(e')$ or $\pi_{A\|T}(e) \rightarrowtriangle_{A\|T} \pi_{A\|T}(e')$.

- Assume $(\tau \circledast \sigma)(e') \in_{\|} |A|$. We can quickly eliminate the second possibility as the two events are not in the same part of $A \parallel T$. We obtain $\pi_S(e) \rightarrowtriangle_S \pi_S(e')$, which contradicts maximality of e_S.
- Assume $(\tau \circledast \sigma)(e') \in_{\|} |C|$. We can quickly eliminate the first possibility as the two events are not in the same part of $S \parallel C$. We obtain $\pi_T(e) \rightarrowtriangle_T \pi_T(e')$. Since σ plays on B while τ plays on $B^{\perp}$, e_S and $\pi_T(e)$ have opposite polarities, so $\pi_T(e)$ is positive. Using courtesy of τ, it follows that $\tau(\pi_T(e)) \rightarrowtriangle_{B^{\perp}\|C} \tau(\pi_T(e'))$, which is impossible as one is in $B^{\perp}$ while the other is in C.

This proves that for every $e_S \in x$ maximal and negative, we do not have $\sigma(e_S) \in_{\|} |B|$. We proceed symmetrically for proving that for every $e_T \in y$ maximal and negative, we do not have $\tau(e_T) \in_{\|} |B|$. □

Lemma A.1.4. *If $y' \odot x' \subseteq^{-} y \odot x \in \mathcal{C}(T \odot S)$ then $y' \subseteq^{-} y \in \mathcal{C}(T)$ and $x' \subseteq^{-} x \in \mathcal{C}(S)$. Moreover, for every $s \in x\backslash x'$ we have $\sigma(s) \in_{\|} |A^{\perp}|$, and for every $t \in y\backslash y'$ we have $\tau(t) \in_{\|} |C|$.*

Proof. We consider $y' \odot x' \subseteq^{-} y \odot x \in \mathcal{C}(T \odot S)$. Since the down-closure is a monotone operation, we have $y' \circledast x' = [y' \odot x']_{\circledast} \subseteq [y \odot x]_{\circledast} = y \circledast x$.

We take an event e maximal such that $e \in (y \circledast x)\backslash(y \odot x)$. We will show that necessarily $e \in (y' \circledast x')$.

Since $(\tau \circledast \sigma)(e) \in_{\|} |B|$, using Lemmas A.1.1 and A.1.3 we find $\pi_S(e) \leq_S s^+ \in x$ and $\pi_T(e) \leq_T t^+ \in y$. Since $(\tau \circledast \sigma)(e) \in_{\|} |B|$, we know that $\pi_T(e)$ and $\pi_S(e)$ have opposite

polarities, so we obtain either $\pi_S(e) <_S s^+$ or $\pi_T(e) <_S t^+$. Using Lemma A.1.1 we obtain $e <_{T \circledast S} e' \in y \circledast x$ with either $\pi_S(e') = s^+$ or $\pi_T(e') = t^+$. Since we assumed maximality of e, this means that $e' \in y \odot x$ and that e' has positive polarity, so $e' \in y' \odot x'$, and then $e \in [y' \odot x']_{\circledast} = y' \circledast x'$.

This means that none of the events of the extension $y' \circledast x' \subseteq y \circledast x$ are sent to B by $\tau \circledast \sigma$. Using Lemma A.1.1, this means that all the events of the extensions $x' \subseteq x$ and $y' \subseteq y$ correspond to events of the extension $y' \odot x' \subseteq^- y \odot x$, and have negative polarity. □

Lemma A.1.5. *If the minimal events of A,B,C are positive, and the minimal events of S, T, T' are negative, then $T \odot S$ and $T' \odot S$ have exactly the same minimal events (**not** up to renaming).*

Proof. The minimal negative events of $A^\perp \parallel C$ are all in A, and minimal events of $T \odot S$ or $T' \odot S$ must be sent to minimal negative events of $A^\perp \parallel C$. It follows that they are on the S side. While we refer for [CCRW17] for the exact construction of $\odot$, the fact is that minimal events of $T \odot S$ are pairs $(x, \varnothing)$ with x any singleton configuration of S, meaning that we obtain the same minimal events for $T \odot S$ and $T' \odot S$. □

A.2 $\oplus$-Covered Configurations

We say that a configuration x is $\oplus$-covered if all its maximal events are positive. We write $\mathcal{C}^+(E)$ for the set of $\oplus$-covered configurations of E. We list four lemmas about the properties of those configurations. In all of those lemmas, we assume A, B, C to be three games and $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ to be two strategies, and write S and T being their respective esps.

- The Lemma A.2.1 proves the stability of the concept of $\oplus$-coveredness by composition, which will be central for payoff games in Section 5.4.
- The Lemma A.2.2 shows that the minimality condition of minimal matching compatible is superfluous when considering only $\oplus$-covered configurations.
- The Lemma A.2.3 shows the interaction between the copy-cat strategy and the concept of $\oplus$-coveredness.
- The Lemma A.2.4 shows that strategies are characterised by their set of $\oplus$-covered configurations.

Lemma A.2.1. *The configuration $y \odot x \in \mathcal{C}(T \odot S)$ is $\oplus$-covered if and only if both $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(T)$ are $\oplus$-covered.*

Proof. If $y \odot x$ is not $\oplus$-covered, then we have $y' \odot x' \subset^- y \odot x$. Using Lemma A.1.4, we obtain that $y' \subseteq^- y$ and $x' \subseteq^- x$. Both cannot be equalities, and if $y' \subset^- y$ then y is not $\oplus$-covered, while if $x' \subset^- x$ then x is not $\oplus$-covered. We assume that $y \odot x$ is $\oplus$-covered. We take an event $s \in x$ maximal. If $\sigma(s) \in_{\|} |B|$ then using Lemma A.1.3 we know that s is positive. If $\sigma(s) \in_{\|} |B|$, using Lemma A.1.1 we obtain $e \in y \odot x$ such that $\pi_S(e) = s$. If e is maximal in $y \odot x$, then it is positive by hypothesis, so s is positive. If e is not maximal in $y \odot x$, then it is not maximal in $y \circledast x$ either. We have $e \rightarrowtriangle_{T \circledast S} e' \in y \circledast x$. Using Lemma A.1.2, we know that either $\pi_{S\|C}(e) \rightarrowtriangle_{S\|C} \pi_{S\|C}(e')$ or $\pi_{A\|T}(e) \rightarrowtriangle_{A\|T} \pi_{A\|T}(e')$.

- In the first case, this means that both e and e' project to the S side of $S \parallel C$. This implies $s \rightarrowtriangle_S \pi_S(e') \in x$, which contradicts maximality of s.
- In the second case, this means that both e and e' project to the A side of $A \parallel T$. This implies that the $\rightarrowtriangle$ causality comes from the game A, which leads to $s <_S \pi_S(e') \in x$, which also contradicts maximality of s.

So every maximal event of x is positive. We proceed symmetrically for y. □

Lemma A.2.2. *If $y \circledast x \in \mathcal{C}(T \circledast S)$ with y and x $\oplus$-covered, then (y, x) is a minimal matching pair of configurations,i.e., $y \odot x$ is defined.*

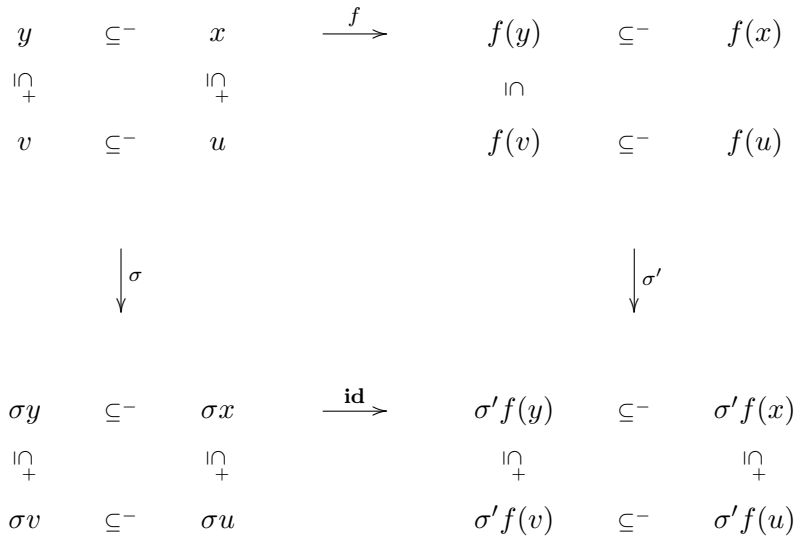


Figure A.1: Diagram for the proof of Lemma A.2.4.

Proof. If $y \circledast x$ is not minimal, then that means we can "remove" at least one event of $y \circledast x$ which is sent to B. In other words, there is a maximal event of $y \circledast x$ which is sent to B. This maximal event is either sent to a maximal negative event of y, or a maximal negative event of x. This contradicts $\oplus$-coveredness of x and y. □

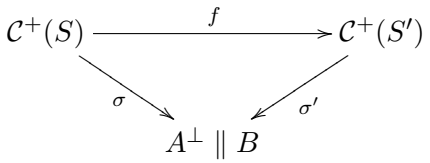
Lemma A.2.3. *For A a game and $z = x \parallel y \in \mathcal{C}(\mathbb{C}_A)$, z is $\oplus$-covered if and only if $x = y$.*

Proof. By definition of copy-cat, we have $x \sqsupseteq y$ with $\sqsubseteq$ the Scott order. We also have that an event $e \in z$ is maximal in z if and only if

- Either $e = (0, a)$, a is maximal in x , and e is positive (so a negative).
- Or $e = (1, a)$, a is maximal in y, and e is positive (so a positive).
- Or $e = (0, a)$, a is maximal in x, e is negative (so a positive), and $a \notin y$.
- Or $e = (1, a)$, a is maximal in y, e is negative (so a negative), and $a \notin x$.

So z is $\oplus$-covered if and only if we are never in the second case, so every event of x positive in A (so negative in $A^\perp$) is in y, and every event of y negative in A is in x. Combining with $x \sqsubseteq y$, which says that every negative event of x is in y, and every positive event of y is in x, we obtain $x = y$. □

Lemma A.2.4. *For $\sigma, \sigma' : A \nrightarrow B$, if there exists a bijection $f : \mathcal{C}^+(S) \to \mathcal{C}^+(S')$ which is an order-isomorphism for the inclusion, i.e., $x \subseteq y \iff f(x) \subseteq f(y)$, and commutes with the strategies, i.e., the following diagram commutes*



then $\sigma \cong \sigma'$.

Proof. We start by using receptivity to extend f as a bijection between $\mathcal{C}(S)$ and $\mathcal{C}(S')$:

- For $x \in \mathcal{C}(S)$, there exists a unique $y \in \mathcal{C}^+(S)$ such that $y \subseteq^- x$. Using receptivity of σ', there exists a unique $f(y) \subseteq^- x' \in \mathcal{C}(S')$ such that $\sigma'\, x' = \sigma\, x$. We take $f(x) := x'$.
- Conversely, for $x' \in \mathcal{C}(S')$, there exists a unique $f(y) \in \mathcal{C}^+(S')$ such that $f(y) \subseteq^- x'$. Using receptivity of σ, there exists a unique $y \subseteq^- x \in \mathcal{C}(S')$ such that $\sigma'\, x' = \sigma\, x$. By uniqueness, we have $f(x) = x'$.

However, it might not be an order-isomorphism any more. We consider $x, u \in \mathcal{C}(S)$ with $x \subseteq u$. There exists a unique $x \supseteq^- y \in \mathcal{C}^+(S)$ and a unique $u \supseteq^- v \in \mathcal{C}^+(S)$. If $x \subseteq^- u$, then $y = v$ and the uniqueness part of the receptivity ensures that $f(x) \subseteq^- f(u)$. So f preserves negative extensions. We now assume $x \subseteq^+ u$. By construction, we have the diagram described in Fig. A.1. We recall that because of Lemma 4.4.11, S' must be race-free, so $f(v) \cup f(x) \in \mathcal{C}(S')$. Using uniqueness of receptivity, it follows that $f(u) = f(v) \cup f(x)$, hence $f(x) \subseteq^+ f(u)$. So f preserves positive extensions. This means that f preserves the order. As f is a bijection, we can make the reverse reasoning and obtain that f reflects the order. So f is an order-isomorphism for the inclusion. This means it is union and intersection preserving, so induces an isomorphism between S and S' per Proposition 4.2.6. This isomorphism commutes with the maps σ and σ', it is an isomorphism of strategies. □

A.3 Test Strategies

The Lemma A.3.2 shows the properties of a test strategy that targets a specific $\oplus$-covered configuration of a given strategy. We write $\mathbf{1}$ for the esp with a single event $\star$ which is positive. We consider A a game. For $x_A \in \mathcal{C}(A)$, we write $\overline{x_A}^{\oplus}$ for its positive saturation, which is the game obtained when taking the unique substructure of A containing all the events of x_A plus all the positive events accessible from x_A. While the set $|\overline{x_A}^{\oplus}|$ of all the events of $\overline{x_A}^{\oplus}$ is a down-closed set of events of A, it is usually not a configuration of A.

We recall that Proposition 4.2.3 allows us to characterise an event structure through its set of configurations.

Definition A.3.1. *For $x_A \in \mathcal{C}(A)$, we define the strategy $\mathrm{test}_A(x_A) : A \nrightarrow \mathbf{1}$ as the identity-on-events map $\mathrm{test}_A(x_A) : T_A(x_A) \to A^{\perp} \parallel \mathbf{1}$, where $T_A(x_A)$ has the same events as $(\overline{x_A}^{\oplus})^{\perp} \parallel \mathbf{1}$, the same polarity, and the following configurations:*

$$y \in \mathcal{C}(T_A(x)) : \begin{cases} y = \{0\} \times z & \text{whenever } z \in \mathcal{C}(\overline{x_A}^{\oplus}) \\ y = \{0\} \times z \cup \{(1, \star)\} & \text{whenever } z \in \mathcal{C}(\overline{x_A}^{\oplus}) \text{ and } \forall e^+ \in x_A, e^+ \in z \end{cases}$$

Checking that this set of configuration indeed comes from an event structure is direct. The $\oplus$-saturation ensures that the strategy is receptive, and since we only put causal link from events positive in x (so negative in the strategy) to the positive event $\star$, the courtesy is satisfied too.

Lemma A.3.2. *We consider A a game and $\sigma : \varnothing \nrightarrow A$, and $x \in \mathcal{C}(S)$ $\oplus$-covered. We write $\sigma\, x = \varnothing \parallel x_A$. We consider the interactive composition $\tau = \mathrm{test}_A(x_A) \odot \sigma$. We have*

- *The pair $(x, x_A \parallel \{\star\})$ is minimal matching compatible.*
- *$\tau((x_A \parallel \{\star\}) \odot x) = \varnothing \parallel \{\star\}$.*

Proof. The pair is trivially matching compatible, but we need to prove the minimality. Assume (y, z) matching compatible with $z \subseteq x$ and $y \subseteq x_A \parallel \{\star\}$ with the same projection on $\varnothing$ and $\mathbf{1}$. In other words, $y = y_A \parallel \{\star\}$ with $y_A \subseteq x_A$. By definition of the test strategy, it follows that y_A contains all the positive events of x_A. We then have $z \subseteq x$ with $\sigma\, z = \varnothing || y_A$. Since y_A contains all the positive events of x_A, and x is $\oplus$-covered, it follows that $z = x$ and $x_A = y_A$. □

Appendix B

Postponed Proofs

B.1 Parallel Tensor and Semi-Bifunctoriality

B.1.1 Definitions and Objective

In Section 5.5.2, we defined the parallel tensor $\otimes^p$ on strategies and claimed that this operation satisfied the property of semi-bifunctoriality. We recall here some of the definitions and propositions: for $\sigma : A \nrightarrow B$ a negative quantum strategy, a configuration $x \in \mathcal{C}(S)$ is either

Empty if it is $\varnothing$.

Pre-Value if it is non-empty and it does not contain any $s \in x$ such that $\sigma(s) \in_{\|} \min(B)$.

Post-Value if it does contain a $s \in x$ such that $\sigma(s) \in \min(B)$. This event is called the value-event of x.

If additionally $\tau : C \nrightarrow D$ is a negative quantum strategy, we say that $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(T)$ are synchronised if they are either both empty, both pre-value, or both post-value. We recall Proposition 5.5.4

Proposition B.1.1 (Parallel Tensor of Strategies)**.** *For $\sigma : A \nrightarrow B$ and $\sigma' : A' \nrightarrow B'$ two negative quantum strategies, there exists a necessary unique (up to isomorphism) negative quantum strategy $\sigma \otimes^p \sigma'$ such that its configurations $z \in \mathcal{C}(S \otimes^p S')$ correspond to pairs written $x \otimes^p x'$ of synchronised configurations $x \in \mathcal{C}(S)$ and $x' \in \mathcal{C}(S')$, and its quantum valuation is:*

$$\mathcal{Q}_{\sigma \otimes^p \sigma'}(x \otimes x') = \mathcal{Q}_\sigma(x) \otimes \mathcal{Q}_{\sigma'}(x')$$

And we recall the semi-bifunctoriality Proposition 5.5.5 that we have yet to prove:

Proposition B.1.2 (Semi-Bifunctoriality of the Parallel Tensor)**.** *For $\sigma : A \nrightarrow B$, $\sigma' : A' \nrightarrow B'$, $\tau : B \nrightarrow C$ and $\tau' : B' \nrightarrow C'$ two negative quantum strategies, we have the semi-bifunctoriality of $\otimes^p$, i.e., up to isomorphism*

$$(\tau \otimes^p \tau') \odot (\sigma \otimes^p \sigma') \cong (\tau \odot \sigma) \otimes^p (\tau' \odot \sigma')$$

whenever τ and τ' are thunkable, or σ and σ' are thunkable.

B.1.2 Initialised Interactive Composition

As a tool in the proof of semi-bifjunctoriality, we will need a slight variation of minimal matching compatible pairs. As in Section 4.3, we consider $f : F \to A \parallel B$ and $g : G \to B \parallel C$ two maps of event structures. We have

$$\begin{array}{ccc} F \parallel C & & A \parallel G \\ & {\scriptstyle f\parallel C} \searrow \quad \swarrow {\scriptstyle A\parallel g} & \\ & A \parallel B \parallel C & \end{array}$$

For any $x \in \mathcal{C}(F)$, we write $f\,x = x_A \parallel x_B$. Similarly, for every $y \in \mathcal{C}(G)$, we write $g\,y = y_B \parallel y_C$. The goal of this section is Proposition B.1.6, which proves that under some conditions on f and g, configurations of $G \odot^{f,g} F$ are in bijections with pairs of configurations of x and y that satisfies properties very alike to the usual minimal matching compatible, but such that the middle part of the interaction is "initialised" is the following sense.

- A configuration z of an event structure is called *initialised* if there is no minimal event e such that $z \mathrel{-\!\subset} z \cup \{e\}$. We write $\mathcal{C}_i(E)$ for the initialised configurations of E.
- A map $f : F \to A \parallel B$ of event structures is called *right-initialised* if for every $x \in \mathcal{C}_i(F)$, either $x_B \in \mathcal{C}_i(B)$ and we write $\overline{x} = x$, or there exists a unique $x \subseteq \overline{x} \in \mathcal{C}_i(F)$ such that $\overline{x}_B \in \mathcal{C}_i(B)$ and $\overline{x} \backslash x$ contains only events that are sent to minimal events of B. This is the case if f is a thunkable strategy as defined in Definition 5.5.1, or the parallel composition of multiple thunkable strategies.
- A map $g : G \to B \parallel C$ of event structures is called *left-initialised* if for every $y \in \mathcal{C}_i(G)$, $y_B \in \mathcal{C}_i(B)$ and moreover for every $m \in \mathcal{C}_i(B)$ containing only minimal events, there exists a unique $y_m \in \mathcal{C}_i(G)$ such that $g(y_m) = m$. This is the case if g is a negative strategy as defined in Definition 5.5.1, or the parallel composition of multiple negative strategies.

Lemma B.1.3. *If $f : F \to A \parallel B$ is right-initialised and $g : G \to B \parallel C$ is left-initialised, then for every $z \in \mathcal{C}_i(G \circledast^{f,g} F)$, either z has an initialised projection on B and we write $\overline{z} = z$, or there exists a unique $z \subseteq \overline{z} \in \mathcal{C}_i(G \circledast^{f,g} F)$ such that $\overline{z}$ has an initialised projection on B and $\overline{z} \backslash z$ contains only events send to minimal events of B.*

Definition B.1.4. *Given $f : F \to A \parallel B$ right-initialised and $g : G \to B \parallel C$ left-initialised, two matching compatible configurations $x \in \mathcal{C}(F)$ and $y \in \mathcal{C}(G)$ are said*

Initialised Matching Compatible *if $x_B = y_B$ is initialised.*

Minimal Initialised Matching Compatible *if moreover for every (x', y') initialised matching compatible, with $x'_A = x_A$, $y'_C = y_C$, $x' \subseteq x$ and $y' \subseteq y$, we have $x' = x$ and $y' = y$.*

We say that $z \in \mathcal{C}_i(G \odot^{f,g} F)$ is the initialised interactive composition of $x \in \mathcal{C}_i(F)$ and $y \in \mathcal{C}_i(G)$ if $[z]_\circledast = \overline{y \circledast x}$.

Lemma B.1.5. *We have the following properties*

- *For every pair $(x, y) \in \mathcal{C}_i(F) \times \mathcal{C}_i(G)$, there exists* at most one *$z \in \mathcal{C}_i(G \odot^{f,g} F)$ which is the initialised interactive composition of x and y. When it exists, we write it $y \odot_i x$.*
- *For every $z \in \mathcal{C}_i(G \odot^{f,g} F)$, there exists* exactly one *pair (x, y) such that $z = y \odot_i x$.*

Proposition B.1.6. *We consider $f : F \to A \parallel B$ is right-initialised and $g : G \to B \parallel C$ is left-initialised. We have a bijection between the set of configurations $z \in \mathcal{C}_i(G \odot^{f,g} F)$ and the set of pairs of minimal initialised matching compatible pairs $(x, y) \in \mathcal{C}_i(F) \times \mathcal{C}_i(G)$. The bijection is given by $z = y \odot_i x$.*

Lemma B.1.7. *The operation $\odot_i$ is union-preserving and intersection-preserving, i.e., if $(x, y), (x', y')$ are minimal initialised matching compatible, and $x \cup x'$ and $y \cup y'$ are configurations, then $(x \cup x', y \cup y')$ is minimal initialised matching compatible and $(y \cup y') \odot_i (x \cup x') = (y \odot_i x) \cup (y' \odot_i x')$; and a similar property for the intersection.*

B.1.3 Proof of Semi-Bifunctoriality

To prove the semi-bifunctoriality, we start by two lemmas which highlight the interaction of thunkability on the interactive composition.

Lemma B.1.8. *We assume $\sigma : A \twoheadrightarrow B$ and $\tau : B \twoheadrightarrow C$ two negative quantum strategies. If τ is thunkable and $y \odot x \in \mathcal{C}(T \odot S)$ then*

- *$y \odot x$ is empty if and only if y and x are both empty.*
- *$y \odot x$ is pre-value if and only if y is empty and x is pre-value.*
- *$y \odot x$ is post-value if and only if both y and x are post-value.*

Proof. For $y \odot x$ to be pre-value, either $y \circledast x$ contains an event sent to B, *i.e.*, y is pre-value and x is post-value, or $y \circledast x$ does not any event sent to B, *i.e.*, y is empty and x is pre-value. Since τ is thunkable, the only pre-value configurations are singletons. So if y is pre-value and x is post-value, that would make (x, y) a non-minimal matching compatible pair of configurations, since $(x \backslash \{v_x\}, \varnothing)$ with v_x the value-event of x is a

matching compatible pair with the same projection on A and B. This proves the first item. The second item is immediate. □

For the following lemma, we rely on $\odot_i$ defined in Appendix B.1.2. We recall that the main difference between $\odot_i$ and $\odot$ is that while $y \odot x$ can be such that the projection of y and x on B are empty, we expect in $y \odot_i x$ that y and x contain as many events minimal in B as possible.

Lemma B.1.9. *We assume $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ two negative quantum strategies. If σ is thunkable, then for $y \odot_i x \in \mathcal{C}_i(T \odot S)$ we have*

- *$y \odot_i x$ is pre-value if and only if y is pre-value and x is post-value.*
- *$y \odot_i x$ is post-value if and only if both y and x are post-value.*

Moreover, $\mathcal{Q}_{\tau\odot\sigma}(y \odot_i x) = \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x)$.

Proof. Since τ is negative, it is left-initialised as defined in Appendix B.1.2. Since σ is thunkable, it is right-initialised as defined in this same section. This allows us to use Proposition B.1.6, meaning that all the configurations of $\mathcal{C}_i(T \odot S)$ are of the form $y \odot_i x$.. For $y \odot_i x$ to be pre-value, either $y \circledast x$ contains an event sent to B, *i.e.*, y is pre-value and x is post-value, or $y \circledast x$ does not contain any event sent to B, *i.e.*, y is empty and x is pre-value. By definition of $\odot_i$, y is non-empty so this proves the first item. The second item is immediate.

The equation on quantum valuations comes from the thunkability of σ, and relies on the property "$d_\sigma(\{m\}, \{m, e\}) = 0$". We know that (x, y) is minimal initialised matching compatible. If (x, y) is also minimal matching compatible, then that means $y \odot x = y \odot_i x$, and we have

$$\mathcal{Q}_{\tau\odot\sigma}(y \odot_i x) = \mathcal{Q}_{\tau\odot\sigma}(y \odot x) = \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x)$$

Otherwise, using the results of Appendix B.1.2, this means that for $y' \odot x' = y \odot_i x$, we have $y' \circledast x' \subset y \circledast x$, with the difference being events that are sent to negative events in B. Since B is an arena, its minimal events are in conflict so $y' \circledast x' \overset{e}{-\!\!-\!\!\subset} y \circledast x$, with e being sent to B. So $y = (y' \sqcup \{\pi_T(e)^-\})$ and $x = (x' \sqcup \{\pi_S(e)^+\}))$. Using thunkability of σ, we obtain that necessarily x' is a singleton, and we have

$$d_\sigma(x'; x) = 0$$

In other words

$$\mathcal{Q}_\sigma(x') = \mathcal{H}_B(x'_B \subseteq^+ x_B) \circ \mathcal{Q}_\sigma(x)$$

On the other side, using obliviousness, we have

$$\mathcal{Q}_\tau(y) = \mathcal{Q}_\tau(y') \circ \mathcal{H}_B(y'_B \subseteq^+ y_B)$$

Combining the two, we obtain

$$\begin{array}{rcccc} \mathcal{Q}_{\tau\odot\sigma}(y \odot_i x) & = & \mathcal{Q}_{\tau\odot\sigma}(y' \odot x') & & \\ & = & \mathcal{Q}_\tau(y') \circ \mathcal{Q}_\sigma(x') & & \\ & = & \mathcal{Q}_\tau(y') \circ \mathcal{H}_B(x'_B \subseteq^+ x_B) \circ \mathcal{Q}_\sigma(x) & = & \mathcal{Q}_\tau(y) \circ \mathcal{Q}_\sigma(x) \end{array}$$

□

We now prove the semi-bifunctoriality.

Proposition B.1.10 (Semi-Bifunctoriality of the Parallel Tensor)**.** *For $\sigma : A \nrightarrow B$, $\sigma' : A' \nrightarrow B'$, $\tau : B \nrightarrow C$ and $\tau' : B' \nrightarrow C'$ two negative quantum strategies, we have the semi-bifunctoriality of $\otimes^p$, i.e., up to isomorphism*

$$(\tau \otimes^p \tau') \odot (\sigma \otimes^p \sigma') \cong (\tau \odot \sigma) \otimes^p (\tau' \odot \sigma')$$

whenever τ and τ' are thunkable, or σ and σ' are thunkable.

Proof. We write S, S', T, T' for the esps associated to the strategies $\sigma, \sigma', \tau, \tau'$. We will prove a union-preserving and intersection-preserving bijection between $\mathcal{C}(T \otimes^p T') \odot (S \otimes^p S')$ and $\mathcal{C}((T \odot S) \otimes^p (T' \odot S'))$, which will allow us to use Proposition 4.2.6 and conclude that they are isomorphic as esps.

We assume that τ and τ' are thunkable. It follows that $\tau \otimes \tau'$ is thunkable too. We take $(y \odot x) \in \mathcal{C}(T \odot S)$ and $(y' \odot x') \in \mathcal{C}(T' \odot S')$. Using Lemma B.1.8, we obtain that

$$((y \odot x), (y' \odot x')) \text{ synchronised} \iff (y, y') \text{ and } (x, x') \text{ synchronised}$$

We now have the following equivalence sequence:

$$\begin{array}{c} (y \odot x) \otimes^p (y' \odot x') \in \mathcal{C}((T \odot S) \otimes^p (T' \odot S')) \\ \Longleftrightarrow \\ \begin{cases} (y \odot x) \parallel (y' \odot x') \in \mathcal{C}((T \odot S) \parallel (T' \odot S')) \\ ((y \odot x), (y' \odot x')) \text{ synchronised} \end{cases} \\ \Longleftrightarrow \\ \begin{cases} (y \parallel y') \odot (x \parallel x') \in \mathcal{C}(T \parallel T') \odot (S \parallel S') \\ (y, y') \text{ and } (x, x') \text{ synchronised} \end{cases} \\ \Longleftrightarrow \\ (y \otimes^p y') \odot (x \otimes^p x') \in \mathcal{C}(T \otimes^p T') \odot (S \otimes^p S') \end{array}$$

This forms a union-preserving and intersection-preserving bijection, as Lemma 4.3.7 ensures that $\odot$ preserves union. Using Proposition 4.2.6, we know they are isomorphic as esps. We check without difficulties that the quantum valuations also match, so we have an isomorphism of quantum strategies.

We now assume instead that σ and σ' are thunkable. It follows that $\sigma \otimes \sigma'$ is thunkable too. We take $(y \odot_i x) \in \mathcal{C}_i(T \odot S)$ and $(y' \odot_i x') \in \mathcal{C}_i(T' \odot S')$. Using Lemma B.1.9, we obtain that

$$((y \odot_i x), (y' \odot_i x')) \text{ synchronised} \iff (y, y') \text{ and } (x, x') \text{ synchronised}$$

We now have the following equivalence sequence:

$$\begin{array}{c}
(y \odot_i x) \otimes^p (y' \odot_i x') \in \mathcal{C}_i((T \odot S) \otimes^p (T' \odot S')) \\
\iff \\
\begin{cases} (y \odot_i x) \parallel (y' \odot_i x') \in \mathcal{C}_i((T \odot S) \parallel (T' \odot S')) \\ ((y \odot_i x), (y' \odot_i x')) \text{ synchronised} \end{cases} \\
\iff \\
\begin{cases} (y \parallel y') \odot_i (x \parallel x') \in \mathcal{C}_i(T \parallel T') \odot (S \parallel S') \\ (y, y') \text{ and } (x, x') \text{ synchronised} \end{cases} \\
\iff \\
(y \otimes^p y') \odot_i (x \otimes^p x') \in \mathcal{C}_i(T \otimes^p T') \odot (S \otimes^p S')
\end{array}$$

This forms a union-preserving and intersection-preserving bijection, as Appendix B.1.2 ensures that $\odot_i$ preserves union. We extend without problems this bijection from $\mathcal{C}_i$ to $\mathcal{C}$ by mapping the empty configuration to the empty configuration. Using Proposition 4.2.6, we know they are isomorphic as esps. Using Lemma B.1.9, we check without difficulties that the quantum valuations also match, so we have an isomorphism of quantum strategies. □

B.2 Visibility and Deadlock-Free Composition

In this section, we provide a new proof for the deadlock-free composition of visible strategies. This proof is more general than the proof the can be found in [Cas17], as it extends to strategies between games that might not be forest-like. We start by recalling the definition of visibility we gave before.

B.2.1 Definitions

Definition B.2.1. *For $f : S \to E$ a map of es, we say that $s' \in |S|$ justifies $s \in |S|$ whenever $s' \leq_S s$ and $f(s') \rightarrowtriangle_E f(s)$.*

Definition B.2.2. *A (potentially partial) map of es $f : S \rightharpoonup E$ is said to be visible whenever for every gcc $\rho = s_0 \rightarrowtriangle_S \ldots \rightarrowtriangle_S s_n$ starting with a minimal event s_0, if $\sigma(s_n)$ is non-minimal in E, then there exists $s' \in \rho$ which justifies s_n.*

A practical characterisation of visibility is through grounded sets.

Definition B.2.3 (Grounded Sets)**.** *In an event structure E, a set of events $X \in \mathcal{P}_{\text{fin}}(E)$ is said to be grounded if for every $e \in X$, either e is minimal in E, or there exists $e' \rightarrowtriangle_E e$ such that $e' \in X$.*

We note that a grounded set are exactly union of gccs.

Lemma B.2.4. *A (potentially partial) map of es $f : S \rightharpoonup E$ is visible if and only if for every gcc ρ of S, $f\,\rho$ is grounded. A (potentially partial) map of es $f : S \rightharpoonup E$ is visible if and only if for every grounded set $G \in \mathcal{P}_{fin}(S)$, $f\,G$ is grounded.*

Proof. We take f visible. By induction on the size of gccs, we obtain that for every gcc ρ of S, we have $f\,\rho$ grounded. Since a grounded set is a union of gccs, and that grounded sets are stable by unions, it follows that the image of every grounded set is grounded. Conversely, if we assume that f preserves grounded sets, then for any gcc $\rho = s_0 \rightarrowtriangle_S \ldots \rightarrowtriangle_S s_n$, $\sigma\,\rho$ must be a grounded set. Using the definition of grounded sets, it follows that there is $e' \rightarrowtriangle_E \sigma\,s_n$ with $e' \in f\,\rho$. This mean that we have $s' \in \rho$ such that $f(s') \rightarrowtriangle_E f(s_n)$. □

While the definition of visibility does not depend on the property of the games, visibility is ill-behaving on some games of arbitrary shape. This is why in earlier papers, visibility was only defined on forest-like games. As a contribution of this thesis, we generalise to a larger class of games than forest-like games: the N-free games.

Definition B.2.5. *A game A is said to be N-free if such that*

$$\forall a, b, c, d \in |A| \text{ distincts with } \{a, b, c, d\} \in \text{Con}_A, \quad \begin{array}{ccc} a & & b \\ \downarrow & \searrow & \downarrow \\ c & & d \end{array} \implies b <_A c$$

We note that we only restrict immediate causality, and if $a <_A e <_A d$ instead, then we do not forbid $\{c, d\} \in \mathrm{Con}_A$. We also note that forest-like games (like arenas in Definition 5.5.1) are always N-free. The N-freeness is central into leveraging the courtesy property of our strategies:

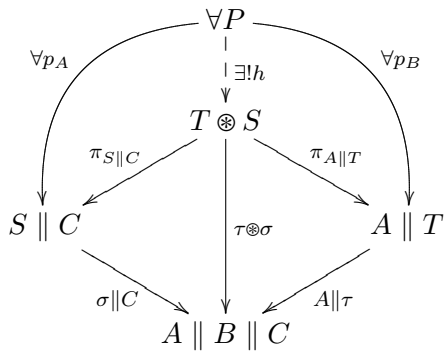
Lemma B.2.6 (Courtesy)**.** *If $\sigma : A \nrightarrow B$ is a strategy with A and B N-free games*[1]*, then whenever $s^+ <_S t^-$ with $\sigma(s)^+ \rightarrowtriangle_{A^\perp \| B} \sigma(t)^-$, we have $s^+ \rightarrowtriangle_S t^-$.*

Proof. Assume we have a chain $s = e_0 \rightarrowtriangle_S \ldots \rightarrowtriangle_S e_n = t$ with $n > 1$ (not necessarily a gcc as s might not be minimal). Since we have $s^+ \rightarrowtriangle_S e_1$, it follows from courtesy that $\sigma(s) \rightarrowtriangle_{A^\perp \| B} \sigma(e_1)$. Similarly, we have $\sigma(e_{n-1}) \rightarrowtriangle_{A^\perp \| B} \sigma(t)$. Assume $n = 2$, *i.e.*, $e_1 = e_{n-1}$. That would make $\sigma(s) \rightarrowtriangle_{A^\perp \| B} \sigma(e_1) \rightarrowtriangle_{A^\perp \| B} \sigma(t)$ and $\sigma(s) \rightarrowtriangle_{A^\perp \| B} \sigma(t)$, which is a contradiction. So $n > 2$ and $\sigma(s), \sigma(t), \sigma(e_1), \sigma(e_{n-1})$ are distinct. They form a N-pattern, so necessarily $\sigma(e_{n-1}) <_{A^\perp \| B} \sigma(e_1)$ or $\{\sigma(e_1), \sigma(t)\} \notin \mathrm{Con}_{A^\perp \| B}$, both contradicting $e_1 <_S e_{n-1} \leq_S t$. □

B.2.2 Category of Visible Strategies

To prove that visibility is preserved under interactive composition, we start by a lemma about the interaction.

Lemma B.2.7. *If we consider the interaction of two visible strategies $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ between N-free games, as described in the following diagram:*



and if we write $\pi_S : T \circledast S \rightharpoonup S$ and $\pi_T : T \circledast S \rightharpoonup T$ for the partial maps of es which are the projection maps respectively $\pi_{S\|C}$ and $\pi_{A\|T}$ post-composed by hiding maps, then π_S and π_T are visible.

[1] A game is an alternating race-free esps. We will not use the race-freeness in this section.

Proof. We take a gcc ρ in $T \circledast S$, and want to prove that its image on $S \parallel C$ and on $A \parallel T$ are grounded. If ρ is empty or singleton, this is trivially true. Otherwise, we proceed inductively. We have $\rho = \rho' \rightarrowtriangle_{T \circledast S} e$. We write $e' \rightarrowtriangle_{T \circledast S} e$ for the final element of ρ'. The event e is send by $\rho \circledast \sigma$ to an event either in A, B or C. We recall that by Lemma A.1.2, we have necessarily $\pi_S(e') \rightarrowtriangle_S \pi_S(e)$ or $\pi_T(e') \rightarrowtriangle_T \pi_T(e)$.

- We assume the former and write $\pi_S(e') = s' \rightarrowtriangle_S s = \pi_S(e)$. By induction hypothesis, $\pi_S \rho'$ and $\pi_T \rho'$ are grounded. Since e is justified by e', it follows that $\pi_S \rho$ is grounded. We need to prove that $\pi_T \rho$ is grounded.
  - If $\sigma(s) \in_{\parallel} |A|$, then $e \notin \mathrm{dom}(\pi_T)$, so $\pi_T \rho = \pi_T \rho'$. So $\pi_T \rho$ is grounded.
  - If $\sigma(s) \in_{\parallel} |B|$ and s positive, then $e \in \mathrm{dom}(\pi_T)$. We write $t = \pi_T(e)$, which is necessarily negative.
    - If $\tau(t)^-$ is minimal in $B^\perp \parallel C$, then by receptivity t^- is minimal in T. So π_T, ρ is grounded.
    - If $\tau(t)^-$ is non-minimal in $B^\perp \parallel C$, then $\sigma(s)^+$ is non-minimal in $A^\perp \parallel B$, so using visibility of σ, there exists a $s_0^- \in \sigma \rho'$ which justifies s^+. The mean that we have $e_0 \in \rho'$ such that $(\tau \circledast \sigma)(e_0) \rightarrowtriangle_{A \parallel B \parallel C} (\tau \circledast \sigma)(e)$. It also means that $(\tau \circledast \sigma)(e_0) \in_{\parallel} |B|$, so $e_0 \in \mathrm{dom}(\pi_T)$. We write $t_0 = \pi_T(e_0)$. We have t_0^+ justifies t^- (the polarities comes from the fact that the games are alternating). Using Lemma B.2.6, it we have $t_0^+ \rightarrowtriangle_T t^-$, so $\pi_T \rho$ is grounded.
  - If $\sigma(s) \in_{\parallel} |B|$ and s negative then using courtesy of σ we must have $\sigma(s'^+) \rightarrowtriangle_{A^\perp \parallel B} \sigma(s^-)$, so $\sigma(s'^+) \in_{\parallel} |B|$. Since $e, e' \in \mathrm{dom}(\pi_T)$, we write $t^+ = \pi_T(e)$ and $t'^- = \pi_T(e')$. Since we know that $\sigma(s'^+) \rightarrowtriangle_{A^\perp \parallel B} \sigma(s^-)$, we have $\tau(t'^-) \rightarrowtriangle_{B^\perp \parallel C} \tau(t^+)$. This mean that $\pi_T(e') = t' <_T t = \pi_T(e)$. Lastly, since $e' \rightarrowtriangle_{T \circledast S} e$, then using Lemma 4.2.5 we have $\pi_T(e') \rightarrowtriangle_T \pi_T(e)$, so $\pi_T \rho$ is grounded.
- In the latter case, we proceed symmetrically. □

Proposition B.2.8. *The copy-cat strategy is visible.*

This is pretty straightforward to prove, as gccs of copy-cat are always of the form $a^- \rightarrowtriangle a^+ \rightarrowtriangle b^- \rightarrowtriangle b^+ \rightarrowtriangle \ldots$, *i.e.*, alternating between a negative event of $A^\perp$ or A and the corresponding positive event in A or $A^\perp$ respectively, with each negative event being justified by the positive event before it.

Proposition B.2.9. *The interactive composition of two visible strategies between N-free games is visible.*

Proof. We consider the interactive composition of $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$. We note that to prove that a strategy is visible, we just need to prove the for all gcc ρ, the $(\tau \odot \sigma)\,\rho$ is grounded. So we take a gcc ρ of $T \odot S$. We look at the interaction $T \circledast S$, and choose a gcc ρ' of $T \circledast S$ such that its hiding is exactly ρ. Using Lemma B.2.7, we obtain that $\pi_T\,\rho'$ and $\pi_S\,\rho'$ are grounded. Since $(\tau \circ \pi_T)\,\rho \cup (\sigma \circ \pi_S)\,\rho = (\tau \circledast \sigma)\,\rho$, then using the visibility of τ and σ, we it follows that $(\tau \circledast \sigma)\,\rho$ is grounded in $A \parallel B \parallel C$, which implies that $(\tau \odot \sigma)\,\rho$ is grounded in $A \parallel C$. □

B.2.3 Deadlock-Free Composition

To prove deadlock-free composition, we need an additional restriction on our games: a game is called *polarised* whenever all its minimal events have the same polarity.

Theorem B.2.10 (Deadlock-Free Composition)**.** *We assume A, B, C to be N-free polarised games. If $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ are two visible strategies, and $x \in \mathcal{C}(S)$ and $y \in \mathcal{C}(T)$ are a pair of matching configurations, then they are matching compatible.*

Proof. In this proof, we see $(\sigma \parallel C^\perp)$ as a visible strategy from A to $B \parallel C^\perp$, and $(A \parallel \tau)$ as a visible strategy from $A \parallel B$ to C, and look at the following diagram:

$$\begin{array}{ccc} S \parallel C^\perp & & A \parallel T \\ \downarrow{\scriptstyle \sigma \parallel C^\perp} & & \downarrow{\scriptstyle A \parallel \tau} \\ A^\perp \parallel B \parallel C^\perp & \xrightarrow{\quad (-)^\perp \quad} & A \parallel B^\perp \parallel C \end{array}$$

We note that whenever $(\sigma \parallel C^\perp)(\ell) = (A \parallel \tau)(r)$, then ℓ and r have opposite polarities.

We take x and y matching configurations, so with $\sigma\, x = x_A \parallel x_B$ and $\sigma\, y = y_B \parallel y_C$ and $x_B = y_B$. Since those are a priori not compatible, we cannot write $y \circledast x$ yet. However, since we want to talk about this set of event to prove that it exists, we build instead the following set E.

We define $E = \{(\ell, r) \in (x \parallel y_C) \times (x_A \parallel y) \mid (\sigma \parallel C^\perp)(\ell) = (A \parallel \tau)(r)\}$, and the relation $(\ell, r) \blacktriangleleft (\ell', r)$ whenever $\ell <_{S \parallel C^\perp} \ell'$ or $r <_{A \parallel T} r'$. The matching configurations are compatible if an only if $(E, \blacktriangleleft)$ is a acyclic. We set up some additional definitions.

- We define $\pi(\ell, r)$ as $(\sigma \parallel C^\perp)(\ell) = (A \parallel \tau)(r)$.
- We write $(\ell, r) \triangleleft (\ell', r')$ whenever $\pi(\ell, r) \rightarrowtriangle_{A \parallel B \parallel C} \pi(\ell', r')$. We call it the justification order. We note that if we have $(\ell', r') \in E$ and $\ell \in x$ such that $(\sigma \parallel C^\perp)(\ell) \rightarrowtriangle_{A \parallel B \parallel C} (\sigma \parallel C^\perp)(\ell')$ then there exists a unique r such that $(\ell, r) \triangleleft (\ell', r)$. Symmetrically, when we have r, there exists a unique corresponding ℓ.

- For $(\ell, r) \in E$, we define its additive depth $d(\ell, r)$ as the sum over the $(\ell_i, r_i) \triangleleft (\ell, r)$ of the $d(\ell_i, r_i)$, plus 1. If the games are forest-like, this is simply the distance to the ground of $\pi(\ell, r)$. We note that if $(\ell, r) \triangleleft (\ell', r')$ then $d(\ell, r) < d(\ell', r')$.
- When we consider a cycle $(\ell_1, r_1) \blacktriangleleft \cdots \blacktriangleleft (\ell_n, r_n) \blacktriangleleft (\ell_1, r_1)$, with $n \geq 2$, we define its length as n and its depth as the sum of all the $d(\ell_i, r_i)$.

Assuming $(E, \blacktriangleleft)$ is not acyclic, we take a cycle of minimal depth $(\ell_1, r_1) \blacktriangleleft \cdots \blacktriangleleft (\ell_n, r_n) \blacktriangleleft (\ell_1, r_1)$. When we write (ℓ_i, r_i), we consider i modulo n. We prove a short lemma that will allow to eliminate a lot of contradictory cases.

Lemma B.2.11 (Justification)**.** *There is no (ℓ, r) such that*

$$(\ell_i, r_i) \trianglelefteq (\ell, r) \triangleleft (\ell_{i+1}, r_{i+1})$$

Proof. We necessarily have $(\ell, r) \blacktriangleleft (\ell_{i+2}, r_{i+2})$. If the hypothesis is an equality, then $(\ell_i, r_i) \blacktriangleleft (\ell_{i+2}, r_{i+2})$ so we can find a shortcut in the cycle. This would decrease the depth of the cycle, so this is a contradiction. If the hypothesis is not an equality, then we also found a cycle of smaller depth using $(\ell_i, r_i) \blacktriangleleft (\ell, r) \blacktriangleleft (\ell_{i+2}, r_{i+2})$. □

We can now analyse this minimal cycle, and find a lot of properties that minimality forces on it.

- This cycle alternate $\ell_i < \ell_{i+1}$ and $r_j < r_{j+1}$. Indeed, if we had $\ell_i < \ell_{i+1} < \ell_{i+2}$, then we could remove (ℓ_{i+1}, r_{i+1}) from the cycle and reduce the size. Same arguments for r_j. Without loss of generality, we assume that $\ell_{2k} < \ell_{2k+1}$ and $r_{2k+1} < r_{2k+2}$ for every k.
- The length n is even. Indeed, if it was odd, we would obtain $\ell_1 < \cdots < \ell_n < \ell_1$, hence $n = 1$. We assumed $n \geq 2$.
- This cycle is alternating in polarities, with (ℓ^-_{2k}, r^+_{2k}) and $(\ell^+_{2k+1}, r^-_{2k+1})$. Indeed, assume

  $$(\ell_{2k-1}, r_{2k-1}) \blacktriangleleft (\ell^+_{2k}, r^-_{2k}) \blacktriangleleft (\ell_{2k+1}, r_{2k+1})$$

  Using courtesy of $A \parallel \tau$, there exists $r_{2k-1} \leq r \twoheadrightarrow_{A \| T} r^-_{2k}$ such that $(A \parallel \tau)(r_{2k-1}) \twoheadrightarrow_{A \| B \| C} (A \parallel \tau)(r^-_{2k})$. We take $\ell \leq \ell_{2k}$ such that $(\ell, r) \triangleleft (\ell^+_{2k}, r^-_{2k})$. We have

  $$(\ell_{2k-1}, r_{2k-1}) \blacktriangleleft (\ell, r) \triangleleft (\ell^+_{2k}, r^-_{2k})$$

  By the justification lemma, we have a contradiction.

- All the $\pi(\ell_i, r_i)$ are in B. Indeed, assume $\pi(\ell_{2k}, r_{2k})$ or $\pi(\ell_{2k+1}, r_{2k+1})$ is in C. Since $\ell_{2k} <_{S \parallel C^\perp} \ell_{2k+1}$, and S and $C^\perp$ are disjoint in $S \parallel C^\perp$, we have both $\pi(\ell_{2k}, r_{2k})$ and $\pi(\ell_{2k+1}, r_{2k+1})$ in C. It follows that we have $\pi(\ell_{2k}, r_{2k}) \leq_C \pi(\ell_{2k+1}, r_{2k+1})$, meaning that $r_{2k} \leq_{A \parallel T} r_{2k+1}$, then

  $$(\ell_{2k-1}, r_{2k-1}) \triangleleft (\ell_{2k}, r_{2k})$$

  By the justification lemma, we have a contradiction. We proceed symmetrically to prove that none of them are in A.

- None of the $\pi(\ell_i, r_i)$ are minimals in B. Indeed, assume $\pi(\ell^-_{2k}, r^+_{2k})$ is minimal. By courtesy, it follows that ℓ^-_i is minimal in $S \parallel C^\perp$. We have

  $$(\ell_{2k}, r_{2k}) \blacktriangleleft (\ell_{2k+1}, r_{2k+1})$$

  Since $(\sigma \parallel C^\perp)(\ell_{2k})$ and $(\sigma \parallel C^\perp)(\ell_{2k+1})$ have opposite polarities, and B is positive or negative, both cannot be minimal events in B, so $(\sigma \parallel C^\perp)(\ell_{2k+1})$ is not minimal. Using visibility of $(\sigma \parallel C^\perp)$, and Lemma B.2.4, we obtain $\ell < \ell_{2k+1}$ such that $(\sigma \parallel C^\perp)(\ell) \rightarrowtriangle_{A \parallel B \parallel C} (\sigma \parallel C^\perp)(\ell_{2k+1})$ and ℓ and ℓ_{2k} are comparable in $S \parallel C^\perp$. Since ℓ_{2k} is minimal, it means $\ell_{2k} \leq \ell$. We take $r < r_{2k+1}$ such that $(l, r) \triangleleft (\ell_{2k+1}, r_{2k+1})$. So we have

  $$(\ell_{2k}, r_{2k}) \overset{\blacktriangleleft}{=} (l, r) \triangleleft (\ell_{2k+1}, r_{2k+1})$$

  By the justification lemma, this is a contradiction.

We now enter the core of the argumentation. We focus on an arbitrary (ℓ_{2k}, r_{2k}). We have

$$(\ell_{2k-1}, r_{2k-1}) \blacktriangleleft (\ell_{2k}, r_{2k})$$

We take a gcc ρ_0 of $A \parallel T$ ending on r^+_{2k} and containing r^-_{2k-1}. Since $(A \parallel \tau)(r_{2k})$ is not minimal, using visibility of $(A \parallel \tau)$ and Lemma B.2.4, we can find b^-_0 in ρ_0 such that $(A \parallel \tau)(b^-_0) \rightarrowtriangle_{A \parallel B \parallel C} (A \parallel \tau)(r^+_{2k})$. We find a^+_0 such that $(a^+_0, b^-_0) \triangleleft (\ell^-_{2k}, r^+_{2k})$. If we had $r^-_{2k-1} \leq b^-_0$, we could find a contradiction using the justification lemma. So $r^-_{2k-1} > b^-_0$. We then have:

$$(\ell_{2k-1}, r_{2k-1}) \blacktriangleright (a_0, b_0) \triangleleft (\ell_{2k}, r_{2k}) \blacktriangleleft (\ell_{2k+1}, r_{2k+1})$$

We now reuse a similar reasoning, taking a gcc ρ_1 of $S \parallel C^\perp$ ending on ℓ^+_{2k+1} and containing ℓ^-_{2k} **and** containing a_0. Since $(\sigma \parallel C^\perp)(\ell_{2k+1})$ is not minimal, using visibility of $(\sigma \parallel C^\perp)$, and Lemma B.2.4, we can find a^-_1 in ρ_1 such that $(\sigma \parallel C^\perp)(a^-_1) \rightarrowtriangle_{A \parallel B \parallel C}$

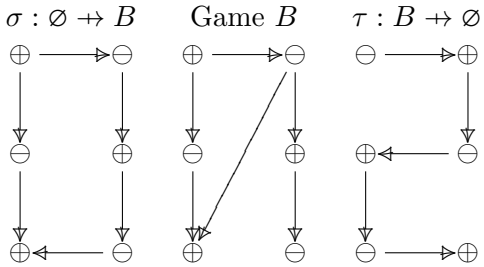


Figure B.1: Visibility does not prevent deadlock without N-freeness

$(\sigma \parallel C^{\perp})(\ell^{+}_{2k+1})$. We find b^{+}_{1} such that $(a^{-}_{1}, b^{+}_{1}) \triangleleft (\ell_{2k+1}, r_{2k+1})$. If we had $\ell^{-}_{2k} \leq a^{-}_{1}$, we could find a contradiction using the justification lemma so $\ell^{-}_{2k} > a^{-}_{1}$. We then have

$$\begin{array}{ccccc} & & (a^{+}_{0}, b^{-}_{0}) & & (a^{-}_{1}, b^{+}_{1}) \\ & \blacktriangleleft & \vartriangle & \blacktriangleleft & \vartriangle \\ (\ell^{+}_{2k-1}, r^{-}_{2k-1}) & \blacktriangleright & (\ell^{-}_{2k}, r^{+}_{2k}) & \blacktriangleleft & (\ell^{+}_{2k+1}, r^{-}_{2k+1}) \end{array}$$

However, we also note that ρ_1 was chosen such that a_0 was in it. So a^{+}_{0} and a^{-}_{1} are comparable. From polarity, they cannot be equal. Since B is N-free, using the courtesy lemma we that $a^{+}_{0} \rightarrowtriangle_S \ell^{-}_{2k}$. Since we have $a^{-}_{1} <_S \ell^{-}_{2k}$ and a_0, a_1, ℓ_{2k} in the same gcc, we must have $a^{-}_{1} \leq a^{+}_{0}$, hence

$$(a_0, b_0) \blacktriangleright (a_1, b_1)$$

We iterate, using dual reasoning for odd indices, and create a chain

$$(a_0, b_0) \blacktriangleright (a_1, b_1) \blacktriangleright (a_2, b_2) \blacktriangleright \dots$$

Since all of them are lower in the justification order than elements of the cycle, which form a finite set, we can take the smallest j such that there is a $i < j$ with $(a_i, b_i) = (a_j, b_j)$. So we have

$$(a_i, b_i) \blacktriangleright \cdots \blacktriangleright (a_j, b_j) = (a_i, b_i)$$

We remark that the depth of this cycle is at least n lower than the initial depth, so we found a contradiction. □

Bibliography

[AB11] S. Abramsky and A. Brandenburger. A unified sheaf-theoretic account of non-locality and contextuality. *CoRR*, 2011. URL http://arxiv.org/abs/1102.0264.

[ABK+15] S. Abramsky, R. S. Barbosa, K. Kishida, R. Lal, and S. Mansfield. Contextuality, cohomology and paradox. In S. Kreutzer, editor, *24th EACSL*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CSL.2015.211.

[AJM00] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Inf. Comput.*, 2000. doi:10.1006/inco.2000.2930.

[AM97] S. Abramsky and G. McCusker. Call-by-value games. In M. Nielsen and W. Thomas, editors, *CSL*, Lecture Notes in Computer Science. Springer, 1997. doi:10.1007/BFb0028004.

[AM99] S. Abramsky and P. Melliès. Concurrent games and full completeness. In *14th Annual Logic in Computer Science*. IEEE Computer Society, 1999. doi:10.1109/LICS.1999.782638.

[BDER97] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Timeless games. In *11th CSL*, 1997. doi:10.1007/BFb0028007.

[BGMZ14] A. Brunel, M. Gaboardi, D. Mazza, and S. Zdancewic. A core quantitative coeffect calculus. In *23rd ESOP*. Springer, 2014. doi:10.1007/978-3-642-54833-8_19.

[Cas17] S. Castellan. *Concurrent structures in game semantics. (Structures concurrentes en sémantique des jeux)*. PhD thesis, University of Lyon, France, 2017. URL https://tel.archives-ouvertes.fr/tel-01587718.

[CCHW18] S. Castellan, P. Clairambault, J. Hayman, and G. Winskel. Non-angelic concurrent game semantics. In *21st FOSSACS*. Springer, 2018. doi:10.1007/978-3-319-89366-2_1.

[CCPW18] S. Castellan, P. Clairambault, H. Paquet, and G. Winskel. The concurrent game semantics of probabilistic PCF. In A. Dawar and E. Grädel, editors, *LICS*. ACM, 2018. doi:10.1145/3209108.3209187.

[CCRW17] S. Castellan, P. Clairambault, S. Rideau, and G. Winskel. Games and strategies as event structures. *Log. Methods Comput. Sci.*, 2017. doi:10.23638/LMCS-13(3:35)2017.

[CCW15a] S. Castellan, P. Clairambault, and G. Winskel. The parallel intensionally fully abstract games model of PCF. In *LICS*. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.31.

[CCW15b] S. Castellan, P. Clairambault, and G. Winskel. The parallel intensionally fully abstract games model of PCF. In *30th LICS*. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.31.

[CCW17] S. Castellan, P. Clairambault, and G. Winskel. Observably deterministic concurrent strategies and intensional full abstraction for parallel-or. In D. Miller, editor, *FSCD*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.FSCD.2017.12.

[CCW19] S. Castellan, P. Clairambault, and G. Winskel. Thin games with symmetry and concurrent Hyland-Ong games. *Log. Methods Comput. Sci.*, 2019. doi:10.23638/LMCS-15(1:18)2019.

[CdV20] P. Clairambault and M. de Visme. Full abstraction for the quantum lambda-calculus. *Proc. ACM Program. Lang.*, POPL, 2020. doi:10.1145/3371131.

[CdVW19] P. Clairambault, M. de Visme, and G. Winskel. Game semantics for quantum programming. *Proc. ACM Program. Lang.*, POPL, 2019. doi:10.1145/3290345.

[CHLW14] S. Castellan, J. Hayman, M. Lasson, and G. Winskel. Strategies as concurrent processes. In *30th MFPS*. Elsevier, 2014. doi:10.1016/j.entcs.2014.10.006.

[Cla20] P. Clairambault. Learning to count up to symmetry. *CoRR*, 2020, 2006.05080. URL `https://arxiv.org/abs/2006.05080`.

[CM10] A. C. Calderon and G. McCusker. Understanding game semantics through coherence spaces. *Electr. Notes Theor. Comput. Sci.*, 2010. doi:10.1016/j.entcs.2010.08.014.

[CP18] P. Clairambault and H. Paquet. Fully abstract models of the probabilistic lambda-calculus. In *27th EACSL CSL*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CSL.2018.16.

[Del11] Y. Delbecque. Game semantics for quantum data. *Electr. Notes Theor. Comput. Sci.*, 2011. doi:10.1016/j.entcs.2011.01.005.

[DH02] V. Danos and R. Harmer. Probabilistic game semantics. *ACM Trans. Comput. Log.*, 2002. doi:10.1145/507382.507385.

[dV19] M. de Visme. Event structures for mixed choice. In W. J. Fokkink and R. van Glabbeek, editors, *30th CONCUR*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CONCUR.2019.11.

[Ead18] H. Eades III. Linear categories: A folklore simplification. https://blog.metatheorem.org/2018/07/24/Linear-Categories-A-Folklore-Simplification.html, 2018.

[Ehr12] T. Ehrhard. The Scott model of linear logic is the extensional collapse of its relational model. *Theor. Comput. Sci.*, 2012. doi:10.1016/j.tcs.2011.11.027.

[ETP14] T. Ehrhard, C. Tasson, and M. Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *The 41st POPL*. ACM, 2014. doi:10.1145/2535838.2535865.

[Gir87] J. Girard. Linear logic. *Theor. Comput. Sci.*, 1987. doi:10.1016/0304-3975(87)90045-4.

[Gir89] J.-Y. Girard. Geometry of interaction 1: Interpretation of system F. *Studies in logic and the foundations of mathematics*, 1989.

[GLR+13] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron. Quipper, a scalable quantum programming language. *ACM SIGPLAN Notices*, 2013. doi:10.1145/2499370.2462177.

[Gou18] E. Goursat. *Cours d'analyse mathématique*. Gauthier-Villars, 1918.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Symposium on the Theory of Computing*. ACM, 1996. doi:10.1145/237814.237866.

[GRTZ02] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden. Quantum cryptography. *Reviews of modern physics*, 2002.

[GRW80] G. Ghirardi, A. Rimini, and T. Weber. A general argument against superluminal transmission through the quantum mechanical measurement process. In *Lettere al Nuovo Cimento (1971-1985)*, 1980. doi:10.1007/BF02817189.

[HH17] I. Hasuo and N. Hoshino. Semantics of higher-order quantum computation via geometry of interaction. *Ann. Pure Appl. Logic*, 2017. doi:10.1016/j.apal.2016.10.010.

[HO00] J. M. E. Hyland and C. L. Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 2000. doi:10.1006/inco.2000.2917.

[HY97] K. Honda and N. Yoshida. Game theoretic analysis of call-by-value computation. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *24th ICALP*, Lecture Notes in Computer Science. Springer, 1997. doi:10.1007/3-540-63165-8_180.

[LFVY17] U. D. Lago, C. Faggian, B. Valiron, and A. Yoshimizu. The geometry of parallelism: classical, probabilistic, and quantum effects. In G. Castagna and A. D. Gordon, editors, *POPL*. ACM, 2017.

[Mel05] P. Melliès. Asynchronous games 4: A fully complete model of propositional linear logic. In *20th LICS*. IEEE Computer Society, 2005. doi:10.1109/LICS.2005.6.

[Mel09] P.-A. Melliès. *Categorical semantics of linear logic*. Panoramas et Synthèses. Société Mathématique de France, 2009.

[Mil77] R. Milner. Fully abstract models of typed *lambda*-calculi. *Theor. Comput. Sci.*, 1977.

[ML98] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 2 edition, 1998.

[MSS13] O. Malherbe, P. Scott, and P. Selinger. Presheaf models of quantum computation: An outline. In *Computation, Logic, Games, and Quantum Foundations.*, 2013. doi:10.1007/978-3-642-38164-5_13.

[NPW79] M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains. In *Semantics of Concurrent Computation*, 1979. doi:10.1007/BFb0022474.

[PEO15] D. N. Pavel Etingof, Shlomo Gelaki and V. Ostrik. Chapter 2: Monoidal categories. In *Tensor categories, Mathematical Surveys and Monograph*, volume Volume 205. American Mathematical Society, 2015.

[Plo81] G. D. Plotkin. Post-graduate lecture notes in advanced domain theory (incorporating the "Pisa Notes"). *Dept. of Computer Science, Univ. of Edinburgh*, 1981.

[PR97] J. Power and E. Robinson. Premonoidal categories and notions of computation. *Math. Struct. Comput. Sci.*, 1997. doi:10.1017/S0960129597002375.

[PSV14] M. Pagani, P. Selinger, and B. Valiron. Applying quantitative semantics to higher-order quantum computing. In *The 41st POPL*. ACM, 2014. doi:10.1145/2535838.2535879.

[PT97] J. Power and H. Thielecke. Environments, continuation semantics and indexed categories. In M. Abadi and T. Ito, editors, *TACS*, Lecture Notes in Computer Science. Springer, 1997. doi:10.1007/BFb0014560.

[PT99a] J. Power and H. Thielecke. Closed Freyd- and kappa-categories. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *26th ICALP*, Lecture Notes in Computer Science. Springer, 1999. doi:10.1007/3-540-48523-6_59.

[PT99b] J. Power and H. Thielecke. Closed Freyd- and kappa-categories. In *ICALP'99, Proceedings*, 1999. doi:10.1007/3-540-48523-6_59.

[Rob19] S. J. Robertson. Topics in Hilbert spaces, spectral theory, and harmonic analysis, 2019, 1909.13156.

[RW11] S. Rideau and G. Winskel. Concurrent strategies. In *LICS*. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.13.

[SCS91] C. Seely, J. R. B. Cockett, and R. A. G. Seely. Weakly distributive categories. In *Journal of Pure and Applied Algebra*. University Press, 1991.

[Sel04] P. Selinger. Towards a quantum programming language. *Math. Struct. Comput. Sci.*, 2004. doi:10.1017/S0960129504004256.

[Sel07] P. Selinger. Dagger compact closed categories and completely positive maps (extended abstract). *Electron. Notes Theor. Comput. Sci.*, 2007. doi:10.1016/j.entcs.2006.12.018.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 1997. doi:10.1137/S0097539795293172.

[SMR61] E. C. G. Sudarshan, P. M. Mathews, and J. Rau. Stochastic dynamics of quantum-mechanical systems. *Phys. Rev.*, 1961. doi:10.1103/PhysRev.121.920.

[Sta14] S. Staton. Freyd categories are enriched Lawvere theories. *Electron. Notes Theor. Comput. Sci.*, 2014. doi:10.1016/j.entcs.2014.02.010.

[Sum09] E. Sumii. A complete characterization of observational equivalence in polymorphic *lambda*-calculus with general references. In E. Grädel and R. Kahle, editors, *23rd CSL 2*, Lecture Notes in Computer Science. Springer, 2009. doi:10.1007/978-3-642-04027-6_33.

[SV06] P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 2006. doi:10.1017/S0960129506005238.

[SV08] P. Selinger and B. Valiron. On a fully abstract model for a quantum linear functional language (extended abstract). *Electron. Notes Theor. Comput. Sci.*, 2008. doi:10.1016/j.entcs.2008.04.022.

[TAO18] T. Tsukada, K. Asada, and C. L. Ong. Species, profunctors and taylor expansion weighted by SMCC: A unified framework for modelling nondeterministic, probabilistic and quantum programs. In A. Dawar and E. Grädel, editors, *LICS*. ACM, 2018. doi:10.1145/3209108.3209157.

[Win07] G. Winskel. Event structures with symmetry. *Electron. Notes Theor. Comput. Sci.*, 2007. doi:10.1016/j.entcs.2007.02.022.

[Win11] G. Winskel. Ecsym notes on event structures, stable families and concurrent games. https://www.cl.cam.ac.uk/ gw104/ecsym-notes.pdf, 2011.

[Win13] G. Winskel. Strategies as profunctors. In F. Pfenning, editor, *FOSSACS*, Lecture Notes in Computer Science. Springer, 2013. doi:10.1007/978-3-642-37075-5_27.

[Win14] G. Winskel. Probabilistic and quantum event structures. In *Horizons of the Mind. A*, 2014. doi:10.1007/978-3-319-06880-0_25.

[ZF10] D. Zhao and T. Fan. Dcpo-completion of posets. *Theor. Comput. Sci.*, 2010. doi:10.1016/j.tcs.2010.02.020.

[Öm05] B. Ömer. Classical concepts in quantum programming. *International Journal of Theoretical Physics*, 2005. doi:10.1007/s10773-005-7071-x.