Quantum Games as Quantum Types

Yannick Delbecque
Doctor of Philosophy
McGill University,
Montreal, Quebec, August 2008

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements for the degree of Doctor of Philosophy in computer science

Acknowledgements

I would like to extend my gratitude to my supervisor Prakash Panangaden for his shared wisdom and insight, as well as for his continued support throughout the most difficult stages. I also would like to thank him for the financial support that allowed me to complete this thesis. I am also grateful for the extra financial support provided by Patrick Hayden.

I would like to thank the many researchers, colleagues and friends whose help and comments played an important part in this research.

Finally, I would like to thank all family and friends without whose encouragement and support I could not have completed this thesis. Special thanks to my mother for her unwavering support and to my father for lighting the spark of scientific curiosity in me. Finally, I am grateful to all those friends who provided me with enough tea keep my mind clear throughout these years.

Abstract

In this thesis, we present a new model for higher-order quantum programming languages. The proposed model is an adaptation of the probabilistic game semantics developed by Danos and Harmer [DH02]: we expand it with *quantum strategies* which enable one to represent quantum states and quantum operations. Some of the basic properties of these strategies are established and then used to construct denotational semantics for three quantum programming languages. The first of these languages is a formalisation of the *measurement calculus* proposed by Danos et al. [DKP07]. The other two are new: they are higher-order quantum programming languages. Previous attempts to define a denotational semantics for higher-order quantum programming languages have failed. We identify some of the key reasons for this and base the design of our higher-order languages on these observations.

The game semantics proposed in this thesis is the first denotational semantics for a λ-calculus equipped with quantum types and with extra operations which allow one to program quantum algorithms. The results presented validate the two different approaches used in the design of these two new higher-order languages: a first one where quantum states are used through references and a second one where they are introduced as constants in the language. The quantum strategies presented in this thesis allow one to understand the constraints that must be imposed on quantum type systems with higher-order types. The most significant constraint is the fact that abstraction over part of the tensor product of many unknown quantum states must not be allowed.

Quantum strategies are a new mathematical model which describes the interaction between classical and quantum data using system-environment dialogues. The interactions between the different parts of a quantum system are described using the rich structure generated by composition of strategies. This approach has enough generality to be put in relation with other work in quantum computing. Quantum strategies could thus be useful for other purposes than the study of quantum programming languages.

Abrégé

Nous présentons dans cette thèse un nouveau modèle pour les langages de programmation quantique. Notre modèle est une adaptation de la sémantique de jeux probabilistes définie par Danos et Harmer [DH02]: nous y ajoutons des *stratégies quantiques* pour permettre la représentation des états et des opérations quantiques. Nous établissons quelques propriétés de base de ces stratégies. Ces propriétés sont ensuite utilisées pour construire des sémantiques dénotationnelles pour trois langages de programmation quantique. Le premier langage est une formalisation du *calcul par mesures* proposé par Danos et al. [DKP07]. Les deux autres langages sont nouveaux: ce sont des langages quantiques d'ordre supérieur dont la syntaxe a été construite à partir d'observations expliquant l'échec des tentatives précédentes pour construire une sémantique dénotationnelle pour de tels langages.

La sémantique de jeux présentée dans cette thèse est la première sémantique dénotationnelle pour de tels λ-calculs équipés de types et d'opérations supplémentaires permettant la programmation d'algorithmes quantiques. Les résultats présentés valident les deux approches différentes utilitées dans la conception de ces deux nouveaux languages d'ordre supérieur: une première où les états quantiques sont indirectement accessibles via des références et une seconde où ils sont introduit directement comme des constantes dans le langage. Les stratégies quantiques présentées permettent de comprendre les contraintes devant être imposées aux systèmes de type quantique comportant des types d'ordre supérieurs. La contrainte la plus importante est le fait que l'abstraction sur une partie d'un état quantique comportant plusieurs qbits inconnus doit être prohibée.

Les stratégies quantiques constituent un nouveau modèle mathématique qui décrit l'interaction entre les données classiques et quantiques par des dialogues entre système et environnement. L'interaction entre les differentes parties d'un système quantique y est décrite à l'aide d'une structure riche en utilisant la composition de strategies. L'approche utilisé est assez générale pour être mise en relation avec d'autres travaux en informatique quantique. Les propriétés des stratégies quantiques pourraient donc être utiles à d'autre fins que l'étude des languages de programmation quantiques.

Contents

1 Introduction **9**

1.1 Quantum programming language theory . . . 9

1.2 Game semantics . . . 10

1.3 Overview . . . 11

2 Background **12**

2.1 Quantum computing . . . 12

2.1.1 Linear algebra and the Dirac notation . . . 12

2.1.2 Quantum mechanics . . . 15

2.1.3 Quantum computation . . . 21

2.2 Quantum programming languages . . . 22

2.2.1 The Measurement Calculus . . . 22

2.2.2 Quantum λ-calculus . . . 24

2.2.3 Categorical quantum mechanics . . . 28

2.2.4 Abstract quantum mechanics . . . 32

2.3 Game semantics . . . 33

2.3.1 Arenas . . . 33

2.3.2 Plays and threads . . . 34

2.3.3 Operations on arenas . . . 35

2.3.4 Probabilistic strategies . . . 36

2.3.5 Strategies and threads . . . 37

3 Quantum games and strategies **40**

3.1 Arenas for quantum systems . . . 40

3.1.1 Quantum Games . . . 41

3.2 Arenas for isolated quantum systems . . . 43

3.3 Quantum strategies for isolated quantum arenas . . . 44

3.3.1 Consistent histories . . . 47

3.4 General quantum arenas . . . 48

3.4.1 Quantum arenas . . . 49

3.4.2 Products of quantum arenas . . . 49

3.4.3 The linear arrow quantum arena . . . 50

3.5 The category of quantum arenas and quantum strategies . . . 55

3.5.1 Quantum strategies as probabilistic strategies . . . 55

3.5.2 Tensor product of strategies with classical interactions 55
3.6 Quantum strategies using other quantum measurements 59
3.6.1 Generalised measurement based quantum strategies 59
3.6.2 POVM based quantum strategies 60
3.6.3 Intervention operators 61

4 Game semantics for the measurement calculus 64
4.1 MCdata 64
4.1.1 Syntax 64
4.1.2 Operational semantics 68
4.2 Denotational semantics 68
4.3 Soundness 73

5 λ-calculus with quantum stores 81
5.1 Critique of the quantum λ-calculus 81
5.2 Simply typed λ-calculus with quantum stores 83
5.2.1 Syntax 83
5.2.2 Types 85
5.2.3 Operational semantics 86
5.2.4 Denotational semantics 89
5.3 Soundness 93

6 λ-calculus with quantum data 101
6.1 Syntax 101
6.1.1 Terms 101
6.1.2 Types 102
6.2 Operational semantics 104
6.3 Denotational semantics 104
6.4 Soundness 111

7 Conclusion 115
7.1 Recapitulation 115
7.2 Discussion 116
7.3 Future work 117

List of Tables

2.1 Basic quantum circuit operations 22
2.2 Quantum λ-calculus subtyping rules 25
2.3 Quantum λ-calculus typing rules 26
2.4 Quantum λ-calculus reduction rules 27

4.1 MCdata typing rules 66
4.2 MCdata reduction rules 69
4.3 MCdata denotational semantics 71

5.1 QSL typing rules 85
5.2 QSL probabilistic reduction rules 87
5.3 QSL denotational semantics 91

6.1 QDL typing rules 103
6.2 QDL probabilistic reduction rules 105
6.3 QDL denotational semantics 106

List of Figures

5.1 QSL teleportation 88

6.1 QDL Teleportation 103
6.2 Strategy for the QDL measurement rules 108
6.3 Strategy for the first QDL tensor rule 109
6.4 Strategy for the second QDL tensor rule 110
6.5 Strategy for the third QDL tensor rule 110

Chapter 1

Introduction

> Une question presque insondable, où nous ne nous arrêterons pas, est de savoir jusqu'à quel point nos moyens de raisonnement offrent, par essence, le caractère de règles de jeu, autrement dit, ne sont valables que dans un certain cadre intellectuel, où on les tient pour impérieux. Y a-t-il toujours dans la logique en général, et dans le syllogisme en particulier, une convention ludique tacite, par laquelle on tient compte de la valeur des catégories et des concepts comme des pions et des cases d'un échiquier? À d'autres de trancher la question.
>
> Johan Huizinga
> *Homo Ludens, essai sur la fonction sociale du jeu*, 1938

1.1 Quantum programming language theory

Quantum algorithms are usually described using the low-level formalism of quantum circuits. This approach is very useful to study the complexity of quantum algorithms, a theme which, together with quantum information and quantum cryptography, is one of the central research preoccupations in the field of quantum computing. Another way to study quantum algorithms was not given much attention until recently: the development of more structured languages to describe quantum computation. The study of the structure and the various semantics of a programming language is an important way to understand how programming constructs, like control flow mechanisms, abstractions, stores and other programming languages features, interact with each other and contribute to the expressiveness and structure of the programming language. Since quantum computing introduces radically new computing constructs, it is natural to try to apply the methods of programming language semantics to understand their contribution in a similar manner.

Many quantum programming languages have been proposed, starting from the quantum pseudocode of Knill [Kni96], to the more recent quantum λ-calculus proposed by Selinger and Valiron [SV06a]. In the last few years the structure of quantum programming languages has become a topic of study in itself, using various ideas from category theory and classical programming language theory. Many important questions in this field can be seen as variants of a central one: *what is the structure of the interactions between classical and quantum data?* These interactions are the key to understanding the basic quantum mechanical operations like measurements and tensor products, which together lead to many counter-intuitive phenomena associated to quantum mechanics, like non-locality. Understanding them is also a central problem if we want to integrate quantum programming constructs in a classical language. An important conceptual problem is the design of a higher-order quantum language, a problem which is also related to these classical-quantum interactions. As a final example of the importance of this question, consider the problem of mixing classical and quantum data in the graphical calculi. These graphical languages are diagrammatic formalisms abstracting from the language of monoidal categories and have proven very useful in understanding and reasoning about abstract quantum mechanics [AC04, CP06a, Sel07]. These graphical languages provide a structure to understand the flow of quantum information in quantum protocols and algorithms. To incorporate classical data in them, such as the data arising from measurements, classical data is represented using a choice of particular basis in the Hilbert space model used for quantum data. This idea has been abstracted in the language of symmetric monoidal categories as *classical objects* equipped with morphisms which allow one to use the classical data encoded as quantum states.

1.2 Game semantics

The goal of this thesis is to adapt *game semantics* to quantum computing. Game semantics was a very successful approach in the field of programming language theory. It was adapted to analyse many different programming language features using a common set of basic concepts. The central idea of game semantics is to represent computations as interactions or dialogues between a system executing a program and its environment. A program is viewed as a *strategy* that tells the system how to choose its next action using the preceding part of the interaction. By adapting the rules governing these interactions, game semantics can be used to model, in a very tight manner, many different languages.

It should be noted that the games referred to in game semantics are not at all like the games discussed in traditional game theory. In game semantics of programming languages, the concept of winning and losing, or of more general payoff schemes, is not used because the focus is on the structure of the possible interactions between the players. In contrast, in traditional game theory one typically does not study the interaction between agents; the focus is to find optimal strategies for the players.

The quantum games we present in this thesis are defined following the ideas of game semantics: they are used to model quantum computation as an interaction between a quantum system and its environment. We believe this approach is an interesting guide when we seek answers about the central question of classical-quantum interactions, the central idea of game semantics being interactions between systems. We introduce *quantum strategies* to the arsenal of game semantics, and use them to analyse various quantum programming languages. In classical programming language theory the main reason to use games and strategies as denotation of programs is to get *full abstraction* results. To prove these results, one has to use the fact that the game's rules can be adapted to characterise the programs tightly. Full abstraction is not our main goal in this work; here we focus on using quantum games and strategies to understand the structure of quantum programming languages in terms of interactions.

1.3 Overview

In chapter 2, we give an overview of quantum computing and of game semantics. We also present three example quantum programming languages. In chapter 3, we define and explore a notion of quantum strategy based on previous work on probabilistic game semantics. The three remaining chapters use quantum strategies to define denotational semantics for a typed variant of the measurement calculus 4 and for two new quantum λ-calculi that we introduce in chapter 5 and chapter 6.

Chapter 2

Background

2.1 Quantum computing

2.1.1 Linear algebra and the Dirac notation

We need first to review some basic linear algebra results. In this thesis, we use the Dirac (or so called "bra-ket") notation widely used in quantum mechanics and quantum computing.

Hilbert spaces A **complex Hilbert space** H is a vector space over $\mathbb{C}$ equipped with an inner product $(-,-)_H$ and which is complete with respect to the associated norm, defined by $||u|| = \sqrt{(u,u)}$. Unless stated otherwise, all Hilbert spaces are assumed to be of finite dimension. Such spaces are isomorphic to $\mathbb{C}^n$ and are automatically complete for any inner product. The elements of H are called **kets**, written with the right half of a bracket: the vector u is denoted by $|u\rangle$. For an indexed family of vectors, it is customary to keep only the indexes in the notation. For example, if e_0, e_1 is a basis of H, the usual notation is $|0\rangle, |1\rangle$ instead of $|e_0\rangle, |e_1\rangle$.

Given a vector $|u\rangle \in H$, the linear function $\langle u| \colon H \to \mathbb{C}$ is defined by

$$\langle u| \left(|v\rangle\right) = \left(|u\rangle, |v\rangle\right)_H.$$

Functions defined in this manner are called **bras**. The function mapping $|u\rangle$ to $\langle u|$ is denoted $\dagger$. The Dirac notation allows a simplification of the inner product notation $(|u\rangle, |v\rangle)$ by using the simpler notation $\langle u|v\rangle$.

An **orthonormal basis** of H is a generating set of vectors $|i\rangle \in H$ such that $\langle i|j\rangle = \delta_{ij}$, where

$$\delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

Dual spaces The **dual** H^* of a Hilbert space H is the vector space of all linear functionals $H \to \mathbb{C}$. The *Riesz representation theorem* for finite dimensional complex Hilbert spaces says that all functionals in H^* are bras.

Given a basis $|i\rangle$ for H, the set of functionals $\langle i|$ is a basis of H^* called the **dual basis** of $|i\rangle$. It is the unique set of functionals $\langle i|$ such that$\langle i|j\rangle = \delta_{ij}$. H^* is always a Hilbert space because we can define $\langle\langle u|, \langle v|\rangle_{H^*}$ by $\langle v|u\rangle$. Note that the map † takes $A\colon H_1 \to H_2$ to $A^\dagger\colon H_2^* \to H_1^*$.

Maps between Hilbert spaces We denote the set of all linear maps $H_1 \to H_2$ by $\hom(H_1, H_2)$; the set of all linear maps $H \to H$ is $\mathbf{M}(H)$. The **matrix representation** with respect to bases $|i\rangle$ and $|j\rangle$ of H_1 and H_2 of a map $A \in \hom(H_1, H_2)$ is the matrix with entries $a_{ij} = \langle j|A|i\rangle = (|j\rangle, A|i\rangle)$.

There is a natural way to extend the map † to maps $A\colon H_1 \to H_2$: the **adjoint** $A^\dagger\colon H_2^* \to H_1^*$ to a map A is defined by

$$A^\dagger(\langle u|) = (A|u\rangle)^\dagger.$$

It follows directly from the definitions that if a_{ij} are the components of the matrix representation of A, then the elements of the matrix representation of $A^\dagger$ are $\overline{a_{ji}}$, where the overbar denotes complex conjugation.

The **unitary** maps $U \in \mathbf{M}(H)$ are those satisfying $U^\dagger = U^{-1}$. Unitary maps preserve inner products and norms since $\langle u|U^\dagger U|v\rangle = \langle u|v\rangle$.

A linear map $A \in \mathbf{M}(H)$ is **Hermitian** if $A^\dagger = A$.

A **projection operator** is a Hermitian map $P\colon H \to H$ such that $P^2 = P$. The set of projection operators on H is denoted by $\mathbf{P}(H)$. Two projection operators are **orthogonal** if $P_1P_2 = P_2P_1 = 0$. This relation is denoted by $P_1 \perp P_2$. A family of projectors P_i is **complete** if they are pairwise orthogonal and

$$\sum_i P_i = I_H.$$

In Dirac notation, $|u\rangle\langle v|$ denotes the linear map $H \to H$ defined by

$$|u\rangle\langle v|\,(|w\rangle) = \langle v|w\rangle|u\rangle.$$

In particular, the map $|u\rangle\langle u|$, which we also denote by $[u]$, is the projection map onto the subspace spanned by $|u\rangle$. Given an orthonormal basis $|i\rangle$, a very useful identity is $\sum_i |i\rangle\langle i| = I_H$.

Given an orthonormal basis $|i\rangle$ of H, the set of maps $|i\rangle\langle j|$ is a basis of $\mathbf{M}(H)$.

Theorem 2.1. *(Spectral decomposition) Let H be a complex Hilbert space. For every Hermitian map $A \in \mathbf{M}$, there is an orthonormal basis $|i\rangle$ and complex numbers λ_i such that*

$$A = \sum_i \lambda_i\, |i\rangle\langle i|.$$

A map $A \in \mathbf{M}$ is **positive** if $\langle u|A|u\rangle \geq 0$ for all $|u\rangle \in H$. The set of positive maps on H is denoted by $\mathbf{Pos}(H)$. A linear operator $\mathbf{M}(H_1) \to \mathbf{M}(H_2)$ is *positive* if it can be restricted to a map $\mathbf{Pos}(H_1) \to \mathbf{Pos}(H_2)$.

Definition 2.2. *The **Löwner partial order** [Löw34] on $\mathbf{M}(H)$ is defined by*

$$A \leq B \iff B - A \in \mathbf{Pos}(H).$$

Tensor products Given two Hilbert spaces H_1 and H_2, we define their **tensor product** $H_1 \otimes H_2$ to be the vector space generated by all pairs $|u_1\rangle \otimes |u_2\rangle$ with the following identifications:

1. $(|u_1\rangle + |u_2\rangle) \otimes |v\rangle = |u_1\rangle \otimes |v\rangle + |u_2\rangle \otimes |v\rangle$
2. $|u\rangle \otimes (|v_1\rangle + |v_2\rangle) = |u\rangle \otimes |v_1\rangle + |u\rangle \otimes |v_2\rangle$
3. $(\lambda|u\rangle) \otimes |v\rangle = \lambda(|u\rangle \otimes |v\rangle) = |u\rangle \otimes (\lambda|v\rangle)$

The Dirac notation convention is to leave the $\otimes$ operator implicit, and even sometimes to merge tensor products into a single ket:

$$|u\rangle \otimes |v\rangle = |u\rangle|v\rangle = |uv\rangle$$

The space $H_1 \otimes H_2$ is also a Hilbert space when the inner product of $|u_1\rangle|u_2\rangle$ and $|v_1\rangle|v_2\rangle$ is defined to be

$$\langle u_1|\langle u_2||v_1\rangle|v_2\rangle = \langle u_1|v_1\rangle\langle u_2|v_2\rangle.$$

Given orthonormal bases $|i\rangle$ and $|j\rangle$ for H_1 and H_2 respectively, the set of vectors of the form $|ij\rangle$ is an orthonormal bases for $H_1 \otimes H_2$.

The tensor product $A_1 \otimes A_2$ of two maps $A_1 \in \hom(H_1, K_1)$ and $A_1 \in \hom(H_2, K_2)$ is a map $H_1 \otimes H_2 \to K_1 \otimes K_2$ defined by

$$A_1 \otimes A_2|u\rangle|v\rangle = (A_1|u\rangle) \otimes (A_2|v\rangle).$$

Given orthonormal bases $|i_1\rangle, |i_2\rangle, |j_1\rangle, |j_2\rangle$ of H_1, H_2, K_1 and K_2 respectively, the matrix representation of $A_1 \otimes A_2$ for bases $|i_1 i_2\rangle$ and $|j_1 j_2\rangle$ can be computed from the elements of the representation of A_1 and A_2:

$$a_{i_1 i_2 j_1 j_2} = \langle j_1 j_2| A_1 \otimes A_2 |i_1 i_2\rangle = \langle j_1|A_1|i_1\rangle\langle j_2|A_2|i_2\rangle = a_{i_1 j_1} a_{i_2 j_2}$$

A map $A \in \hom(H, K)$ can always be extended to a map

$$H_1 \otimes H \otimes H_2 \to H_1 \otimes K \otimes H_2,$$

namely $I_{H_1} \otimes A \otimes I_{H_2}$. We will often abuse the notation and omit the identity maps from such tensor products, denoting $I_{H_1} \otimes A \otimes I_{H_2}$ simply by A. When there could be ambiguity on which component A is acting, we use superscript labels to remove the ambiguity.

Trace and partial trace The **trace** $\mathrm{tr}(A)$ of $A \in \mathbf{M}(H)$ is defined as follows: take any orthonormal basis $|i\rangle$ of H, and put

$$\mathrm{tr}(A) = \sum_i \langle i|A|i\rangle.$$

This definition can be shown to be independent of the choice of basis. The trace operator tr is linear and *cyclic*, meaning that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

The **partial trace** operation tr^{H_2} takes elements in $\mathbf{M}(H_1 \otimes H_2)$ to elements of $\mathbf{M}(H_1)$. It is defined in a manner similar to the trace, but by summation over a basis $|i\rangle$ of H_2:

$$\mathrm{tr}^{H_2}(A) = \sum_i \langle i|A|i\rangle.$$

This is also independent of the choice of basis. Note that $|i\rangle$ and $\langle i|$ implicitly denote $I_{H_1} \otimes |i\rangle$ and $I_{H_1} \otimes \langle i|$.

A map $\mathcal{E}\colon \mathbf{M}(H_1) \to \mathbf{M}(H_2)$ is said to be **trace preserving** if $\mathrm{tr}\,(\mathcal{E}(A)) = \mathrm{tr}\,(A)$ for all $A \in \mathbf{M}(H_1)$. It is **trace non-increasing** if $\mathrm{tr}\,(\mathcal{E}(A)) \leq \mathrm{tr}\,(A)$ for all $A \in \mathbf{M}(H_1)$.

Using traces it is possible to define an inner product on $\mathbf{M}(H)$ using the formula

$$\langle A, B\rangle = \mathrm{tr}(A^\dagger B).$$

Since H is assumed to be finite-dimensional, this inner product automatically gives $\mathrm{M}(H)$ a Hilbert space structure.

A linear map $\mathcal{E}\colon \mathbf{M}(H_1) \to \mathbf{H_2}$ has an adjoint $\mathcal{E}^*$ with respect to this inner product which satisfies

$$\mathrm{tr}\,(A\mathcal{E}(B)) = \mathrm{tr}\,(\mathcal{E}^*(A)B)\,.$$

It is easy to show that $\mathcal{E}$ preserves traces if and only if $\mathcal{E}^*$ is **unital**, i.e. $\mathcal{E}^*(I) = I$.

2.1.2 Quantum mechanics

Basic postulates

Quantum mechanics is the physical theory build from the following four postulates.

I. Quantum systems A quantum system A is described by a separable Hilbert space H_A over the field of complex numbers. A **state** of A is a ray (one dimensional subspace) in H_A. Unless stated otherwise, we work with normalized vector representatives of states, that is to say that the ray spanned by $|\phi\rangle \in H_A$ with $\langle\phi|\phi\rangle = 1$ is identified with $|\phi\rangle$.

We work only with complex Hilbert spaces of finite dimension, all of which are separable and isomorphic to $\mathbb{C}^n$ for some n.

II. Evolution The evolution over time of a quantum system A is described by a unitary operator U on H_A: if the system starts in state $|\phi\rangle$, then after the evolution the system is in state $U|\phi\rangle$.

III. Measurement A **measurement** is the process by which information about the state of a quantum system A is obtained. There are many types of measurements used in quantum theory, but we assume that the most basic kind is described in what follows.

A *projective measurement* of the state of a quantum system A is a family of projection operators $\mathcal{P} = \{P_i \mid i = 1, \ldots, n\}$ on H_A such that:

1. $P_i P_j = \delta_{ij} P_i$,

2. $\sum_i P_i = I_{H_A}$.

If a measurement $\mathcal{P}$ is made on a system in state $|\phi\rangle$, the result i is observed with probability $\langle\phi|P_i|\phi\rangle$. Measuring the state of the system changes it; if the measurement result is i, the state after the measurement is the normalized projected vector

$$\frac{P_i|\phi\rangle}{\sqrt{\langle\phi|P_i|\phi\rangle}}.$$

IV. Compound systems The Hilbert space describing the quantum system obtained by combining two quantum systems A and B is $H_A \otimes H_B$.

Entanglement

The fact that the state space of a compound system AB is the tensor product $H_A \otimes H_B$ has important consequences which distinguish quantum systems from classical ones. The main distinguishing feature is the existence of **entangled states** which cannot be written as a tensor product $|u\rangle|v\rangle$ of two states $|u\rangle$ and $|v\rangle$.

To illustrate this, suppose a quantum system AB is in the following entangled state:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

If two independent measurements

$$\mathcal{P}_A = \left\{P_1^A, P_2^A\right\}, \quad \mathcal{P}_B = \left\{P_1^B, P_2^B\right\}$$

are sequentially performed on each component, the results obtained are correlated. Indeed, the possible results for the first measurement are $i = 1$ or 2, with probability

$$\operatorname{tr}\left(P_i^A \otimes I \, |\beta_{00}\rangle\langle\beta_{00}|\right).$$

When the B subsystem is measured with $\mathcal{P}^B$, the result will be either $j = 1$ or 2, with probability

$$p_{ij} = \operatorname{tr}\left(\left(I_A \otimes P_j^B\right)\left(P_i^A \otimes I_B\right)|\beta_{00}\rangle\langle\beta_{00}|\right).$$

We can see that the distributions for i and j are not independent, since in general

$$p_i p_j = \operatorname{tr}\left(P_i^A \otimes I_B \, |\beta_{00}\rangle\langle\beta_{00}|\right)\operatorname{tr}\left(I_A \otimes P_i^B \, |\beta_{00}\rangle\langle\beta_{00}|\right) \neq p_{ij},$$

where $p_i = \sum_j p_{ij}$ and $p_j = \sum_i p_{ij}$. In the case where the projectors P_i^A P_j^B are the projectors onto the canonical basis $|i\rangle\langle i|$ $|j\rangle\langle j|$, the joint probability distribution is $p_{ij} = 1$ if $i = j$ and 0 otherwise. The meaning of this is that if the system A is measured in the canonical basis and i is observed, someone measuring the system B with knowledge of the result of the measurement at A knows with certainty that the result will be $j = i$.

Mixed states

We introduce below the main formalism used to describe and manipulate quantum states about which there is only partial information.

An **ensemble** of quantum states is a finite set of states $\{|\phi_i\rangle\}$, $i \in I$, together with a corresponding set of probabilities p_i such that $\sum_i p_i = 1$. To any ensemble there is an associated operator $\rho = \sum_i p_i|\phi_i\rangle\langle\phi_i|$. This operator is always positive and has trace 1; these two conditions are the key to get the following mathematical description:

Definition 2.3. *A positive operator ρ is called a* ***density matrix*** *(or* ***density operator****) if* $\mathrm{tr}(\rho) = 1$ *and* ***subdensity matrix*** *(or* ***subdensity operator****) if* $\mathrm{tr}(\rho) \leq 1$.

We denote the set of all density matrices of a Hilbert space H by $\mathbf{D}(H)$, and the set of all subdensity matrices by $\mathbf{SD}(H)$. A simple consequence of the spectral decomposition theorem is that every density matrix can be decomposed as an ensemble, though not necessarily uniquely.

Another important way to think about density matrices is given by the following result, which is in fact a consequence of Gleason's result [Gle57]:

Proposition 2.4. *(Gleason's theorem) Let H be a finite dimensional Hilbert space with* $\dim(H) > 2$. *For every function* $p\colon \mathbf{P}(H) \to [0, 1]$ *such that*

1. $p(I) = 1$ *and*
2. $p(P_1 + P_2) = p(P_1) + P(P_2)$ *if* $P_1 \perp P_2$,

there is a density matrix ρ such that $p(P) = \mathrm{tr}(P\rho)$.

Finally, note that the restriction of the Löwner partial order to $\mathbf{SD}(H)$ is a *ω-directed complete poset* (every countable directed set has a least upped bound) with the zero map as minimum element [Sel04b].

Other types of measurements

The projective measurements used in the description of the quantum mechanics postulates is not the only way to describe quantum measurements. The other descriptions all involve the idea that measurements are performed by making a quantum system A interact with a measurement apparatus, which is just another quantum system B. Following this point of view, the measurement process takes place in the combined system AB, and cannot in general be described by a family of projectors on subspaces of H_A alone.

A **positive operator valued measure** (henceforth referred to as a *POVM*) on a Hilbert space H is a family of positive operators A_m such that

$$\sum_m A_m = I_H.$$

If the system is in state ρ, performing the POVM measurement A_m will yield result m with probability $p_m = \mathrm{tr}(A_m\rho)$. Unless the operators A_m are defined in some way that allows one to compute the state after the measurement yielded the result m (as it is the case below with generalised measurements and quantum interventions), there is no unique way to determine the state after the measurement has been performed.

Contrarily to the case of projective measurements, in a POVM measurement the maps A_m associated to measurement results are not necessarily pairwise orthogonal. This has the consequence that there can be POVM measurements with different measurement results than is possible with any projective measurement, since in the latter case orthogonality forces the number of different outcomes to be less than $\dim(H)$. This can be explained by the interpretation of POVM as being a projective measurement in an enlarged system. This interpretation is possible because Neumark's theorem [Neu43] implies that that any POVM can be seen as the restriction by partial trace of a projective measurement on a larger Hilbert space.

Another kind of measurement is called a **generalised measurement**. These are specified by giving a family of maps $M_m \colon H_A \to H_A$, indexed by the possible measurement results, and satisfying the condition

$$\sum_m M_m^\dagger M_m = I.$$

The probability of observing m if a generalised measurement is performed while the system is in state ρ is

$$p_m = \operatorname{tr}\left(M_m \rho M_m^\dagger\right),$$

and the measurement process leaves the system in the state

$$\frac{1}{p_m} M_m \rho M_m^\dagger.$$

Superoperators and interventions

It is natural to seek a description for the physical evolution of unknown quantum states as described by density matrices. This description must satisfy various conditions. First, it must be a map that sends subdensity matrices to subdensity matrices. Second, it must preserve convex combinations of density matrices, because we want these maps to preserve probability distributions. Finally, if the evolution map is applied to part of a larger system, and the rest of the system is left unchanged, then the resulting larger map for the whole system must still send density matrices to density matrices.

Definition 2.5. *A **superoperator** $\mathcal{E}$ is a positive linear map $\mathbf{M}(H_1) \to \mathbf{M}(H_2)$ such that*

1. *$\mathcal{E}$ is trace non-increasing,*
2. *$\mathcal{E}$ is **completely positive**: $\mathcal{E} \otimes I_{\mathbf{M}(H_3)}$ is positive for all Hilbert spaces H_3*

It can be verified that superoperators satisfy all the above requirements. Complete positivity is a necessary condition because there are maps that are positive but not completely positive; one can consider, for example, the transposition map

$$\mathrm{T}\colon \mathbf{M}\left(\mathbb{C}^2\right) \to \mathbf{M}\left(\mathbb{C}^2\right)$$

defined by $\mathcal{T}(|i\rangle\langle j|) = |j\rangle\langle i|$. If we extend $\mathcal{T}$ to $\mathcal{T} \otimes \mathcal{I}_{\mathbf{M}(\mathbb{C}^2)}$ and apply this extended operator to the positive matrix $\sum_{ij} |ii\rangle\langle jj| = \sum_{ij} |i\rangle\langle j| \otimes |i\rangle\langle j|$, we get

$$(\mathcal{T} \otimes \mathcal{I})\left(\sum_{ij} |i\rangle\langle j| \otimes |i\rangle\langle j|\right) = \sum_{ij} \mathcal{T}\left(|i\rangle\langle j|\right) \otimes |i\rangle\langle j| = \sum_{ij} |j\rangle|i\rangle \otimes |i\rangle|j\rangle,$$

or, in matrix form

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

which is clearly not positive.

Superoperators can be characterised in various useful ways. The first one is known as *Kraus decomposition*. The next result is an adaptation of Choi's theorem for completely positive maps [Cho75, Kra83].

Proposition 2.6. *For any superoperator $\mathcal{E}$: there is a set of matrices $\{E_i\}$ satisfying $\sum_i E_i^\dagger E_i \leq I$ such that*

$$\mathcal{E}(A) = \sum_i E_i A E_i^\dagger$$

*We call the matrices E_i the **elements** (also called the* Kraus elements*) of the decomposition of the superoperator.*

Note that the decomposition given in this proposition is not necessarily unique. For example, a simple computation shows that the elements

$$E_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad E_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and the projection maps onto the canonical basis $F_1 = |0\rangle\langle 0|$ and $F_2 = |1\rangle\langle 1|$ both define the same superoperator.

Example 2.7. The superoperator elements for some fundamental examples are as follows:

- Adding an ancilla $|\phi\rangle$: $I \otimes |\phi\rangle$

- A unitary transformation U: $\{U\}$
- A projective measurement (not necessarily complete) $\{P_i\}$: $\{P_i\}$
- Tracing out a subsystem with orthonormal basis $\{|i\rangle\}$: $\{I \otimes \langle i|\}$

Another useful characterisation of superoperators is a result showing how to decompose them into elementary operations. It says that every superoperator can be thought of as a sequence of operations consisting of adding an *ancilla* state to the starting space, then applying a unitary transformation, then performing a projective measurement and finally tracing out part of the system in the resulting state. Note that is it possible to learn something about the projective measurement result in the process. In that case the superoperator describing the projective measurement step has $\{P_i\}$ as elements, where $\sum P_i \leq I$, which entails that the superoperator is trace decreasing since

$$\mathrm{tr}\left(\sum_i P_i \rho P_i\right) = \sum_i \mathrm{tr}(P_i \rho) \leq \mathrm{tr}(\rho)$$

for any density matrix ρ. The following proposition is shown in [NC00].

Proposition 2.8. *Every superoperator $\mathcal{E}$: $\mathbf{M}(H_1) \to \mathbf{M}(H_2)$ can be decomposed into a sequence of ancilla-adding, unitary, projective measurement and partial trace superoperators.*

Intervention operators

Peres introduced in [Per00] a very general description of quantum measurements called **intervention operators**. The measurement process is conceived as a unitary interaction of a measurement apparatus with the quantum system to be measured, followed by a projective measurement on the combined system. Mathematically, Peres shows that this process is described by superoperators: if the system is initially in state ρ, then, after reading the result m with probability

$$p_m = \mathrm{tr}\left(\sum_i E_{mi} \rho E_{mi}^\dagger\right),$$

the system is left in state

$$\rho_m = \frac{\sum_i E_{mi} \rho E_{mi}^\dagger}{p_m}.$$

Note that p_m can be written as $\mathrm{tr}(A_m \rho)$ if we put

$$A_m = \sum_i E_{mi}^\dagger E_{mi}.$$

In general, we define a **quantum intervention** to be a family of superoperators

$$\mathcal{E}_m \colon \mathrm{SD}(H_A) \to \mathrm{SD}(H_{B_m})$$

indexed by the possible measurement outcomes m, such that $\sum_m p_m = 1$ for all ρ. Note that the output space H_{B_m} may depend on the measurement outcome; this feature makes quantum interventions more general than superoperators. Since we must have $\sum_m p_m = 1$, $\mathcal{E}_m$ must satisfy the following condition:

$$\sum_m \operatorname{tr}(\mathcal{E}_m(\rho)) = \sum_m \operatorname{tr}\left(\sum_i E_{mi}\rho E_{mi}^\dagger\right) = \operatorname{tr}\left(\sum_m A_m \rho\right) = 1,$$

and since this must hold for all ρ, this is equivalent to asking that $\sum_m A_m = I$, i.e. when the matrices A_m are the components of a POVM.

Note that quantum interventions are different from *quantum instruments* [DL70], another similar generalisation of superoperators, because the output state of a quantum instrument is in a fixed Hilbert space while in a quantum intervention the output space depends on the measurement result.

2.1.3 Quantum computation

The field of quantum informatics originates in the 1970's, when the first quantum information theory results were proved. These provided some insight into the power and limitation of the idea of using quantum states to carry information. [Hol73]. In the 1980's, the first formal model of quantum computation was introduced in the form of a universal quantum Turing machine [Deu85]. The most widely cited quantum algorithms were discovered in the 1990's: the Deutsch-Jozsa algorithm [DJ92], the Grover algorithm for searching an unsorted database [Gro96], and the Shor algorithm for factoring [Sho94]. All three are examples of algorithms using quantum resources in a clever way in order to perform tasks more efficiently than can be done using classical algorithms.

Quantum algorithms

The basic token of information in quantum computing is called a **qbit**: it is a quantum state in the Hilbert space $\mathbb{C}^2$ which is taken as the simplest piece of information. The **computational basis** of the space of qbits is the canonical basis of $\mathbb{C}^2$ and its two vectors are conventionally denoted by $|0\rangle$ and $|1\rangle$. They are seen as an embedding of the classical bits $0, 1$ into the space of qbits.

The common way to describe quantum computation is *quantum circuits*, which can be described as follows. A set of qbits to operate upon is fixed, together with input and output subsets of these qbits. Each qbit that is not part of the input set is assumed to be prepared in some fixed initial state. The computation itself is represented as a sequence of unitary transformations on the set of all qbits associated to the algorithm. At the end, non-output qbits are measured or simply discarded (traced out). Some of the most important unitary operations used in quantum circuits are given in table 2.1.

Controlled operations Some steps in quantum algorithms involve operations that are conditionally applied. Let U be an operation on some set B of the qbits involved

Table 2.1 Basic quantum circuit operations

Pauli X	$X = \text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli Y	$Y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$
Pauli Z	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard	$\text{H} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

in a quantum algorithm, and A be one of the other qbits. We can define a new unitary operation $\wedge U$ ("control-U") on the subsystem AB by

$$\wedge U|0\rangle|u\rangle = |0\rangle|u\rangle \text{ and } \wedge U|1\rangle|u\rangle = |1\rangle U|u\rangle.$$

This means that U is applied on the B subsystem if and only if the A qbit is $|1\rangle$. An important example is the *controlled not* operation $\wedge X$ on two qbits. In the canonical basis, $\wedge X$ operates by flipping the value the second qbit if the control qbit is $|1\rangle$ and not changing the value of the second qbit if the control qbit is $|0\rangle$. This makes $\wedge X$ the main way to introduce control flow in quantum programs.

Universality Most quantum circuits are described using a limited number of unitary operations. A set of unitary transformations is said to be **universal** if it has the property that every unitary transformation can be approximated with arbitrary good precision by composition and tensors of unitary transformations from the given set. This means that it suffices to use only transformations taken from a universal set in order to be able to construct quantum circuits for all possible quantum algorithms.

2.2 Quantum programming languages

There are many proposed quantum programming languages. The reader can find a listing of most of the proposed quantum programming languages in the surveys of Selinger and Gay [Sel04a, Gay05]. We overview below three quantum programming languages, chosen to represent three important classes of languages: the low level measurement calculus, the functional quantum λ-calculus and finally the categorical abstract language of dagger compact closed categories, which abstracts some important aspects of quantum mechanics.

2.2.1 The Measurement Calculus

The *measurement calculus* is a quantum programming language developed around the idea that quantum circuits can always be described as a special kind of circuit where the only operations allowed are a unitary transformation on two qbits used to introduce entanglement, one qbit measurements, and one qbit unitary transformations picked from a limited set and which can be chosen according to previous measurement results. It was introduced by Danos et al. [DKP07] and is based on the one-way model of Raussendorf and Briegel [RB01], which introduced the idea of a measurement-based description of quantum computation. One of the main results obtained is a reduction procedure taking general measurement calculus programs, called *patterns*, to a standard form where the allowed operations are always applied in a specific order. This allows one to study parallelism in pattern computation, since the standard form, described below, reveals the structure of dependencies between measurements and quantum operations.

Patterns

For any $\alpha \in [0, 2\pi]$, we put

$$|+_\alpha\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\alpha}|1\rangle\right) \qquad |-_\alpha\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - e^{i\alpha}|1\rangle\right).$$

We denote $|+_0\rangle$ and $|-_0\rangle$ respectively by $|+\rangle$ and $|-\rangle$.

A **pattern type** is a finite set of qbits $\{H_i, i \in I\}$ with two subsets $\mathsf{in}, \mathsf{Out}$ of I. Let X_i, Y_i, Z_i be the usual Pauli operators on qbit i, and M_i^α be the projector $[+_\alpha]$ on qbit i.

The operations on the qbits of a pattern type are called *commands*. They are of three kinds:

i. Measurement The measurement commands allow one to measure a qbit with the projective measurement $\mathcal{P} = \{M^\alpha, I - M^\alpha\}$. All measurements are considered to be destructive. The measurement result of a such a measurement is called a **signal**. The signal values associated to the two projectors are respectively 1 if the first projector is applied, and 0 if it is the second projector that is applied. Two signals s and t can be combined using addition modulo 2 to get a new signal $s \oplus t$ (sum modulo 2).

ii. Correction One can change the state of an output qbit by applying the Pauli operators X or Z to it.

iii. Entanglement The entanglement command E_{ij} entangle the qbits i, j of the pattern type by applying to them the controlled-Z operator (denoted $\wedge Z$).

Signals are used to modify commands as follows:

$$[X_i]^s = \begin{cases} X & \text{if } s = 1 \\ I & \text{if } s = 0 \end{cases} \qquad [Z_i]^s = \begin{cases} Z & \text{if } s = 1 \\ I & \text{if } s = 0 \end{cases} \qquad \left[M_i^\alpha\right]^{s,t} = M^{(-1)^s\alpha + t\pi}$$

A **pattern** consists in a pattern type $(I, \mathsf{In}, \mathsf{Out})$ with a finite command sequence

$$E_1, \ldots, E_n$$

on it that satisfies the following three conditions:

1. no command depends on signals from qbits not yet measured,

2. no command is applied to a qbit after it has been measured,

3. no qbit in Out is measured, all other qbits are measured.

It is also assumed that all non-input qbits are initially in the $|+\rangle$ state. We use the convention that the signal associated to the qbit i is s_i.

Example 2.9. As shown in [AL04, DKP07], the Hadamard operation H can be implemented as the following pattern:

$$\left(\{1,2\},\{1\},\{2\},[X_2]^{s_1}M_1^0\,E_{12}\right).$$

Suppose that the qbit 1 is in state $|+\rangle$. Since all non-input qbits are assumed to be in state $|+\rangle$ at the beginning, the E_{12} command is applied to $|+\rangle|+\rangle$. The qbit array is left in state

$$\begin{aligned}
\wedge Z|+\rangle|+\rangle &= \wedge Z\,\frac{|0\rangle+|1\rangle}{\sqrt{2}}|+\rangle \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle|+\rangle+|1\rangle|-\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(\left(\frac{|+\rangle-|-\rangle}{2}\right)|+\rangle+\left(\frac{|+\rangle+|-\rangle}{2}\right)|-\rangle\right) \\
&= |+\rangle\frac{|+\rangle+|-\rangle}{2}+|-\rangle\frac{|+\rangle-|-\rangle}{2} \\
&= |+\rangle|0\rangle+|-\rangle|1\rangle
\end{aligned}$$

The first qbit is then measured in the $\{|+\rangle,|-\rangle\}$ basis. After the measurement, the array is left either in the state $|+\rangle|0\rangle$ or the $|-\rangle|1\rangle$ state, and the signal s_1 is respectively set to 1 or 0. The correction command X_2 is then applied conditionally according to the value of s_1, and the second qbit is left in state $|1\rangle = H|+\rangle$ as required. A similar computation shows that when the qbit 1 is in the $|-\rangle$ state at the beginning, qbit 2 ends in state $|0\rangle = H|-\rangle$.

Example 2.10. The controlled-not operation $\wedge X$ is implemented as the pattern

$$[X_4]^{s_3}[Z_4]^{s_2}[Z_1]^{s_2}M_3^0M_2^0E_{34}E_{23}\mathsf{E}_{13}$$

Example 2.11. The following is a pattern implementing teleportation [BBC$^+$93] (i.e. the identity function from one qbit to another): on the qbits labelled 1, 2 and 3,

$$[X_3]^{s_2}[Z_3]^{s_1}M_2^0M_1^0E_{23}E_{12}$$

2.2.2 Quantum λ-calculus

The λ-calculus is a formal language that was introduced in the 1930s by Church and Kleene and also studied in an equivalent form (combinatory logic) by Curry. As a programming language, one of its main distinguishing features is that it is a *higher-order* language in which functions can take other functions as arguments. Many variants of the λ-calculus have been studied : with control constructions, with recursion operators, with probabilistic choices, with stores, etc. The study of λ-calculus semantics has led

Table 2.2 Quantum λ-calculus subtyping rules

$$\frac{}{\alpha <: \alpha} \qquad \frac{}{X <: X} \qquad \frac{}{\top <: \top} \qquad \frac{A <: B}{!A <: B} \qquad \frac{!A <: B}{!A <: !B}$$

$$\frac{A_1 <: B_1 \qquad A_2 <: B_2}{A_1 \otimes A_2 <: B_1 \otimes B_2} \qquad \frac{A_2 <: A_1 \qquad B_1 <: B_2}{A_1 \multimap B_1 <: A_2 \multimap B_2}$$

where α is a constant type, X is a type variable.

to the development of many rich fields of informatics such as domain theory and game semantics.

A first adaptation of the λ-calculus to quantum computing was proposed by Maymin in the 90's [May96, May97]. Another important contribution was made later by van Tonder [vT04]; it emphasised for the first time the connection between the no-cloning theorem and the necessity to use quantum variables linearly (without duplication) in quantum λ-calculi. In this thesis, we take as a starting point Selinger and Valiron's proposal which was first defined in Valiron's master's thesis [Val04] based on earlier work by Selinger [Sel04b]. We review in what follows the more recent version found in [SV06a].

Syntax

The quantum λ-calculus is designed around the idea of *classical control with quantum store*: quantum algorithms are described in a setting where a classical computing device is allowed to operate on the state of a quantum register using unitary transformations and quantum measurements.

Following this view, the *terms* of the quantum λ-calculus are defined as follows:

$$\begin{aligned} M, N, P ::= {} & x \mid MN \mid \lambda x.M \mid \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{meas} \mid \\ & \mathsf{new} \mid U \mid * \mid \langle M, N \rangle \mid \mathsf{let}\, \langle x, y \rangle = M\, \mathsf{in}\, N \end{aligned}$$

where U ranges over unitary transformations and x over variables. The meas constant is the function which measure a qbit to return the measurement result. The new constant is used to create a new qbit in one of the computational basis state. The variable x is considered bound in $\lambda x.\, M$. The set of free variables of a term M is denoted by $\mathrm{FV}(M)$. We use $M[N/x]$ to denote substitution in M of a term N for every occurrence of $x \in \mathrm{FV}(M)$ when no free variable of N becomes bound. We identify $\lambda x.\, M$ and $\lambda y.\, M[y/x]$.

The *types* of the quantum λ-calculus are defined by:

$$A, B ::= \ \mathrm{bool} \mid \mathrm{qbit} \mid X \mid !A \mid A \multimap B \mid \top \mid A \otimes B.$$

The type system also involves a subtyping relation defined by the rules given in table 2.2.

Table 2.3 Quantum λ-calculus typing rules

$$\frac{A <: B}{\Gamma, x\colon A \vdash x\colon B} \qquad \frac{A_c <: B}{\Gamma \vdash c\colon B} \quad (A_c \text{ is the type of the constant } c)$$

$$\frac{\Gamma_1, !\Delta \vdash P\colon \text{bool} \qquad \Gamma_2, !\Delta \vdash M\colon A \qquad \Gamma_2, !\Delta \vdash N\colon B}{\Gamma_1, \Gamma_2, !\Delta \vdash \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N : A}$$

$$\frac{\Gamma_1, !\Delta \vdash M\colon A \multimap B \qquad \Gamma_2, !\Delta \vdash N\colon A}{\Gamma_1, \Gamma_2, !\Delta \vdash MN\colon B}$$

$$\frac{\Gamma, x\colon A \vdash M\colon B}{\Gamma \vdash \lambda x.M\colon A \multimap B} \qquad \frac{\Gamma, !\Delta, x\colon A \vdash M\colon B}{\Gamma, !\Delta, \vdash \lambda x.M\colon !^{n+1}(A \multimap B)} \; \mathsf{FV}(M) \cap |\Gamma| = \emptyset$$

$$\frac{\Gamma_1, !\Delta \vdash M_1\colon !^n A_1 \qquad \Gamma_2, !\Delta \vdash M_2\colon !^n A_2}{\Gamma_1, \Gamma_2, !\Delta \vdash \langle M_1, M_2\rangle\colon !^n\,(A_1 \otimes A_2)} \qquad \frac{}{\Gamma \vdash *\colon !^n\top}$$

$$\frac{\Gamma_1, !\Delta \vdash M\colon !^n(A_1 \otimes A_2) \qquad \Gamma_2, !\Delta, x_1\colon !^n A_1, x_2\colon !^n A_2 \vdash N\colon A}{\Gamma_1, \Gamma_2, !\Delta, \vdash \mathsf{let}\, \langle x_1, x_2\rangle = M\, \mathsf{in}\, N\colon A}$$

The types of constants are defined as follows:

$$\mathsf{meas}\colon !(\text{qbit} \multimap !\text{bool}) \qquad \mathsf{new}\colon !(\text{bool} \multimap \text{qbit})$$
$$\mathsf{true}, \mathsf{false}\colon \text{bool} \qquad U\colon !(\text{qbit}^n \multimap \text{qbit}^n)$$

A **context** Γ is a function assigning types to variables taken from some finite set, which is denoted using the usual notation $x_1\colon T_1, \ldots, x_n\colon T_n$. The domain of Γ is denoted by $|\Gamma|$. It is convenient to use the notation $!\Gamma$ for $x_1\colon !T_1, \ldots, x_n\colon !T_n$.

A typing judgement is a triple of the form $\Gamma \vdash M\colon T$, where Γ is a context, M is a term and T is a type. Valid typing judgements are those derived using the typing rules are given in table 2.3.

Note that these rules forbid duplication of unknown quantum data, since it is not possible to derive a typing judgement of the form $x\colon \text{qbit} \vdash x \otimes x\colon \text{qbit}$.

Operational semantics

We describe next how quantum programs described as λ-calculus terms are executed. This is done by giving rules telling how to reduce terms to simpler forms of base types. The reduction relation needs to be probabilistic to be able to deal with measurement operations. Furthermore, we need to take into account the state of the quantum register at each reduction step. To see that this is necessary, consider the term

$$\mathsf{if}\ (\mathsf{meas}\,(U\,(\mathsf{new}\,0)))\ \mathsf{then}\, M\, \mathsf{else}\, N,$$

which intuitively should reduce to M or N with different probability distributions depending on the state $U|0\rangle$. The state is modified by the measurement action, and any further reduction of M or N should be done using this modified register. To formalise this, we associate a qbit to each free variable of a term as follows: a **program state** is a triple $[Q, M, L]$ where

Table 2.4 Quantum λ-calculus reduction rules

$$\frac{}{[Q,(\lambda x.\,M)V] \downarrow [Q, M[V/x]]} \qquad \frac{[Q,N] \downarrow^p [Q',N']}{[Q,MN] \downarrow^p [Q',MN']} \qquad \frac{[Q,M] \downarrow^p [Q',M']}{[Q,MV] \downarrow^p [Q',M'V]}$$

$$\frac{[Q,M_1] \downarrow^p [Q',M_1']}{[Q,\langle M_1,M_2\rangle] \downarrow^p [Q',\langle M_1',M_2\rangle]} \qquad \frac{[Q,M_2] \downarrow^p [Q',M_2']}{[Q,\langle V_1,M_2\rangle] \downarrow^p [Q',\langle V_1,M_2'\rangle]}$$

$$\frac{}{[Q,\text{if } 0 \text{ then } M \text{ else } N] \downarrow [Q,N]} \qquad \frac{}{[Q,\text{if } 1 \text{ then } M \text{ else } N] \downarrow [Q,M]}$$

$$\frac{[Q,P] \downarrow^p [Q',P']}{[Q,\text{if } P \text{ then } M \text{ else } N] \downarrow^p [Q,\text{if } P' \text{ then } M \text{ else } N]}$$

$$\frac{}{[Q,U\langle p_1,\dots,p_n\rangle] \downarrow [UQ,\langle p_1,\dots,p_n\rangle]}$$

$$\frac{}{[Q,\text{meas } q_i] \downarrow^{\|[0]^iQ\|} [[0]^iQ/\|[0]^iQ\|,0]} \qquad \frac{}{[Q,\text{meas } q_i] \downarrow^{\|[1]^iQ\|} [[1]^iQ/\|[1]^iQ\|,1]}$$

$$\frac{}{[Q,\text{new } 0] \downarrow [Q \otimes |0\rangle,0]} \qquad \frac{}{[Q,\text{new } 1] \downarrow [Q \otimes |1\rangle,1]}$$

$$\frac{[Q,M] \downarrow^p [Q',M']}{[Q,\text{let } \langle x_1,x_2\rangle = M \text{ in } N] \downarrow^p [Q',\text{let } \langle x_1,x_2\rangle = M' \text{ in } N]}$$

$$\frac{}{[Q,\text{let } \langle x_1,x_2\rangle = \langle V_1,V_2\rangle \text{ in } N] \downarrow [Q',N[V_1/x_1,V_2/x_2]]}$$

1. Q a state in the Hilbert space $H = (\mathbb{C}^2)^{\otimes n}$ for n qbits, $n \geq 0$,
2. M is a term,
3. L is a partial function assigning variables of M to qbits of Q that is defined on all free variables,
4. quantum data is used linearly in M, i.e. no variable of type qbit is used more than once in M.

We can simplify the notation of program states by labeling the variables with qbit indexes, so that we can denote program states by pairs $[Q, M]$.

The operational semantics of the quantum λ-calculus is given by a small-step probabilistic reduction relation described in table 2.4; a call-by-value strategy is adopted by Selinger and Valiron. An important observation is that the value to which a given term is reduced depends on the reduction strategy chosen. This happens in all languages with operations that have side effects, like quantum measurements operations. There is nothing special in the quantum case in this regard. For example, assuming that x is a classical integer store holding the value 1, the term $\langle x := x + 1, x\rangle$ will reduce in a classical language to either $\langle 2, 2\rangle$ or $\langle 2, 1\rangle$ depending on which component is reduced first.

2.2.3 Categorical quantum mechanics

Another approach to the problem of providing a structured description of quantum computation is *categorical quantum mechanics*. This consists in using concepts from category theory (which we review briefly below) to create an abstract description of quantum mechanics where it is possible to express quantum algorithms and protocols. Abstraction allows one to study other models of quantum mechanics than the usual Hilbert space model. The proposed abstract categorical language can be interpreted using mathematical objects other than complex Hilbert spaces, such as sets and relations, and in turn any protocol or algorithm described in the abstract language can be interpreted using these objects.

Categories

We begin this review of categorical quantum mechanics by giving a brief overview of category theory. A more complete account of category theory can be found in [Mac71, BW99]. Note that the concepts of category theory are also used in game semantics and in programming language theory in general. The concepts described in what follows will be used throughout this thesis.

Category theory can be described as a theory of *structures*, where, in contrast to model theory where sets and relations are used to describe them, the focus is on the structure-preserving maps. Instead of defining a particular structure as a set equipped with various relations (including operations, functions, distinguished elements, etc.) which satisfy certain conditions or axioms, we define the class of such structures by imposing certain conditions on the maps between them. Category theorists assume there is always a minimum amount of relations between these maps to be able to express more complex constructions. Namely, there should be a notion of *composition* of two maps and each structure should have an associated *identity map*:

Definition 2.12. *A* ***category*** $\mathbf{C}$ *is a structure consisting of*

1. *a family of* ***objects*** $\mathrm{Ob}(\mathbf{C})$*,*
2. *a family of* ***morphisms*** $\mathrm{Mor}(\mathbf{C})$
3. *two mappings* $\mathrm{Dom}_{\mathbf{C}}$ *and* $\mathrm{Codom}_{\mathbf{C}}$ *from morphisms to objects,*
4. *for each object* $X \in \mathrm{Ob}(\mathbf{C})$ *a morphism* 1_X *with* $\mathrm{Dom}(1_X) = \mathrm{Codom}(1_X) = X$*,*
5. *a composition operation* $\circ$ *which takes two morphisms* f *and* g *with* $\mathrm{Dom}(g) = \mathrm{Codom}(f)$ *to a morphism* $g \circ f$ *with* $\mathrm{Dom}(g \circ f) = \mathrm{Dom}(f)$ *and* $\mathrm{Codom}(g \circ f) = \mathrm{Codom}(g)$.

The composition operator is usually left implicit, writing gf instead of $g \circ f$. We also use the notation $f; g$ for $g \circ f$. Equations involving morphisms in a category are usually represented as **diagrams** where objects are nodes and morphisms are arrows.

For example, the composition of two morphisms can be illustrated in this way:

$$\begin{tikzcd}
X \arrow[rr, "gf"] \arrow[dr, "f"'] & & Z \\
& Y \arrow[ur, "g"'] &
\end{tikzcd}$$

Some important ideas about structures can be expressed using category theory. For example, an **isomorphism** between two objects X and Y is a morphism $f\colon X \to Y$ for which there is another morphism f^{-1} such that $ff^{-1} = 1_Y$ and $f^{-1}f = 1_X$. The important point about this simple definition is that it does not use any knowledge about the internal structure of X and Y, it uses only the morphisms between the two objects in order to tell if they are isomorphic or not.

The structure-preserving maps between categories are called **functors**. A functor F from $\mathbb{C}$ to $\mathbb{D}$ is a pair of maps, one sending objects of $\mathbb{C}$ to objects of $\mathbb{D}$, and one sending morphisms $\hom_{\mathbf{C}}(X, Y)$ to morphisms $\hom_{\mathbf{D}}(F(X), F(Y))$ (we usually denote both by F since the argument type removes any ambiguity); these maps must satisfy the following conditions:

1. $F(1_X) = 1_{F(X)}$
2. if gf is defined, then $F(gf) = F(g)F(f)$.

A **contravariant functor** F is defined as a functor, but with $F(f)\colon F(Y) \to F(X)$ for $f\colon X \to Y$, i.e. as a functor that "reverses the arrows".

A **natural transformation** α between functors $F, G\colon \mathbf{C}, \mathbf{D}$ is a family of morphisms $\alpha_X\colon F(X) \to G(X)$ indexed by the objects of $\mathbf{C}$ such that for all $f\colon X \to Y$

$$\begin{tikzcd}
F(X) \arrow[r, "\alpha_X"] \arrow[d, "F(f)"'] & G(X) \arrow[d, "G(f)"] \\
F(Y) \arrow[r, "\alpha_Y"'] & G(Y)
\end{tikzcd}$$

A **product** of two objects A, B of a category $\mathbb{C}$, if it exists, is an object $A \times B$ with two **projection morphisms** p_A and p_B such that for all objects C with a pair of morphisms f_A, f_B there is a unique paring morphism $\langle f_A, f_B\rangle$ such that

$$\begin{tikzcd}
& A \times B \arrow[dl, "p_B"'] \arrow[dr, "p_A"] & \\
A & & B \\
& C \arrow[ul, "f_A"] \arrow[ur, "f_B"'] \arrow[uu, "\langle f_A, f_B\rangle"] &
\end{tikzcd}$$

All products are isomorphic in the sense defined above. We usually choose one representative that we call the product of A and B. Note that fixing B in $A \times B$ defines a functor

$$- \times B\colon \mathbb{C} \to \mathbb{C}.$$

An object T is said to be **terminal** in a category $\mathbb{C}$ if for all objects X there is a unique morphism $t_X \colon X \to T$. If $\mathbb{C}$ has a terminal object, it is unique up to isomorphism.

A **Cartesian category** $\mathbb{C}$ is a category with a terminal object and products. A Cartesian category is said to be **closed** if in addition there is a functor

$$- \Rightarrow B \colon \mathbb{C} \to \mathbb{C}$$

such that there is a bijection (in the category of sets)

$$\Lambda \colon \hom(A \times B, C) \to \hom(A, B \Rightarrow C)$$

which is natural both in A and C. Note that this is equivalent to saying that

$$\frac{(A \times B) \xrightarrow{f \times \mathrm{id}_B} (A' \times B) \xrightarrow{g} C}{A \xrightarrow{f} A' \xrightarrow{\Lambda(g)} B \Rightarrow C}$$

for all morphisms f and g.

Cartesian closed categories have a close relationship to logic and λ-calculi. This relation can be summarised in "slogan" form :

Categories	**Logic**	**λ-calculi**
Objects	Propositions	Types
Morphisms	Proofs	Terms
Composition	Cut-elimination	β-reduction

This is known as the *Curry-Howard-Lambek correspondence*, and is the cornerstone of the applications of category theory in computer science. A detailed explanation of this correspondence can be found in [LS86].

The last important general categorical concept necessary to abstract quantum mechanics categorically is symmetric monoidal categories. These are categories equipped with an extra tensor operation on objects.

Definition 2.13. *A* ***monoidal category*** $(\mathbf{C}, \otimes, I, \alpha, \rho, \lambda,)$ *is a category* $\mathbf{C}$ *equipped with a* ***tensor*** *bifunctor* $\otimes \colon \mathbf{C} \times \mathbf{C} \to \mathbf{C}$*, a distinguished object I and natural isomorphisms:*

$$\alpha_{XYZ} \colon (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z) \quad \textit{(associativity)}$$
$$\lambda_X \colon X \otimes I \to X \quad \textit{(left identity)} \qquad \rho_X \colon I \otimes X \to X \quad \textit{(right identity)}$$

such that the following diagrams commute for all objects A, B, C and D.

$$\begin{tikzcd}
((A \otimes B) \otimes C) \otimes D \arrow[r, "\alpha_{A,B,C\otimes D}"] \arrow[d, "\alpha_{A\otimes B,C,D}"'] & (A \otimes (B \otimes C)) \otimes D \arrow[r, "\alpha_{A,B\otimes C,D}"] & A \otimes ((B \otimes C) \otimes D) \arrow[d, "\mathrm{id}_A \otimes \alpha_{B,C,D}"] \\
(A \otimes B) \otimes (C \otimes D) \arrow[rr, "\alpha_{A,B,C\otimes D}"'] & & A \otimes (B \otimes (C \otimes D))
\end{tikzcd}$$

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha_{A,I,B}"] \arrow[dr, "\rho_A \otimes \mathrm{id}_B"'] & & A \otimes (I \otimes B) \arrow[dl, "\mathrm{id}_A \otimes \lambda_B"] \\
& A \otimes B &
\end{tikzcd}$$

The two conditions imposed on the natural isomorphism associated to monoidal categories are called **coherence conditions**; they imply that all diagrams constructed from identity morphisms, α, ρ and λ by composition and tensors are commutative.

Definition 2.14. *A **symmetric monoidal category** is a monoidal category* $\mathbf{C}$ *equipped with a **symmetry natural isomorphism*** $\sigma_{XY} \colon X \otimes Y \to Y \otimes X$ *subject to the coherence condition*

$$\begin{tikzcd}
X \otimes I \arrow[rr, "\sigma_{XI}"] \arrow[dr, "\rho_X"'] & & I \otimes X \arrow[dl, "\lambda_X"] \\
& X &
\end{tikzcd}$$

A symmetric monoidal category is **closed** when there is a bijection

$$\Lambda \colon \ \mathrm{hom}(A \otimes B, C) \to \mathrm{hom}(A, B \multimap C)$$

which is natural in A and B. This is similar to the definition of Cartesian closed categories. Note that we use the notation $B \multimap C$, which is used in linear logic to denote linear implication [Gir87], instead of $B \Rightarrow C$ which is used to denote intuitionistic implication. This usage is natural since symmetric monoidal closed categories and Cartesian closed categories are respectively models of the multiplicative fragment of linear logic and the conjontion and implication fragment of intuitionistic logic.

An important class of symmetric monoidal categories is obtained by considering symmetric monoidal categories equipped with a "dualisation functor".

Definition 2.15. *A **compact closed category** C is a symmetric monoidal category where for each object X there is a **dual** object X^*, and two natural transformations*

$$\nu_X \colon I \to X \otimes X^* \qquad \epsilon_X \colon X^* \otimes X \to I,$$

*called the **unit** and **counit**, that satisfy the following two coherence conditions:*

$$\begin{tikzcd}
X \arrow[r, "\rho_A^{-1}"] \arrow[d, equal] & I \otimes X \arrow[r, "\nu_X \otimes 1"] & (X \otimes X^*) \otimes X \arrow[d, "\alpha"] \\
X & X \otimes I \arrow[l, "\lambda_X"] & X \otimes (X^* \otimes X) \arrow[l, "1 \otimes \epsilon"]
\end{tikzcd}$$

$$\begin{tikzcd}
X^* \arrow[r, "\lambda_{X^*}^{-1}"] \arrow[d, equal] & X^* \otimes I \arrow[r, "1 \otimes \nu_X"] & X^* \otimes (X \otimes X^*) \arrow[d, "\alpha^{-1}"] \\
X^* & I \otimes X \arrow[l, "\rho_{X^*}"] & (X^* \otimes X) \otimes X^* \arrow[l, "1_{X^*} \otimes \epsilon_X"]
\end{tikzcd}$$

A compact closed category is always a symmetric monoidal closed category: we get this by defining $X \multimap Y$ by $X^* \otimes Y$. The "$*$" operator can easily be shown to be a contravariant functor satisfying $(X^*)^* \simeq X$.

2.2.4 Abstract quantum mechanics

The main idea in abstract quantum mechanics is to introduce an extra operation † in a symmetric monoidal category.

Definition 2.16. *A* ***dagger category*** *is a category equipped with an involutive and contravariant endofunctor † which is the identity on objects.*

This structure suffices to define abstractions of the following basic linear algebraic concepts:

Definition 2.17.

1. $f : X \to Y$ *is* unitary *if* $f^\dagger = f^{-1}$ *(i.e. if* $f^\dagger f^{-1} = f^{-1} f^\dagger = 1$*)* ,
2. $f : X \to Y$ *is* Hermitian *if* $f = f^\dagger$,
3. $f : X \to Y$ *is* positive *if there is another morphism g such that* $f = g^\dagger g$.

Definition 2.18. *A* ***dagger symmetric monoidal category*** *is a symmetric monoidal category with a dagger structure such that † preserves the symmetric monoidal structure coherently, i.e. with* $(X \otimes Y)^\dagger = X^\dagger \otimes Y^\dagger$ *and such that all the coherence isomorphisms are unitary morphisms.*

Definition 2.19. *A* ***dagger compact closed monoidal category*** *(also know as* strongly compact closed categories*) is a dagger symmetric monoidal category such that*

$$\begin{tikzcd}
I \arrow[r, "\epsilon^\dagger"] \arrow[dr, "\eta"'] & X \otimes X^* \arrow[d, "\sigma"] \\
& X^* \otimes X
\end{tikzcd}$$

commutes.

Abramsky and Coecke [AC04] have shown that dagger compact closed monoidal categories with biproducts provide enough structure to describe finite dimensional quantum mechanics abstractly. The main motivating example is the category of finite dimensional complex Hilbert spaces and linear maps, which satisfy all the dagger compact closed category axioms. Another interesting example is the category of sets and relations. It is shown above that important concepts like unitary and Hermitian maps can be defined in this abstract setup. It is in fact possible to define an abstract version of the Dirac notation in this categorical language. Scalars are defined as morphisms $s \colon I \to I$ – in the case of complex Hilbert spaces, it is easy to check that these maps are in bijection with the complex numbers. Scalars can be abstractly multiplied using tensor products and left and right identity natural isomorphisms. Kets are abstracted using the fact that elements of a complex Hilbert space H are in bijection with maps $\mathbb{C} \to H$, i.e. as abstract maps $|\phi\rangle : I \to X$ in a dagger compact closed category. With the dagger functor, one can also define abstract bras, inner products, orthogonality, bases, etc. In the presence of biproducts, it is also possible to define spectral decompositions and measurements. All this provides enough to express the postulates of quantum mechanics. Abramsky and Coecke showed that this is also enough to describe protocols such as teleportation, entanglement swapping and logic gate teleportation.

Abstract quantum mechanics has been further developed by Selinger in [SV06a, Sel07], where an abstract definition of completely positive maps is given, and in more recent work of Coecke, Pavlovic and Paquette [Coe07, CP06b, CP06a] where a new approach to abstract measurements is developed which allows the use of a graphical calculus integrating measurement operations without the difficulties of integrating biproducts into similar graphical calculi for general monoidal categories. Note that this last approach differs from ours: it is based on the idea that classical data can be copied and discarded while quantum data cannot. This is abstracted in the categorical notion of *Classical objects*, which are objects equipped with morphisms that abstract the properties of linear maps between Hilbert spaces that copy and discard vectors in a specified base. Using classical objects, it is possible to make measurement results, which are classical, interact with quantum data. Classical data is thus "encoded" as quantum data. In our approach, quantum data is represented as a special kind of probabilistic strategy.

2.3 Game semantics

Game semantics is the study of the interpretation of logical or programming languages using concepts associated to games like moves, plays and strategies. The central idea in game semantics is that a system can be described by the various ways that it can interact with its environment. These interactions can be described as sequences of actions, or "plays", in a game. Various types of systems can be described by imposing various constraints to these interactions. The roots of this approach are in the work made by logicians in the 50's and 60's to create interpretations of classical and intuitionistic logics in terms of games. In the 90's we saw a surge in applications of this general idea in the study of many logical systems and programming languages [Bla92, AJM94, AJ94, AJM94, HO00, AM99].

The aim of this thesis is to adapt game semantics ideas in a way appropriate to construct interpretations for quantum programming languages. In order to be able to represent quantum operations like measurements which give probability distributions on the measurement results, we need to work with probabilistic strategies. We end this chapter by a review of the basic definitions and facts of probabilistic game semantics, as presented in [DH02].

2.3.1 Arenas

The basic notion used in most of game semantics is the *arena*. Intuitively, it is the specification of the rules of interaction between the system and the environment where both agents can perform actions, or moves, taken from a specified set. The roles of the system and the environment are usually played by two players which are respectively named *Player* and *Opponent*. The choice of these names, widely used in the literature, do not indicate that the two agents are in competition and that one of them could win; we care only about the interactions between the two agents, and hence the name "arena" is used instead of the name "game" to indicate the absence of rewards or of winning conditions.

Definition 2.20. *An* ***arena*** *A is a triple $(M_A, \lambda_A, \vdash_A)$ where M_A is a set of* moves*, the function*

$$\lambda_A \colon M_A \to \{\mathrm{O}, \mathrm{P}\} \times \{\mathrm{Q}, \mathrm{A}\} \times \{\mathrm{I}, \mathrm{N}\}$$

is a labeling which assigns moves to the two players Opponent *and* Player*, and tells us which moves are* Questions *and which are* Answers*, and whether they are* Initial *or* Noninitial *moves, and finally $\vdash_A \subseteq M_A \times M_A$ is a relation, called the* ***enabling relation****, such that*

(A1) if $a \vdash_A b$, then $\lambda_A^{\mathrm{OP}}(a) \neq \lambda_A^{\mathrm{OP}}(b), \lambda_A^{\mathrm{QA}}(a) \neq \lambda_A^{\mathrm{QA}}(b)$,

(A2) if $\lambda_A^{\mathrm{IN}}(a) = \mathrm{I}$, then $\lambda_A(a) = \mathrm{OQI}$,

(A3) if $a \vdash b$ and $\lambda_A^{\mathrm{QA}}(b) = \mathrm{A}$ then $\lambda_A^{\mathrm{QA}}(a) = \mathrm{Q}$,

where the functions λ_A^{OP}, λ_A^{QA} and λ_A^{in} are λ_A composed with the projections on the sets $\{\mathrm{O}, \mathrm{P}\}$, $\{\mathrm{Q}, \mathrm{A}\}$ and $\{\mathrm{I}, \mathrm{N}\}$.

We use the convention that M_A^X, where X is some list of superscripts taken from the set of move labels $\{\mathrm{O}, \mathrm{P}, \mathrm{Q}, \mathrm{A}, \mathrm{I}, \mathrm{N}\}$ denote the set of moves labeled with these labels. Moves in an arena are thus of various types, and the constraints on the enabling relation $\vdash_A$ limit the possible interactions in the arena by limiting which moves can be made at a certain point given the past interactions. The condition (A1) forces that only Player moves enable Opponent moves and *vice versa*, (A2) asks for all initial moves to be questions by Opponent and finally (A3) says that answers can only be enabled by questions.

2.3.2 Plays and threads

Interactions between Opponent and Players are described by sequences of moves. A **play** in A is a sequence of moves $s \in M_A^*$. This does not take into account the enabling relation; we define a **justified play** to be a play where each occurrence of a non-initial move b has a pointer to a previous occurrence of a move a with $a \vdash_A b$. We finally need to enforce alternation of the two players. A **legal play** is a justified play where Opponent and Player alternate; we denote the set of legal plays in A by $\mathcal{L}_A$. Note that because all initial moves are Opponent moves, Opponent is always making the first move. The sets of odd and even length legal plays are respectively denoted by $\mathcal{L}_A^{\mathrm{odd}}$ and $\mathcal{L}_A^{\mathrm{even}}$.

Example 2.21. The **bool** arena is defined as follows:

1. $M_{\mathbf{bool}} = \{?, 0, 1\}$
2. $\lambda_{\mathbf{bool}}(?) = (\mathrm{O}, \mathrm{Q}, \mathrm{I})$ and $\lambda_{\mathbf{bool}}(0) = \lambda_{\mathbf{bool}}(1) = (\mathrm{P}, \mathrm{A}, \mathrm{N})$
3. $? \vdash_{\mathbf{bool}} 0, 1$

The legal plays in **bool** are those of the form

$$\varepsilon, ?b_0?b_1 \ldots ?b_n?, ?b_0?b_1 \ldots ?b_n,$$

where $b_k \in \{0, 1\}$. In these plays, each b_k is justified by the preceding occurrence of the ? move.

Example 2.22. The **empty arena** I is the arena with no moves at all. The only legal play in I is the empty play ε.

Nothing in the various restrictions imposed on justified and on legal plays forbid the case where there are many initial moves. It is possible for Opponent to start many interactions in parallel by making many initial moves. Formally, suppose $sa \in \mathcal{L}_A$. Starting from a and following the justification pointers will always lead to an occurrence of an initial move b, which we call the **hereditary justifier** of a in sa. We can see that every legal play will be partitioned in subplays, each one consisting of all occurrences of moves hereditarily justified by a given initial move. These subplays are called **threads**. The **current thread** of a legal play sa ending with an opponent move, denoted by $\lceil sa \rceil$, is the thread of sa where a occurs. If sa ends with a Player move, the current thread is then defined by $\lceil s \rceil a$. We want the current thread to be a legal play, so it is necessary to impose an extra condition on legal plays: a legal play s is **well-threaded** if for every subplay ta ending with a Player move, the justifier of a is in $\lceil t \rceil$. In a well-threaded play, player always plays in the last thread where Opponent played.

2.3.3 Operations on arenas

In game semantics, complex types obtained by type operations must correspond to arenas constructed by corresponding operations. Given arenas A, B, their **product** $A \odot B$ is defined by

- $M_{A \odot B} = M_A + M_B$ (disjoint union)
- $\lambda_{A \odot B} = [\lambda_A, \lambda_B]$ (copairing)
- $m \vdash_{A \odot B} n$ iff $m \vdash_A n$ or $m \vdash_B n$.

The product arena $A \odot B$ is intuitively understood as the arena where at each of Opponent's turns she can choose to play a move in either A or B, and where Player must answer in the last component where Opponent played.

The **linear arrow** operation $A \multimap B$ is defined similarly:

- $M_{A \multimap B} = M_A + M_B$

- $\lambda_{A\multimap B} = \left[\langle \overline{\lambda}_A^{\mathrm{OP}}, \lambda_A^{\mathrm{QA}}, \overline{\lambda}_A^{\mathrm{IN}} \rangle, \lambda_B\right]$
- $m \vdash_{A\multimap B} n$ iff $m \vdash_A n$ or $m \vdash_B n$ or $\lambda_B^{\mathrm{IN}}(n) = \lambda_A^{\mathrm{IN}}(m) = \mathrm{I}$.

where $\overline{\lambda}_A^{\mathrm{OP}}$ inverts the roles of the two players and $\overline{\lambda}_A^{\mathrm{IN}}$ makes all moves of A noninitial. This time, after Opponent makes an initial move in B, at each of his turns Player can choose to play either one of his moves in B or an Opponent move in A.

The empty arena I has important properties with respect to arena operations:

$$A \odot I = I \odot A = A$$
$$I \multimap A = A$$

2.3.4 Probabilistic strategies

Given a legal play s in an arena A, let $\mathrm{next}_A(s) = \{a \in M_A | sa \in \mathcal{L}_A\}$ be the set of all moves that can be legally made after the play s.

Definition 2.23. *A* ***probabilistic strategy*** *for Player is a function* $\sigma\colon \mathcal{L}_A^{\mathrm{even}} \to [0,1]$ *such that*

(S1) $\sigma(\epsilon) = 1$

(S2) $\sigma(s) \geq \sum_{b\in\mathrm{next}(sa)} \sigma(sab)$

The set of **traces** of a strategy σ in A is the set of even length legal plays which are assigned a non-zero probability by σ: it is denoted $\mathcal{T}_\sigma$. A strategy σ is **deterministic** if $\sigma(s) = 1$ for all $s \in \mathcal{T}_\sigma$.

It is possible to describe a probabilistic strategy σ in conditional form:

$$\sigma(b \mid sa) = \frac{\sigma(sab)}{\sigma(s)}$$

The probability $\sigma(b \mid sa)$ is the probability of Player choosing to play b after the play sa. We say that σ is **total** if for all $sa \in \mathcal{L}_A$ we have that

$$\sum_{b\in\mathrm{next}(sa)} \sigma(b \mid sa) = 1$$

Composition of strategies is the way interactions between parts of a program are encoded in game semantics. Given two strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$, we define a new strategy $\sigma;\tau\colon A \multimap C$ obtained by letting σ and τ "interact" on B. Before giving the definition of composition, it is necessary to formalise this notion of interaction.

The set of interactions for A, B, C is

$$\mathcal{I}_{A,B,C} = \{u \in (M_A + M_B + M_C)^* \mid u|_{AB} \in \mathcal{L}_{A\multimap B}, u|_{BC} \in \mathcal{L}_{B\multimap C}, u|_{AC} \in \mathcal{L}_{A\multimap C}\}$$

where $u|_{AB}$ is the sub sequence of u obtained by deleting the moves of C, and similarly for $u|_{BC}$. The case of $u|_{AC}$ is a bit different because deleting from u the moves of B and their associated pointers might leave the moves of A or C that are justified by B-moves without justifiers. In this case, we define the justifiers of $u|_{AC}$ to be as follows: a move a in C justified by a move b in B will be justified by the first move of either A or C we get to by following back the justification pointers from a in u.

The set of **witnesses** $\mathrm{wit}(s)$ of $s \in \mathcal{L}_{A \multimap C}$ in an interaction $\mathcal{I}_{A,B,C}$ is the set of interactions $u \in \mathcal{I}_{A,B,C}$ such that $u|_{AC} = s$.

The composition of two strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$ can now be defined as follows:

$$[\sigma;\tau](s) = \sum_{u \in \mathrm{wit}(s)} \sigma(u|_{AB})\tau(u|_{BC}).$$

The *identity strategy* (or so-called "copycat strategy") $\mathrm{id}_A\colon A \multimap A$ is neutral with respect to composition. A typical play is as follows:

$$\begin{tikzcd}
A_l \arrow[r, "\mathrm{id}_A"] & A_r \\
 & a_1 \\
a_1 & \\
a_2 & \\
 & a_2 \\
\vdots &
\end{tikzcd}$$

Formally, the identity strategy is defined as the deterministic strategy with trace

$$\mathcal{T}(1_A(s)) = \{s \in \mathcal{L}_{A_l \multimap A_r} \mid \forall s' \sqsubseteq^{\mathrm{even}} s.\ s'|_{A_l} = s'|_{A_r}\}.$$

Using all the structure defined so far it is possible to define the category **PStrat** of arenas and probabilistic strategies. Taking arenas as objects, a morphism $A \to B$ is a strategy in $A \multimap B$. Composition of strategies is the needed composition, with the identity strategies as identity morphisms. It is associative, and it is shown in [DH02] that probabilistic strategies are closed under composition.

This category is also symmetric monoidal. The operation $\odot$ is a tensor product, which acts on morphisms as follows. Given $\sigma\colon A \to C$ and $\tau\colon B \to D$ and $s \in \mathcal{L}^{\mathrm{even}}_{(A\odot B)\multimap(A'\odot B')}$, we set

$$[\sigma \odot \tau](s) = \sigma(s|_{A \multimap C})\tau(s|_{C \multimap D}).$$

All coherence isomorphisms are easily defined using variants of the copycat strategy.

2.3.5 Strategies and threads

Threads have an important role in game semantics as a way to characterize the strategies that encode programs with side-effects, like stores. This is achieved by forcing Player to use only the limited information available in the current thread instead of using all the information that can be extracted from the whole previous plays, including moves made in other threads.

A strategy σ is **well-threaded** if $\mathcal{T}_\sigma$ consists only of well-threaded plays. Note that this condition forces Player to answer in the last thread where Opponent played. Given two well-threaded plays $sab \in \mathcal{L}_A^{even}$ and $ta \in \mathcal{L}_A^{odd}$ with $\lceil sa \rceil = \lceil ta \rceil$, we define match$(sab, ta)$ to be the unique legal play tab with b justified as in $\lceil sa \rceil$. A well-threaded strategy σ is said to be **thread independent** if $sab \in \mathcal{T}_\sigma$, $t \in \mathcal{T}_\sigma$, $a \in \text{next}(t)$ and $\lceil sa \rceil = \lceil ta \rceil$ implies that

$$\frac{\sigma(sab)}{\sigma(s)} = \frac{\sigma((\text{match}(sab, ta))}{\sigma(t)}.$$

The meaning of this condition is that if Player plays according to σ, Player chooses his answers with probabilities that only depend on the current thread, i.e. $\sigma(b \mid sa) = \sigma(b \mid ta)$.

The diagonal strategy $\Delta_A : A \to A \odot A$ is defined as the deterministic strategy with trace set

$$\left\{ s \in \mathcal{L}^{even}_{A \multimap A_l \odot A_r} \mid \forall s' \sqsubseteq^{\text{even}} s.s'|_{A_l} \in \text{id}_{A_l} \wedge s'|_{A_r} \in \text{id}_{A_r} \right\}.$$

This is similar to the definition of the identity strategy: Δ instructs Player to use copying strategies between A and its two copies A_l and A_r. Possible conflicts in A are resolved by separating in different threads moves made according to the left or the right copy plays. There is also a unique strategy $\Diamond_A \multimap I$, namely the trivial strategy with trace $\{\varepsilon\}$.

The *pairing* of two thread independent strategies $\sigma \colon A \multimap B$ and $\tau \colon A \multimap C$ is defined by

$$\langle \sigma, \tau \rangle = \Delta_A ; \sigma \odot \tau$$

Thus when Player plays using the pair strategy $\langle \sigma, \tau \rangle$, he plays using σ after an initial move in B, and using τ after an initial move in C.

For each arena A, $(A, \Delta_A, \Diamond_A)$ is a *comonoid*, meaning that the following two diagrams commute:

$$\begin{tikzcd}
(A \odot A) \odot A \arrow[rr, "\alpha"] & & A \odot (A \odot A) \\
A \odot A \arrow[u, "\Delta_A \odot \text{id}_A"] & & A \odot A \arrow[u, "\text{id}_A \odot \Delta_A"'] \\
& A \arrow[ul, "\Delta_A"] \arrow[ur, "\Delta_A"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
A & A \odot I \arrow[l, "\rho_A"'] & A \odot A \arrow[l, "\text{id}_A \odot \Diamond_A"'] \arrow[r, "\Diamond_A \odot \text{id}_A"] & I \odot A \arrow[r, "\lambda_A"] & A \\
& & A \arrow[u, "\Delta_A"] \arrow[ull, equal] \arrow[urr, equal] & &
\end{tikzcd}$$

The following proposition is an important fact about thread independent strategies, proved in [Har99]:

Proposition 2.24. *A strategy $\sigma : A \multimap B$ is thread independent if and only if σ is a comonoid homomorphism for the comonoid, that is, if for all A and B the following* *commutes:*

$$\begin{tikzcd}
A \arrow[r, "\Delta_A"] \arrow[d, "\sigma"'] & A \odot A \arrow[d, "\sigma\odot\sigma"] \\
B \arrow[r, "\Delta_B"'] & B \odot B
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "\diamond_A"] \arrow[d, "\sigma"'] & I \\
B \arrow[ur, "\diamond_B"'] &
\end{tikzcd}$$

It is a known fact in category theory[Jac94] that the last proposition implies that the restriction of **PStrat** to thread independent strategies is a Cartesian closed category. This is based on the fact that we can use pairing as defined above to get products. Note that projections strategies are defined as copying strategies:

$$\begin{array}{lccccc}
\pi_A : & (A & \odot & B) & \multimap & A \\
 & & & & & a_1 \\
 & a_1 & & & & \\
 & a_2 & & & & \\
 & & & & & a_2 \\
 & & & \vdots & &
\end{array}
\qquad
\begin{array}{lccccc}
\pi_B : & A & \odot & B & \multimap & B \\
 & & & & & b_1 \\
 & & & b_1 & & \\
 & & & b_2 & & \\
 & & & & & b_2 \\
 & & & \vdots & &
\end{array}$$

Chapter 3

Quantum games and strategies

3.1 Arenas for quantum systems

The central objective of this thesis is to develop a new model for quantum programming languages by adapting the concepts of game semantics to quantum computation. The core of game semantics is the idea of identifying states of a system with the processes by which the environment gets information about the system. These processes are described as sequences of actions performed by the environment and the system. The most basic systems are described as simple "question-answer" interactions.

To adapt game semantics to quantum systems, it is necessary to identify the kind of actions that can be performed and the find appropriate restrictions on the interactions to make them compatible with quantum mechanics. We take the following extended form of the "slogan" correspondence describing game semantics to physical systems as a general guideline:

λ-calculi	**Games**	**Physics**
Types	Arenas	State spaces
Terms	Strategies	Dynamics
Reduction	Composition of strategies	Composition of dynamics

This guideline is formulated with general physical systems in mind, but here we focus on the case of quantum physical systems. Following the correspondence, a quantum state should be described by the way Player, incarnating the physical system, interacts with Opponent, which has the role of the environment, when they play in some appropriate arena. More specifically, if Opponent asks for information about the state of a quantum system, Player should answer with some information about this state. We need to specify the kind of interactions that they can have in this process.

A possibility is to identify knowledge of the state of a quantum system to knowledge of the density matrix ρ describing it, as it is done in some of the work on quantum knowledge theory [vdMP03]. In that case, a typical play in the arena describing a quantum system would be of the form $?\rho$. This conception of quantum knowledge is criticised in [DP05, D'H05], where it is argued that an agent may get knowledge about the state of a quantum system in the following ways:

1. if the agent prepares the system in a known state,
2. if the agent measures the state of the system,
3. if information about the state of the system is communicated to the agent.

We adopt this point of view and adapt it to the context of game semantics. We consider Opponent can only be given information about a quantum system under the control of Player, knowledge that Player can only get by measuring or preparing the system. This means that the possible answers to such a question must correspond to the measurement outcomes.

3.1.1 Quantum Games

The central new structure introduced this thesis is a quantum game which is a variant of the games used in the field of game semantics. There is an emerging field of research developing around the idea of *quantum games*. To understand how the work presented in this thesis differs from what is done in the quantum game field, we present an outline of the latter.

Quantum games with payoffs are the central object of study of *quantum game theory*. They are presented as generalisation of classical von Neumann games [MvN47, OR94]. A classical two player **von Neumann game** G can be described as a pair of set of strategies S_A and S_B, one for each player A and B, together with a payoff function

$$M\colon S_A \times S_B \to \mathbb{R} \times \mathbb{R}.$$

If both A and B chose their strategies s_A and s_B, we get an element of $S_A \times S_B$, and the associated tuple $M(s_A, s_B)$ gives the payoff of each player when they play using the chosen strategies. A **Nash equilibrium** of G is a pair of strategies (s_A, s_B) such that no player can improve his or her payoff by choosing another strategy if the other player keeps using the same strategy. The von Neumann theorem says that in zero-sum two players games (where the payoff of one player is opposite to the payoff of the other) there is always a Nash equilibrium of probabilistic strategies.

The focus of the research in quantum game theory is to study Nash equilibria when strategies are described as quantum states and chosen using quantum operations. A typical two player quantum game starts with some fixed state ρ in some complex Hilbert space $H_A \otimes H_B$, where A and B are two quantum systems associated to two players, and comes equipped with privileged orthonormal bases $|s_A\rangle$ and $|s_B\rangle$. The elements of these bases are thought as the possible strategies for A and B. Each player chooses a strategy using some quantum operations $\mathcal{E}_A$ and $\mathcal{E}_B$ on the associated quantum system. These operations are taken from a predefined set. Finally, the state $[\mathcal{E}_A \otimes \mathcal{E}_B](\rho)$ is measured with the projective measurement $\{|s_A s_B\rangle\langle s_A s_B|\}$; the value observed is then used to determine the payoff of each player.

This scheme can be seen as a generalisation of the classical von Neumann games, Consider a classical game G as above for the case of two players A and B. We suppose that the players choose strategies in S_A and S_B by applying some permutations π_A and π_B of S_A and S_B to some fixed strategies s_A and s_B. The chosen strategies are thus $\pi_A(s_A)$ and $\pi_B(s_B)$. Since the players can chose among all permutations, they can pick any strategies they want. Described in this way, the game G can be put in "quantum form" as follows. Let H_A and H_B be the Hilbert spaces spanned by S_A and S_B respectively (we assume there is a finite number of strategies for each player). The available strategies correspond to base vectors $|s_A\rangle \in H_A$ and $|s_B\rangle \in H_B$ and thus instead of using permutations, the players use the corresponding permutation matrices. This means that to select a strategy player A applies a permutation matrix M_A, which selects the strategy corresponding to the state $|s'_a\rangle = M_{s'_A}|s_A\rangle$. To determine the payoffs, one has to measure the state $\left(\pi_{s'_A} \otimes M_{B'}\right)|s_A s_B\rangle$ using the projectors onto the bases $|s_A\rangle, s_A \in S_A$ and $|s_B\rangle, s_B \in S_B$ and associate the payoff $M(s_A, s_B)$ when s_A and s_B are observed. To represent a probabilistic choice of strategy, one can use the quantum operations obtained by convex combinations of the permutation operations above, i.e. superoperators of the form

$$\mathcal{E}(\rho) = \sum_{s'_A} p_{s'_A} \pi_{s'_A} \rho \pi^{\dagger}_{s'_A},$$

where $p_{s_A} \in [0, 1]$ and $\sum_{s_A \in S_A} p_{s_A} = 1$. In general quantum games, the starting state is allowed to be an entangled state over $|\varphi\rangle$, the permutation matrices are allowed to be any quantum operation (in most of the literature unitary operations are used) and the measurement results can be any quantum measurements. Variants of known classical games, like the prisoner's dilemma, have been studied to determine their Nash equilibriums and compare them to the classical equilibriums. These are constructed using various schemes which allow using other quantum operations than the permutations by allowing the starting state to be an entangled state.

It was shown first in [Mey99, Mey00, EWL99] that there are simple games where there are new Nash equilibriums when the players use quantum strategies that are not present when they use probabilistic strategies. In fact, Mayer proved an analogue to von Neumann's theorem for quantum games. Recent work developed these ideas by investigating various quantum analogues of classical matrix games and using quantum games in quantum information and complexity theory [CHTW04, GW07]. While these results are interesting, they do not provide the framework needed to develop a quantum analogue of classical arenas as described in the first section of this chapter. This is because matrix games hide the detailed interactions occurring when the players use their chosen strategies. There is also a more profound conceptual problem with the approach used in quantum game theory: the choice of a strategy in a quantum game as described above is done by applying a quantum operation on a state. This is described as a "quantum strategy" where probabilistic choices are generalised, but, since quantum games can be seen as matrix games, these "quantum choices" amount in fact to classical probabilistic choices in a set of quantum operations. It is possible to make this a general principle: games can be played with quantum states and with the possibility of using quantum operations on these states, but the only way a player can make a decision about his or her next next move using information about a quantum state is by measuring it and using the measurement result classically. The quantum game concept developed in this thesis follows this principle.

This view of quantum games or interactions is a good way to understand a central difficulty encountered when studying higher-order languages : as pointed out by Selinger [Sel04b], reasonable attempts to find a closed monoidal category extending the category of superoperators where one can define interpretations of higher-order quantum programming languages fail. The above principle may help to understand why this is the case. Consider the core defining property of symmetric monoidal-closed categories: the adjunction

$$\frac{X \otimes Y \to Z}{X \to Y \multimap Z}.$$

In the categories used to define denotational semantics of classical programming languages, this adjunction property can be understood as saying that there is an object $Y \multimap Z$ that can be used to define a *parametrised family of terms* over some other object X, and that the parameter can always be also thought of as an argument in a term $X \otimes Y \to Z$. To apply the term corresponding to a certain parameter in X, one must first determine this parameter.

In the quantum case such a parametrisation should be a way to use a state to choose a quantum operation. If we follow the principle proposed above, this must be done by measuring the parameter state in X and using the measurement result to choose the quantum operation $Y \to Z$. If such a process describe an adjunct, has to correspond to a quantum operation $X \otimes Y \to Z$. If we consider the quantum parametrisation to be given by a family of superoperators $\mathcal{E}_m \colon \mathbf{SD}(Y) \to \mathbf{SD}(Z)$, with $\mathcal{E}_m(\rho) = \sum_k E_{mk} \rho E_{mk}^{\dagger}$ and indexed with the measurement results of a generalized measurement $\mathcal{M} = \{M_m\}$ over X, the resulting operation $X \otimes Y \to Z$ will map $\rho \in \mathbf{SD}(X \otimes Y)$ to

$$\sum_m (M_m \otimes E_{mk}) \rho \left(M_m^{\dagger} \otimes E_{mk}^{\dagger}\right).$$

Since not every superoperator $X \otimes Y \to Z$ can be written in this last form, we cannot hope to have the desired general correspondence. This can also be explained by the fact that in $X \otimes Y \to Z$ the parameter X and the argument Y can be entangled in the quantum case, while they are independent in $X \to Y \multimap Z$.

3.2 Arenas for isolated quantum systems

We proceed in what follows to the formalisation of the ideas described in section 3.1. We begin by considering simple isolated quantum systems described by a complex Hilbert space H. Starting from an arena as defined in section 2.3, we need to add H to the structure. We also need to add restrictions on the enabling relations so that the answers to a quantum question can be used as measurement outcomes.

Definition 3.1. *An **isolated quantum system arena** (IQSA) A is an arena $|A| = (M_A, \lambda_A, \vdash_A)$ together with an associated Hilbert space H_A such that for all $q \in M_A^{\mathrm{Q}}$ the number of enabled moves $|\{a \mid q \vdash_A a\}|$ is* $\dim(H_A)$.

The arena $|A|$ is called the **underlying arena** of the IQSA A. A **quantum play** in an IQSA A is a play in $|A|$ together with, for each occurrence of a quantum question q, a projective measurement $\mathcal{P}_q = \left\{P^q_a \mid q \vdash_A a\right\}$. We denote a quantum play as a regular play where we replace the quantum questions by their associated measurements, for example:

$$\mathcal{P}_{q_1} a_1 \mathcal{P}_{q_2} a_2 \ldots$$

If there are many occurrences of a question q, we refer to the nth occurrence with $q[n]$.

Given a quantum play s, the play of $|A|$ obtained by forgetting the associated quantum measurements is the **underlying play** $|s|$. For an IQSA A, the set of **legal quantum plays** is

$$\mathcal{L}_A = \{s \mid \text{ is a quantum play and } |s| \in \mathcal{L}_{|A|}\}.$$

Example 3.2. The IQSA **qbit** describing the possible states of a qbit is defined as follows. The underlying arena is

- $M^{QOI}_{\mathbf{qbit}} = \{?\}$, $M^{APN}_{\mathbf{qbit}} = \{0, 1\}$;
- $? \vdash 1, ? \vdash 0$.

and H_A is taken to be $\mathbb{C}^2$. Some possible quantum plays are

$$\begin{aligned}
s_1 &= \{[0]_0, [1]_1\}_?\\
s_2 &= \{[+]_0, [-]_1\}_? 1\\
s_3 &= \{[0]_0, [1]_1\}_? 0 \{[+]_0, [-]_1\}_? 1\\
s_4 &= \{0_0, I_1\}_?\, 1
\end{aligned}$$

Note that the underlying plays of quantum plays in **qbit** are plays of **bool**.

Example 3.3. Given any Hilbert space H we define a IQSA $[H]$ in a similar way as the definition of the IQSA **qbit**, but using in general $0, \ldots, \dim(H) - 1$ as possible answers. When $\dim(H) = 2$, we will also use the name **qbit** for the IQSA $[H]$. In the case of the trivial Hilbert space $H = 0$, we adopt the convention that $[0]$ is the empty arena I.

The special case $\mathbb{C}$ is important. In this IQSA, there is only one possible complete projective measurement $\mathcal{P} = \{I_0\}$. There is also only one possible answer, 0.

3.3 Quantum strategies for isolated quantum arenas

In the last section we identified the kind of actions occurring in the interaction between a quantum system and an environment extracting information from it. The next step is to identify the strategies that describe quantum states.

When a quantum system is in a certain state, for each quantum measurement there is a probability distribution over the measurement outcomes. We thus need to begin by providing a way to allow the possible interactions to describe this distribution.

Definition 3.4. *A probabilistic strategy σ in a IQSA A is defined as a function*

$$\sigma\colon \mathcal{L}^{\text{even}}_A \to [0, 1]$$

such that

1. $\sigma(\varepsilon) = 1$
2. $\sigma(s) \geq \sum_{b \in \mathrm{next}_A(sa)} \sigma(sab)$

A probabilistic strategy of a quantum arena A can be viewed as a probabilistic strategy for $|A|$ and *vice versa*.

We can now define a probabilistic strategy in the IQSA $[H]$ that corresponds to a given state ρ.

Definition 3.5. *Let ρ be a density matrix over H. The probabilistic strategy $[\rho]$ in $[H]$ associated to ρ is defined by*

1. $[\rho](\epsilon) = 1$
2. $[\rho](\mathcal{P}_{?[1]}m_1 \dots \mathcal{P}_{?[n]}m_n) = \mathrm{tr}\left(P^{?[n]}_{m_n} \dots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \dots P^{?[n]}_{m_n}\right)$

We need to check that $[\rho]$ is a probabilistic strategy. The first condition of the definition is automatically verified. The second condition is also satisfied: let $s\mathcal{P}_?m \in \mathcal{L}^{\mathrm{even}}_{[H]}$. We have that

$$\begin{aligned}
\sum_{m \in \mathrm{next}(s\mathcal{P}_?)} [\rho]\,(s\mathcal{P}_?m) &= \sum_{m\colon\, ? \vdash m} \mathrm{tr}\left(P^?_m P^{?[n-1]}_{m_{n-1}} \dots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \dots P^{?[n-1]}_{m_{n-1}} P^?_m\right) \\
&= \mathrm{tr}\left(\left(\sum_i P^?_i\right) P^{?[n-1]}_{m_{n-1}} \dots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \dots P^{?[n-1]}_{m_{n-1}}\right) \\
&= \mathrm{tr}\left(P^{?[n-1]}_{m_{n-1}} \dots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \dots P^{?[n-1]}_{m_{n-1}}\right) \\
&= [\rho](s).
\end{aligned}$$

The probability of Player answering m to the question $\mathcal{P}_?$ asked after a quantum play $s = \mathcal{P}_{?[1]}m_1 \dots \mathcal{P}_{?[n]}m_n$ is given by

$$[\rho](m \mid s\mathcal{P}_?) = \frac{[\rho]\,(s\mathcal{P}_?m)}{[\rho](s)} = \frac{\mathrm{tr}\left(P^?_m \rho_s P^?_m\right)}{\mathrm{tr}(\rho_s)}$$

where ρ_s is the subdensity matrix $P^{?[n]}_{m_n} \dots P^{?[1]}_{m_1} \rho_s P^{?[1]}_{m_1} \dots P^{?[n]}_{m_n}$.

This makes the strategies $[\rho]$ thread dependent. To see this, consider for example how the strategy $[|+\rangle\langle+|]$ evaluates on the two quantum plays:

$$\{[0]_0, [1]_1\}_?\, 0\, \{[0]_0, [1]_1\}_?\, 0$$
$$\{[0]_0, [1]_1\}_?\, 1\, \{[0]_0, [1]_1\}_?\, 0$$

The probability of answering 0 at the end of the first play is 1 while it is 0 in the second play. The strategy dictates the use of a different probability distribution in the second thread according to the answer given in the first one.

Note also that $[\rho]$ is total because $\mathrm{tr}(\rho) = 1$, and thus

$$\sum_m \sigma(m \mid s\mathcal{P}_?) = \sum_m \frac{\mathrm{tr}(P_m \rho_s)}{\mathrm{tr}(\rho_s)} = 1$$

Is there a density matrix ρ associated to a probabilistic strategy for a IQSA? It is not the case in general. Consider for example any probabilistic strategy σ in the IQSA H for which $\sigma(m \mid \mathcal{P}_?)$ is the uniform distribution with value $1/\dim(H)$ for all m. The existence of such examples is due to the fact that we do not impose any special restrictions on probabilistic strategies. We now define a restricted family of probabilistic strategies that correspond to quantum states.

Definition 3.6. *A probabilistic strategy for a IQSA H is called a* ***quantum strategy*** *if the following three conditions hold*

1. *if* $s\mathcal{P}_?m, s\mathcal{P}'_?m \in \mathcal{L}_{[H]}$ *and* $P^?_m = P'^?_m$*, then* $\sigma(m \mid s\mathcal{P}_?) = \sigma(m \mid s\mathcal{P}'_?)$
2. *for any three projective measurements* $\mathcal{P}_?$, $\mathcal{P}'_?$ *and* $\mathcal{P}''_?$ *with* $P'^?_{m_1} \perp P''^?_{m_2}$ *and* $P^?_m = P'^?_{m_1} + P''^?_{m_2}$ *we have that* $\sigma(m \mid s\mathcal{P}_?) = \sigma(m_1 \mid s\mathcal{P}'_?) + \sigma(i_2 \mid s\mathcal{P}''_?)$
3. *for every projective measurement* $\mathcal{P}_?$ *such that* $\sum_m P^?_m = I_H$ *we have* $\sigma(m \mid s\mathcal{P}_?) = 1$

Lemma 3.7. *Let* σ *be a total quantum strategy in* $[H]$*, with* $\dim(H) > 2$*,* s *be a quantum play. The function* $p\colon \mathrm{P}(H) \to [0, 1]$ *which sends* $P \in \mathrm{P}(H)$ *to*

$$p(P) = \sigma(m \mid s\mathcal{P}_?),$$

where $\mathcal{P}_?$ *is any complete projective measurement with* $P^?_m = P$*, satisfies the conditions of Gleason's theorem (proposition 2.4).*

Proof. Fix some quantum play s. By the first condition of the definition of quantum strategy, p is well-defined.

Let $P \perp Q$, $P, Q \in \mathrm{P}(H)$. By definition,

$$p(P + Q) = \sigma(m \mid s\mathcal{P}_?)$$

for some $\mathcal{P}_?$ with $P^?_m = P + Q$.

It is always possible to define a projective measurement $\mathcal{P}'_?$ by splitting $P^?_m$ into two. If the rank of $P + Q$ is 0 or 1, we simply take $\mathcal{P}'_? = \mathcal{P}_?$. If the rank of $P + Q$ is strictly greater that 1, set $m_1 = m$. In that case, there must be a move m_2 with $P^?_{m_2} = 0$. Let $\mathcal{P}'_?$ be defined by $P'^?_{m_1} = P$, $P'^?_{m_2} = Q$ and $P'^?_{m'} = P^?_{m'}$ for $m' \neq m_1, m_2$. It then follows from the second condition of the definition of quantum strategy that

$$p(P) + p(Q) = \sigma(m_1 \mid s\mathcal{P}'_?) + \sigma(m_2 \mid s\mathcal{P}'_?) = \sigma(m \mid s\mathcal{P}_?) = p(P + Q)$$

To evaluate $p(I)$, the last condition of the definition of quantum strategy gives us that for any $\mathcal{P}_?$ with $P^?_m = I$

$$p(I) = \sigma(m \mid s\mathcal{P}_?) = 1,$$

so p satisfies the conditions of Gleason's theorem. □

This lemma implies that when $\dim(H) > 2$, there is a quantum state ρ_s for each even length quantum play s in $[H]$.

Lemma 3.8. *For a quantum strategy σ in H, with* $\dim(H) > 2$*, we have that*

$$\sigma(m \mid s\mathcal{P}_?) = \operatorname{tr}\left(P^?_m \rho_s\right)$$

Since the defining conditions of quantum strategies do not impose any constraints on the relation between threads, these ρ_s need not be related to one another in any special way. The strategies $[\rho]$ satisfy the following extra condition.

Definition 3.9. *Suppose* $\dim(H) > 2$*. A total quantum strategy σ on H is said to be **physically realisable** if for all quantum plays $s\mathcal{P}_?m \in \sigma$ we have that*

$$\rho_{s\mathcal{P}m} = \frac{P^?_m \rho_s P^?_m}{\sigma\left(s\mathcal{P}_?m\right)}$$

where ρ_s is the density matrix associated to s by lemma 3.7.

Theorem 3.10. *For every total physically realisable quantum strategy σ on a IQSA* $[H]$*, with* $\dim(H) > 2$*, there is a density matrix ρ such that* $[\rho] = \sigma$*.*

Proof. For each play $s = \mathcal{P}_{?[1]}m_1 \ldots \mathcal{P}_{?[n]}m_n \in \mathcal{L}_{[H]}$, the strategy σ determines a density matrix ρ_s which satisfies

$$\sigma\left(m_{n+1} \mid s\mathcal{P}_{?[n+1]}\right) = \operatorname{tr}\left(P^{?[n+1]}_{m_{n+1}} \rho_s\right).$$

Taking $\rho = \rho_\varepsilon$, an easy induction on the length of s shows that

$$\rho_s = P^{?[n]}_{m_n} \cdots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \cdots P^{?[n]}_{m_n},$$

and thus that

$$\sigma(s) = \operatorname{tr}\left(P^{?[n]}_{m_n} \cdots P^{?[1]}_{m_1} \rho P^{?[1]}_{m_1} \cdots P^{?[n]}_{m_n}\right) = [\rho](s).$$

□

3.3.1 Consistent histories

In addition to quantum knowledge theory and probabilistic game semantics, the definition of quantum strategies given above was inspired by an alternative approach to quantum mechanics know as *quantum consistent histories* theory [Gri84, Omn88a, GMH93]. The main goal of this theory is to describe a quantum system using sequences of measured properties of the system. Each such history of the system must be assigned a weight in such a way that classical reasoning using probabilities is valid. The problem is to identify on which sets of histories this is possible. We summarize the usual solution to this problem in what follows. The central idea is that a measure of compatibility between histories is introduced and used to define the compatible ones.

Definition 3.11. *Let H be a Hilbert space. A **decoherence functional** is a map*

$$d\colon \mathrm{P}(H) \times \mathrm{P}(H) \to \mathrm{P}(H)$$

such that for all $P, P', Q \in \mathrm{P}(H)$

1. *$d(P,P) \in \mathbb{R}$ and $d(P,P) \geq 0$*
2. *$d(P,Q) = \overline{d(Q,P)}$*
3. *$d(I_H, I_H) = 1$ and $d(0,P) = 0$*
4. *If $P \perp P'$, then $d(P \oplus P', Q) = d(P,Q) + d(P',Q)$*

Properties of quantum systems are usually described using projectors. We call a projector

$$P_1 \otimes \cdots \otimes P_n \in \mathrm{P}(H \otimes \cdots \otimes H)$$

a **quantum history** in H. Given a density matrix $\rho \in \mathbf{D}(H)$, we can define a decoherence functional on $\mathrm{P}(H \otimes \cdots \otimes H)$ by

$$d_\rho(P_1 \otimes \cdots \otimes P_n, Q_1 \otimes \cdots \otimes Q_n) = \mathrm{tr}(P_n \ldots P_1 \rho Q_1 \ldots Q_n).$$

The sets of histories where classical reasoning on probabilities is valid can be defined as follows:

Definition 3.12. *Let H be a Hilbert space and d be a decoherence functional on $\mathrm{P}(H)$. A subset $S \subseteq \mathrm{P}(H)$ is **consistent** with respect to d if $p(P) = d(P,P)$ defines a probability distribution on S.*

The central result of the theory is a characterisation of consistent sets (explained for example in [Gri03])

Proposition 3.13. *$S \subseteq \mathrm{P}(H)$ is consistent for a decoherence functional d if and only if for all $P, Q \in S$ with $P \neq Q$ $\mathrm{Re}(d(P,Q)) = 0$.*

Note that consistent sets of quantum histories are defined using the stronger condition $d(P,Q) = 0$ for any two orthogonal quantum histories P and Q.

Histories are closely related to plays. It is straightforward to associate a decoherence functional ρ to each strategy $[\rho]$ in $[H]$. Given a quantum play $s = \mathcal{P}_{?[1]} m_1 \ldots \mathcal{P}_{?[n]} m_n$ in $\mathcal{L}_H^{\mathrm{even}}$, we have a quantum history

$$P_{m_1}^{?[1]} \otimes \cdots \otimes P_{m_n}^{?[n]}.$$

By construction we have that

$$\sum_{m_1,\ldots,m_n} P_{m_1}^{?[1]} \otimes \cdots \otimes P_{m_n}^{?[n]} = I$$

The strategy $[\rho]$ induces a probability distribution on the set of quantum histories:

$$p\left(P_{m_1}^{?[1]} \otimes \cdots \otimes P_{m_n}^{?[n]}\right) = \sigma\left(\mathcal{P}_{?[1]} m_1 \ldots \mathcal{P}_{?[n]} m_n\right).$$

3.4 General quantum arenas

Up to this point, we have studied isolated quantum systems arenas in order to understand how quantum states can be represented as strategies. Since our goal is to be able to study quantum types using quantum arenas, we need to extend the basic classical arena operations $\odot$ and $\multimap$ to quantum arenas.

3.4.1 Quantum arenas

The definition of an IQSA A is simply to add a Hilbert space H_A to the specification of an arena which acts as the space where the projective measurements are made. In a general arena involving many quantum systems, the projective measurements are performed in different Hilbert spaces. Hence we need to extend definition 3.1 as follows:

Definition 3.14. *A* quantum arena *A is an arena $(M_A, \vdash_A, \lambda_A)$ together with for each $a \in M^{\mathrm{Q}}_A$ a Hilbert space H_a such that* $|\{b \mid a \vdash_A b\}| = \dim(H_a)$

We can see IQSA as special cases of quantum arenas where the same Hilbert space is associated to every quantum question. Quantum plays and probabilistic strategies in quantum arenas are defined in the same manner as for IQSA, except that the projective measurements $\mathcal{P}_a$ associated to a question a are taken in H_a.

3.4.2 Products of quantum arenas

Definition 3.15. *Given two quantum arenas A and B, $A \odot B$ is the quantum arena with $|A| \odot |B|$ as underlying arena and where the Hilbert space H_a for $a \in M^{\mathrm{Q}}_{A \odot B}$ is taken to be the Hilbert space H_a in the component A or B where a comes from.*

The $\otimes$ operation extends to morphisms in the same way as the probabilistic case explained in section 2.3.4.

Let's examine the anatomy of a quantum play in a quantum arena of the form $A \odot B$. The projective measurement associated to each question is taken on the component where the question is asked. A typical play in $[H_A] \odot [H_B]$ is of the form

$$\begin{array}{ccc}
[H_A] & \odot & [H_B] \\
 & & \mathcal{P}_{?_B[1]} \\
 & & m^B_1 \\
\mathcal{P}_{?_A[1]} & & \\
m^A_1 & & \\
 & & \mathcal{P}_{?_B[2]} \\
 & & m^B_2
\end{array}$$

This play involve the two systems A and B independently. We can look at these measurements in the complex Hilbert space $H_A \otimes H_B$ describing the combination of the two systems. It is clear that the measurements occurring in quantum plays of $[H_A] \odot [H_B]$ correspond to measurements of one of the two forms

$$\{P_{m^A} \otimes I_B \mid P_{m^A} \in \mathrm{P}(H_A)\} \quad \text{or} \quad \{I_A \otimes Q_{m^B} \mid Q_{m^B} \in \mathrm{P}(H_B)\}.$$

By comparison, in the quantum arena $[H_A \otimes H_B]$, Opponent can use a projective measurement with any projectors on $H_A \otimes H_B$. This allows her to choose the projections onto the Bell states, which is not possible in the case of $H_A \odot H_B$.

Given a quantum state $\rho_A \odot \rho_B$ over $H_A \otimes H_B$, we can define a strategy $[\rho_A \otimes \rho_B]$ in $[H_A] \odot [H_B]$ as we did in the case of a IQSA $[H]$. Given a play $s = \mathcal{P}_{?[1]}m_1 \dots \mathcal{P}_{?[n]}m_n$, where the questions $?[k]$ and their associated answers m_k can be from either the A or the B component, we set

$$[\rho_A \otimes \rho_B](s) = \operatorname{tr}\left(P^{?[n]}_{m_n} \dots P^{?[1]}_{m_1} (\rho_A \otimes \rho_B)\, P^{?[1]}_{m_1} \dots P^{?[n]}_{m_n}\right).$$

Note that in this expression it is possible to use a state ρ over $H_A \otimes H_B$ which is not a tensor product $\rho_A \otimes \rho_B$.

The difference between $[H_A] \odot [H_B]$ and $[H_A \otimes H_B]$ is important, as the limitations in the allowed measurements in the first case make it an unsuitable candidate to represent joint quantum systems. This is because there are families of states in $H_A \otimes H_B$ which cannot be distinguished by measurements of the form allowed in $[H_A] \odot [H_B]$ together with classical processing of the results. Moreover, the states in these families can even be chosen to be separated. Such an example is the following set of orthogonal states in $\mathbb{C}^3 \otimes \mathbb{C}^3$:

$$\begin{array}{lll}
|1\rangle \otimes |1\rangle & |0\rangle \otimes (|0\rangle + |1\rangle) & |0\rangle \otimes (|0\rangle - |1\rangle) \\
|2\rangle \otimes (|1\rangle + |2\rangle) & |2\rangle \otimes (|1\rangle - |2\rangle) & (|1\rangle + |2\rangle) \otimes |0\rangle \\
(|1\rangle - |2\rangle) \otimes |0\rangle & (|0\rangle + |1\rangle) \otimes |2\rangle & (|0\rangle - |1\rangle) \otimes |2\rangle
\end{array}$$

where $|0\rangle$, $|1\rangle$ and $|2\rangle$ is an orthonormal basis of $\mathbb{C}^3$. This example is discussed in the paper [BDF$^+$99], where it is proved that even with classical communication, two parties cannot distinguish these states with certainty using separate measurements on each component of the system. The paper gives other examples of such phenomena.

We introduce another product operation on quantum arenas where any question $\mathcal{P}_a$ over a tensor product space can be asked.

Definition 3.16. *Let A and B be two quantum arenas. The quantum arena $A \otimes B$ is defined by*

1. $M_{A \otimes B} = \{(a, b) \in M_A \times M_B \mid \lambda_A(a) = \lambda_B(b)\}$
2. $\lambda_{A \otimes B}((a, b)) = \lambda_A(a) = \lambda_B(b)$
3. $H_{(a,b)} = H_a \otimes H_b$ *for* $(a, b) \in M^{\mathrm{Q}}_{A \otimes B}$
4. $(a_1, b_1) \vdash_{A \otimes B} (a_2, b_2)$ *if* $a_1 \vdash b_1$ *and* $a_2 \vdash b_2$

It is easy to see that in particular $[H_A] \otimes [H_B] = [H_A \otimes H_B]$. To simplify the notation, we will denote the measurement results (m_A, m_B) in quantum arenas of the form $A \otimes B$ by $m^A m^B$. When we do not need to refer to the measurement results in each component, a generic tuple of measurement results is denoted by $\overline{m}$.

3.4.3 The linear arrow quantum arena

We now turn to the other basic arena operation which is used to represent quantum operations as strategies.

Definition 3.17. *Given two quantum arenas A, B, $A \multimap B$ is the quantum arena with $|A| \multimap |B|$ as underlying arena and where, for $m \in M^{\mathrm{Q}}_{A \multimap B}$, H_m is defined to be H_m in the component A or B where a comes from.*

A typical single-threaded play in $[H_A] \multimap [H_A]$ is as follows:

$$\begin{tikzcd}
{[H_A]} \arrow[r, -o] & {[H_B]} \\
& \mathcal{P}_{?_B} \\
\mathcal{P}_{?_A[1]} \arrow[ur] & \\
m_1^A \arrow[u] & \\
\vdots & \\
\mathcal{P}_{?_A[n]} \arrow[uuuur] & \\
m_n^A \arrow[u] & \\
& m^B \arrow[uuuuuu]
\end{tikzcd}$$

In this play, Opponent wants to know the result of measuring the output with the projective measurement $\mathcal{P}_{?_B}$. Player answers this by asking Opponent to measure the input state with the projective measurements $\mathcal{P}_{?[1]}, \dots, \mathcal{P}_{?[n]}$. Player can then use the measurement results to decide how to answer to Opponent's initial question.

Important classes of quantum operations can be defined in the quantum arena

$$[H_A] \multimap [H_B].$$

Example 3.18. A unitary transformation $U \colon H \to H$ can be represented as a total deterministic strategy $[U] \colon H \multimap H$ where Player plays following this pattern:

$$\begin{tikzcd}
{[H_l]} \arrow[r, -o, "{[U]}"] & {[H_r]} \\
& \mathcal{P}_{?_r} \\
\mathcal{P}_{?_l} & \\
m & \\
& m
\end{tikzcd}$$

where

$$\mathcal{P}_{?_r} = U \mathcal{P}_{?_l} U^\dagger = \left\{ U P_m U^\dagger \mid m = 0 \dots \dim(H_r) - 1 \right\}.$$

We can check that the proposed strategies for unitary transformations behave in the proper way: if we compose $[U] \colon H \multimap H$ with a state strategy $[\rho] \colon I \multimap H$, what we get is the state strategy $[U\rho U^\dagger]$:

$$\begin{aligned}\left[U\rho U^\dagger\right](\mathcal{P}_?m) &= \mathrm{tr}\left(P_m\left(U\rho U^\dagger\right)P_m\right)\\ &= \mathrm{tr}\left((P_mU)\rho(P_mU)^\dagger\right)\\ &= [U][\rho](\mathcal{P}_?m)\end{aligned}$$

Example 3.19. Let $[H_A]$ and $[H_B]$ be two Hilbert spaces arenas. The partial trace strategy $[\mathrm{tr}^B]$ is defined as the deterministic, total and thread independent strategy which assigns weight 1 to single threaded plays of the form

$$\begin{array}{ccc} [H_A]\otimes[H_B] & \xrightarrow{[\mathrm{tr}^B]} & [H_A] \\ & & \mathcal{P}_? \\ \mathcal{P}'_? & & \\ m^Am^B & & \\ & & m^A \end{array}$$

where $\mathcal{P}'_? = \{(P^?_{m^A}\otimes I)\}_?$. Note that, since the measurement results in $[H_A]\otimes[H_B]$ must be of the form m^Am^B, we can fix arbitrarily an index m^B_0 in the indexing of the elements of $\mathcal{P}'$. When this strategy is composed with a state strategy $[\rho]$, the resulting strategy is the state strategy $[\mathrm{tr}^B\rho]$ associated with the reduced density matrix. Let $|j\rangle$ be an orthonormal basis of H_B. We have that

$$\begin{aligned}\mathrm{tr}^B;[\rho]\left(\mathcal{P}_?\mathcal{P}'_?m^Am^B_0m^A\right) &= \sum_m \mathrm{tr}\left(P'_{mm^B_0}\rho\right)\delta_{m^Am}\\ &= \mathrm{tr}\left(P_{m^A}\otimes I\rho\right)\\ &= \mathrm{tr}\left(P_{m^A}\otimes\left(\sum_j |j\rangle\langle j|\right)\rho\right)\\ &= \mathrm{tr}\left(P_{m^A}\sum_j\langle j|\rho|j\rangle\right)\\ &= \mathrm{tr}\left(P_{m^A}\,\mathrm{tr}^B(\rho)\right)\\ &= \left[\mathrm{tr}^B(\rho)\right]\left(\mathcal{P}_?m^A\right)\end{aligned}$$

Note that the projection strategies on $[H_A]\odot[H_B]$ also compute the trace of ρ when composed with a strategy $[\rho]\colon I \multimap [H_A]\odot[H_B]$. For the projection π_A on the first component:

$$\begin{array}{ccc} [H_A]\ \odot\ [H_B] & \xrightarrow{\pi_A} & [H_A] \\ & & \mathcal{P}_? \\ \mathcal{P}_? & & \\ m^A & & \\ & & m^A \end{array}$$

we have that $\pi_A[\rho] = \left[\mathrm{tr}^B(\rho)\right]$.

Example 3.20. The effect of performing a projective measurement Q on a space H can also be represented with a strategy $[Q] : [H] \multimap [H]$. Player plays according to the following pattern:

$$\begin{array}{lcl}
[H] & \overset{[Q]}{\multimap} & [H] \\
 & & \mathcal{P}_? \\
Q_? & & \\
m_1 & & \\
\mathcal{P}_? & & \\
m_2 & & \\
 & & m_2
\end{array}$$

The state after the measurement Q is performed is $Q(\rho)$ – we abuse the notation and use Q to denote both the set of projectors $\{Q_m\}$ and the associated superoperator defined by

$$\rho \mapsto \sum_m Q_m \rho Q_m.$$

Composing with a strategy $[\rho]$ in $I \multimap [H]$, we get that

$$\begin{aligned}
[\mathcal{P}][\rho]\,(\mathcal{P}_? m_2) &= \sum_{m_1} \mathrm{tr}\left(P^?_{m_2} Q^?_{m_1} \rho Q^?_{m_1} P^?_{m_2}\right) \\
&= \mathrm{tr}\left(P^?_{m_2} \sum_{m_1} \left(Q^?_{m_1} \rho Q^?_{m_1}\right) P^?_{m_2}\right) \\
&= \mathrm{tr}\left(P^?_{m_2} Q(\rho)\right).
\end{aligned}$$

The above examples make it possible to represent three of the four components of the decomposition of superoperators: unitary transformations, projective measurements, partial traces. Unfortunately, it is not possible to do the same for the missing preparation: there is no such strategy to describe preparation of a new state. Suppose we want to define such a strategy in the arena $[H_A] \multimap [H_A \otimes H_B]$ which corresponds to the operation that takes a state ρ to a state $\rho \otimes |\varphi\rangle\langle\varphi|$. Using projective measurements, we need to associate to a question $\mathcal{P}_{?_{AB}}$ in $[H_A \otimes H_B]$ a question $\mathcal{P}_{?_A}$ in $[H_A]$ such that

$$\mathrm{tr}\left(P^{?_{AB}}_m \left(\rho \otimes |\varphi\rangle\langle\varphi|\right)\right) = \mathrm{tr}\left(P^{?_A}_m \rho\right).$$

Following a scheme similar to the case of unitary strategies, we would take

$$P^{?_A}_m = \langle\varphi| P^{?_{AB}}_m |\varphi\rangle.$$

The problem is that $\langle\varphi| P^{?_{AB}}_m |\varphi\rangle$ is not a projector in general. This is a difficulty that makes it impossible to get a general correspondence between strategies in $[H_A] \multimap [H_B]$ and superoperators from H_A to H_B following the scheme used in the examples above.

As it is the case with general probabilistic strategies in $[H_A]$, a general probabilistic strategy in $[H_A] \multimap [H_B]$ needs not to respect the laws of quantum mechanics. One such example would be a strategy σ which makes Player ignore the $[H_A]$ component and play in $[H_B]$ using a probabilistic strategy which does not correspond to a quantum state. When composed with a state strategy $[\rho]$ in $[H_A]$, this gives a strategy in $[H_B]$ which is not a quantum state. Thus the strategy σ does not correspond to a superoperator.

It is possible to see both the unitary and partial traces constructions as special cases of a construction involving trace-preserving superoperators. Suppose we have such a superoperator $\mathcal{E}\colon \mathbf{SD}(H) \to \mathbf{SD}(H)$, with $\dim(H_A) \geq \dim(H_B)$ such that $\mathcal{E}^*$ maps projectors to projectors. We define a strategy as above using the adjoint $\mathcal{E}^*$ to $\mathcal{E}$:

$$\begin{tikzcd}
{[H]} \arrow[r, multimap, "{[\mathcal{E}]}"] & {[H]} \\
 & \mathcal{P}_? \\
\mathcal{E}^*\,(\mathcal{P}_?) & \\
m & \\
 & m
\end{tikzcd}$$

Since $\mathcal{E}$ is trace-preserving, $\mathcal{E}^*$ is unital and

$$\sum_m \mathcal{E}^*(P_m) = \mathcal{E}^*(I) = I.$$

This shows that $\mathcal{E}(\mathcal{P}_?)$ is a complete projective measurement. As in the above examples, a direct computation shows that $[\mathcal{E}][\rho] = [\mathcal{E}(\rho)]$.

Product of strategies

Suppose we have two probabilistic strategies $\sigma\colon [H_A] \multimap [H_B]$ and $\tau\colon [H_C] \multimap [H_D]$. Can we define a tensor product strategy $\sigma \otimes \tau$ in the arena $[H_A] \otimes [H_C] \multimap [H_B] \otimes [H_D]$? It is impossible to do so in general, since this would require, in a typical case, that we construct a projective measurement $\mathcal{P}'_?$ in $H_A \otimes H_C$ from a projective measurement $\mathcal{P}_?$ in $H_B \otimes H_D$:

$$\begin{tikzcd}
{[H_A] \otimes [H_C]} \arrow[r, multimap] & {[H_B] \otimes [H_D]} \\
 & \mathcal{P}_? \\
\mathcal{P}'_? &
\end{tikzcd}$$

The strategy σ may provide a way to connect a projective measurement on H_B to a projective measurement on H_A and similarly τ may connect a projective measurement on H_D to a projective measurement on H_C, but there is no way to separate $\mathcal{P}_?$ into two projective measurements to use σ and τ to define $\mathcal{P}'_?$. Yet, it is possible in many important examples to define a strategy $\sigma \otimes \tau$ that acts as expected. Consider for example the situation where σ and τ are two unitary transformation strategies, say $\sigma = [U_1]$ and $\tau = [U_2]$. In this case, we can define $\mathcal{P}'_?$ as $(U_1 \otimes U_2)^\dagger \mathcal{P}_? (U_1 \otimes U_2)$. This give that $[U_1] \otimes [U_2] = [U_1 \otimes U_2]$. We can define similarly a tensor strategy of two partial traces strategy with the property $[\mathrm{tr}_1] \otimes [\mathrm{tr}_2] = [\mathrm{tr}_1 \otimes \mathrm{tr}_2]$.

3.5 The category of quantum arenas and quantum strategies

In the previous sections we have introduced all the necessary concepts needed to define a category **QStrat** of quantum arenas and strategies.

Given two probabilistic strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$, we define their composition in the same way as in section 2.3:

$$[\sigma;\tau](s) = \sum_{u \in \mathrm{wit}(s)} \sigma(u|_{AB})\, \tau(u_{BC}).$$

We take quantum arenas $[H]$ as objects of **QStrat**. Since not all strategies

$$\sigma\colon [H_A] \multimap [H_B]$$

preserve quantum states, we need to restrict the definition of morphisms $[H_A] \to [H_B]$ to the strategies $\sigma\colon [H_A] \multimap [H_B]$ such that for every $[\rho]\colon I \to [H_A]$, the composition $\sigma[\rho]$ is also a strategy of the form $[\rho']$ for some state ρ' in H_B. The composition of two such strategies clearly satisfies the same condition. The identity strategies trivially have this property.

3.5.1 Quantum strategies as probabilistic strategies

In the application of quantum strategies as interpretation of terms of quantum languages, quantum arenas and strategies cohabit with classical arenas and strategies that represent classical terms and data. We thus need to be able to mix classical and quantum strategies.

Quantum strategies are defined as special kind of probabilistic strategies, and thus the category **QStrat** can be embedded in the category **PStrat**. Given a quantum arena A, we define the (probabilistic) arena $\overline{A}$ to be the arena obtained by replacing questions with all possible $(a, \mathcal{P})$ such that $\mathcal{P}$ is a projective measurement on H_a and keeping the same labelling and enabling structure, so that every play in $\overline{A}$ is a quantum play in A. This extends trivially to strategies in A, which can be identified with probabilistic strategies in $\overline{A}$. An arena $[H]$ is sent to a probabilistic arena $\overline{[H]}$ and a probabilistic strategy $\sigma\colon [H_A] \multimap [H_B]$ in **QStrat** is sent to the strategy $\overline{\sigma}$.

In the subsequent chapters, we will work in the category **Pstrat**, identifying $\overline{A}$ and A. Note that while the two arena operations $\multimap$ and $\odot$ are preserved in this embedding, the tensor operation between quantum arenas cannot be extended to all probabilistic arenas. This is an important fact that guided the design of the type system of the two quantum λ-calculi presented in chapter 5 and 6.

3.5.2 Tensor product of strategies with classical interactions

In 3.4.3 we showed how to defined the tensor product of strategies representing various important quantum operations. In general the application of a quantum operation is conditional on some previous classical data, and embedding **Qstrat** in **Pstrat** allows us to encode these dependencies using probabilistic strategies.

There are two different cases to consider. In the first case, we have quantum states in H that depend on some classical interaction in an arena U:

$$\begin{array}{ccc}
U & \overset{\sigma}{\multimap} & [H] \\
 & & \mathcal{P}_? \\
a_1 & & \\
\vdots & & \\
a_n & & \\
 & & m
\end{array}$$

Since the answer m to $\mathcal{P}_?$ must describe a quantum state, we assume that there is a density matrix ρ_s, $s = a_1 \dots a_n$, such that

$$\sigma\,(m \mid \mathcal{P}_? s) = \mathrm{tr}\left(P^?_m \rho_s\right).$$

A simple example of this situation is a conditional preparation strategy prep which behaves like state $|b\rangle\langle b|$ according to some boolean value $b \in \{0, 1\}$. This strategy is described by the following typical play:

$$\begin{array}{ccc}
\mathbf{bool} & \overset{\sigma}{\multimap} & [H] \\
 & & \mathcal{P}_? \\
? & & \\
b & & \\
 & & m
\end{array}$$

where

$$\mathrm{prep}\,(m \mid \mathcal{P}_? ?b) = \mathrm{tr}\left(P^?_m \,|b\rangle\langle b|\right).$$

We can define a tensor $\sigma \otimes \tau$ of two such strategies σ and τ as follows:

$$\begin{array}{ccccc}
U_A & \odot & U_B & \overset{\sigma\otimes\tau}{\multimap} & [H_A] \otimes [H_B] \\
 & & & & \mathcal{P}_? \\
a_1 & & & & \\
\vdots & & & & \\
a_n & & & & \\
 & & b_1 & & \\
 & & \vdots & & \\
 & & b_m & & \\
 & & & & m^A m^B
\end{array}$$

where the probability that Player answers $m^A m^B$ to $\mathcal{P}_?$ after the interactions $s = a_1 \dots a_n$ and $t = b_1 \dots b_m$ is $\mathrm{tr}\,(P_{m^A m^B} \rho_s \otimes \rho_t)$. Note that while we take the tensor product of the two output quantum arenas, we must take the classical game product of the classical input arenas.

The second case to consider is when σ and τ are both strategies that correspond to conditional quantum operations. The general pattern is similar to the purely quantum case presented in 3.4.3, but we add to the input arena of σ and τ two classical arenas U_A and U_B where the classical part of the interaction occurs:

$$\sigma\colon U_A \odot [H_A] \multimap [H_B], \quad \tau\colon U_C \odot [H_C] \multimap [H_D].$$

For example, consider a typical thread in the first case: the interaction looks as follows:

$$\begin{array}{ccccc}
U_A & \odot & [H_A] & \overset{\sigma}{\multimap} & [H_B] \\
 & & & & \mathcal{P}_? \\
a_1 & & & & \\
\vdots & & & & \\
a_n & & & & \\
 & & \mathcal{E}_s^*(\mathcal{P}_?) & & \\
 & & m & & \\
 & & & & m
\end{array}$$

where $s = a_1 \ldots a_n$ and $\mathcal{E}_s$ is a trace-preserving superoperator such that $\mathcal{E}_s^*$ preserve projectors. Note this is a deterministic strategy.

Assuming that σ and τ are two strategies as above, we can define $\sigma \otimes \tau$ by the following typical play:

$$\begin{array}{ccccccc}
U_A & \odot & U_B & \odot & [H_A] \otimes [H_C] & \overset{\sigma\otimes\tau}{\multimap} & [H_B] \otimes [H_D] \\
 & & & & & & \mathcal{P}_? \\
a_1 & & & & & & \\
\vdots & & & & & & \\
a_{n_A} & & & & & & \\
 & & b_1 & & & & \\
 & & \vdots & & & & \\
 & & b_{n_B} & & & & \\
 & & & & \left(\mathcal{E}_s^A \otimes \mathcal{E}_t^B\right)^* (\mathcal{P}_?) & & \\
 & & & & mm' & & \\
 & & & & & & mm'
\end{array}$$

where the probability that Player answers mm' to $\mathcal{P}_?$ after the interactions $s = a_1 \ldots a_{n_A}$ and $t = b_1 \ldots b_{n_B}$ is $\mathrm{tr}\,(P_{mm'}\,\rho_s \otimes \rho_t)$.

Diagonal strategies and quantum strategies

We mentioned in section 2.3 the important role played by the diagonal strategy

$$\Delta\colon A \to A \odot A$$

in game semantics: thread independent strategies $\sigma\colon A \to B$ are duplicable:

$$\begin{tikzcd}
A \arrow[r, "\Delta"] \arrow[d, "\sigma"'] & A \odot A \arrow[d, "\sigma\odot\sigma"] \\
B \arrow[r, "\Delta"'] & B \odot B
\end{tikzcd}$$

It is well-know in game semantics that thread dependent strategies describe processes with *side-effects*. These are usually associated with *stores* where content is affected by various operations like incrementation or assignment. In a classical language, the effect of these operations is not immediately visible: it is only through access to the store content that the previously made operations can affect the future of the computation.

Quantum strategies of the form $[\rho]\colon I \multimap [H]$ are thread dependant because they encode the dynamics of the evolution of a quantum state: measurements are operations performed on the state and their effect is not visible to the environment until the next measurement is performed. Since in a quantum play each measurement operation correspond to a different thread, quantum strategies must be thread dependent.

The strategy $[\rho]$ is defined assuming that Player provides the measurement result answers to an Opponent question by observing the state resulting from the last performed measurement (or ρ in the case of the initial question). The following diagram is not commutative:

$$\begin{tikzcd}
I \arrow[r, "\Delta"] \arrow[d, "{[\rho]}"'] & I \odot I \arrow[d, "{[\rho]\odot[\rho]}"] \\
{[H]} \arrow[r, "\Delta"'] & {[H] \odot [H]}
\end{tikzcd}$$

This means that a quantum strategy cannot be *cloned* using the Δ strategy. On the one hand, the role of Δ in $[\rho]; \Delta$ is to allow Opponent to have access to $[\rho]$ from two different instances of the arena $[H]$. A question $\mathcal{P}_?$ in either the left or the right $[H]$ is answered using $[\rho]$, and the future answers are affected in the same way in both cases. On the other hand, $\Delta \circ [\rho] \odot [\rho]$ behaves like two independent instances of $[\rho]$: a question $\mathcal{P}_?$ in the final left $[H]$ arena is answered using the left $[\rho]$ strategy and does not affect the right $[\rho]$ strategy.

We can define another strategy $\{\rho\}$ which is not dynamical and is thus duplicable. This is done by assuming that Player has an infinite supply of quantum states ρ, and that each measurement asked is performed on a fresh state. The strategy $\{\rho\}$ corresponding to this scenario is defined by

$$\{\rho\}(m \mid s\mathcal{P}_?) = \operatorname{tr}\left(P^?_m \rho\right)$$

This new strategy is thread independent since

$$\frac{\{\rho\}\,(s\mathcal{P}_?m)}{\{\rho\}(s)} = \operatorname{tr}\left(P^?_m\rho\right) = \frac{\{\rho\}\,(t\mathcal{P}_?\,m)}{\{\rho\}(t)},$$

and thus this time $\{\rho\}$ is duplicable, i.e. the following commutes:

$$\begin{tikzcd}
I \arrow[r, "\Delta"] \arrow[d, "\{\rho\}"'] & I \odot I \arrow[d, "\{\rho\}\odot\{\rho\}"] \\
{[H]} \arrow[r, "\Delta"'] & {[H] \odot [H]}
\end{tikzcd}$$

This relation between $\{\rho\}$ and Δ is saying that if we have an infinite supply of copies of a quantum system all in a certain quantum state, we can use them as two infinite supplies of quantum systems in that state.

Note that there is no equivalent to the diagonal strategy for the tensor operation $\otimes$. This is because such a strategy $D\colon [H] \multimap [H] \otimes [H]$ would be cloning unknown quantum states, which is impossible. We can look at this from a new angle with the concept of quantum strategy. To define D, one would need to take a projective measurement $\mathcal{P}_?$ in $H \otimes H$ and transform it into a projective measurement on H, but there is no natural way to do so.

3.6 Quantum strategies using other quantum measurements

In the previous section, quantum plays are defined using projective quantum measurements. It is possible to work with the other types of quantum measurement described in section 2.1.2. An important motivation in doing so is the problem exposed in 3.4.3 of defining a preparation strategy using projective measurements, which is due to the fact that the family of projectors is not closed under certain operations.

In a quantum arena A, the number of possible answers to a question $\mathcal{P}_a$ must be the dimension of the associated Hilbert space H_a. This assumption was made so that it is never the case that there are more possible measurement results to a question $\mathcal{P}_a$ than the dimension of H_a. Since we work below with other types of quantum measurements, and since these measurements can handle more measurement results than $\dim(H_a)$, we drop this limitation when working with other types of measurements than projective measurements.

3.6.1 Generalised measurement based quantum strategies

Suppose that in a quantum play we allow using generalised measurements of the form $\mathcal{M}_a = \{M_b \mid a \vdash b\}$ instead of projective measurements. It is still possible to define strategies for quantum states, unitary transformations and partial traces in a similar way as when working with projective measurements.

Given any state ρ in H we define a strategy $[\rho]$ in $[H]$ in a similar manner as with projective measurements. The strategy $[\rho]$ makes Player answer an initial question $\mathcal{M}_?$ with the measurement result m with probability $\operatorname{tr}\left(M_m^? \rho \left(M_m^?\right)^\dagger\right)$.

In the case of a unitary transformation U, we can define a strategy $[U]$ where Player plays as follows:

$$\begin{array}{ccc} [H] & \multimap & [H] \\ & & \mathcal{M}_{?_B} \\ \mathcal{M}_{?_A} & & \\ m & & \\ & & m \end{array}$$

where $\mathcal{M}_{?_A} = \left\{M_m^{?_B} U\right\}$. This defines a new generalized measurement since:

$$\sum_i (M_m U)^\dagger (M_m U) = U^\dagger \left(\sum_m M_m^\dagger M_m\right) U = U^\dagger I U = I$$

As with projective measurements, when we compose this with a strategy $[\rho]\colon I \multimap [H]$, we get

$$\begin{aligned} [U][\rho](\mathcal{M}_{?_B}\mathcal{M}_{?_A} mn) &= \sum_n \operatorname{tr}\left((M_n U)\rho(M_n U)^\dagger\right)\delta_{mn} \\ &= \operatorname{tr}\left(M_m \left(U\rho U^\dagger\right) M_m^\dagger\right) \\ &= [U\rho U^\dagger]\left(\mathcal{M}_{?_B}\mathcal{M}_{?_A} mn\right). \end{aligned}$$

Strategies for partial traces using generalised measurements are defined similarly. It is also possible to define a strategy representing the effect of performing a generalised measurement, as in example 3.20.

The problem of the preparation strategy is still present. The usual scheme fails for a similar reason as in the case of projective measurements. Since we want a strategy in $[H_A] \multimap [H_A \otimes H_B]$, we need to associate to a question $\mathcal{M}_{?_{AB}}$ in $[H_A \otimes H_B]$ a question $\mathcal{M}_{?_A}$ in $[H_A]$ such that

$$\operatorname{tr}\left(M_m^{?_{AB}}(\rho \otimes |\varphi\rangle\langle\varphi|)\left(M_m^{?_{AB}}\right)^\dagger\right) = \operatorname{tr}\left(M_m^{?_A}\rho\left(M_m^{?_A}\right)^\dagger\right).$$

The natural candidate is $M_m^{?_A} = M_m^{?_{AB}}|\varphi\rangle$. It is easy to check that $\sum_m \left(M_m^{?_{AB}}\right)^\dagger M_m^{?_A} = I_A$, but $\mathcal{M}_{?_A}$ is not a generalised measurement because $M_m^{?_A}$ is a map from H_A to $H_A \otimes H_B$, and is thus not a map in $\mathbf{M}(H_A)$.

3.6.2 POVM based quantum strategies

Suppose that we use POVM measurements $\mathcal{A}_a = \{A_m \mid a \vdash M\}$ over H_a in quantum plays instead of projective measurements. It is again possible to define strategies for quantum states, unitary transformation and partial traces with a similar construction as in the case of projective measurements and generalised measurements. It is even possible to define a strategy $[\mathcal{E}]$ in $[H_A] \multimap [H_B]$ for any trace-preserving superoperator $\mathcal{E}$. The strategy $[\mathcal{E}]$ is defined as follows:

$$\begin{array}{ccc} [H_A] & \multimap & [H_B] \\ & & \mathcal{A}_{?_B} \\ \mathcal{A}_{?_A} & & \\ m & & \\ & & m \end{array}$$

where $A_m^{?_A} = \mathcal{E}^*\left(A_m^{?_B}\right)$. This is always a positive operator. The adjoint $\mathcal{E}^*$ being unital, we have that

$$\sum_m A_m^{?_A} = \sum_m \mathcal{E}^*\left(A_k^{?_B}\right) = \mathcal{E}^*\left(\sum_m A_m^{?_B}\right) = \mathcal{E}^*(I) = I,$$

And thus $\mathcal{E}^*\left(\mathcal{A}_{?_B}\right)$ is a POVM. If we compose $[\mathcal{E}]$ with $[\rho]$, we get

$$\begin{aligned}
[\mathcal{E}][\rho](\mathcal{A}_{?_B}\mathcal{A}_{?_A}nm) &= \sum_j \operatorname{tr}\left(\sum_k E_k^\dagger A_j E_k \rho\right)\delta_{ij} \\
&= \operatorname{tr}\left(A_i \sum_k E_k \rho E_k^\dagger\right) \\
&= \operatorname{tr}\left(A_i \mathcal{E}(\rho)\right) \\
&= [\mathcal{E}(\rho)]\left(\mathcal{A}_{?_B}\mathcal{A}_{?_A}nm\right)
\end{aligned}$$

so $[\mathcal{E}]$ acts on quantum state strategies as $\mathcal{E}$ acts on quantum states.

There is a important limitation when using POVM based quantum strategies. The quantum state left after a POVM is not specified, so it is not possible to have quantum strategies with multiple threads as with projective measurements. In particular, there is no natural way to use the scheme of example 3.20 with POVM based quantum strategies since it involves two successive measurements of the input state. It is possible to work with the convention that a POVM $\{A_m\}$ updates a state ρ to $\sqrt{A_m}\rho\sqrt{A_m}$ when the measurement result is m. We use this idea in the construction of a denotational semantics in the next chapter.

3.6.3 Intervention operators

The last generalisation of the definition of quantum play we consider is the case of intervention operators. This is the most important example; we use this generalisation in chapter 5 and 6.

In this case, the projective measurements associated to question moves are replaced with quantum interventions. We associate to a question q a family of superoperators

$$\mathcal{E}_q = \left\{\mathcal{E}_m^q \colon \mathbf{SD}(H_q) \to \mathbf{SD}(H_m)\right\}$$

indexed by the possible measurement results m. When working with quantum interventions, we need to drop the limitation imposed in definition 3.1 on the number of answers to a given question. This is because quantum interventions can take a state ρ in a certain space and map it to a state in a different space. Note that in a general play, the different occurrences of a question q will not all have the same associated Hilbert space H_q.

The strategy describing a state $[\rho]$ is defined similarly as in the other cases: Player answers m to $\mathcal{E}_?$ with probability $\operatorname{tr}\left(\mathcal{E}^?_m(\rho)\right)$. Note that the state after the intervention is $\mathcal{E}^?_m(\rho)$, which is in $\mathbf{SD}(H_m)$. In general, this will be a different space than the space before the answer is given, namely $\mathbf{SD}(H_q)$. If Opponent asks another question $\mathcal{E}_{q[2]}$ after receiving her answer to $\mathcal{E}_q$, all possible Player answers will have probability zero when the domain of $\mathcal{E}_{q[e]}$ is different than $\mathbf{SD}(H_m)$. When the domain and $\mathbf{SD}(H_m)$ match, the question $\mathcal{E}_{q[2]}$ is answered using the normalised state $\mathcal{E}^?_m(\rho)/\operatorname{tr}\left(\mathcal{E}^?_m(\rho)\right)$. In general, a typical play in **qstore** is a sequence of the form

$$\mathcal{E}_{?[1]}m_1 \ldots \mathcal{E}_{?[n]}m_n$$

consisting of alternating quantum interventions and measurement results. The strategy $[\rho]$ in **qstore** is defined by

$$[\rho]\left(m_n \mid \mathcal{E}_{?[1]}m_1 \ldots \mathcal{E}_{?[n]}\right) = \operatorname{tr}\left(\mathcal{E}^{?[n]}_{m_n} \ldots \mathcal{E}^{?[1]}_{m_1}(\rho)\right).$$

Note that we consider $[\rho]$ to be a partial strategy: it is possible that Opponent asks $\mathcal{E}_?$ using an intervention operator with an input space which is not the same as the last output space or with the space from which the starting state is taken. We define $[\rho]$ as assigning probability zero to all plays where this is the case.

The scheme used to represent quantum operations with the other quantum measurements formalisms can also be used with intervention operators. Suppose that $\mathcal{F}$ is a trace-preserving superoperator. We have that

$$\begin{tikzcd}
{[H_A]} \arrow[r, "{[\mathcal{F}]}", "\circ"{pos=1, description, draw=none}] & {[H_B]} \\
 & \mathcal{E}_{?_B} \\
\mathcal{E}_{?_A} & \\
m & \\
 & m
\end{tikzcd}$$

where $\mathcal{E}_{?_A}$ is taken to be $\mathcal{E}_{?_B}\mathcal{F} = \left\{\mathcal{E}^{?_B}_m \circ \mathcal{F}\right\}$. Since $\mathcal{F}$ preserves traces, $\mathcal{E}_{?_B}\mathcal{F}$ is again a quantum intervention:

$$\sum_m \operatorname{tr}(\mathcal{E}^{?_B}_m \mathcal{F}(\rho)) = \sum_m \mathcal{E}^{?_B}_m \rho' = 1,$$

where $\rho' = \mathcal{F}(\rho)$. When we compose the strategy $[\mathcal{F}]$ with a state strategy $[\rho]$, we get that

$$\begin{aligned}[\mathcal{F}][\rho]\left(n \mid \mathcal{E}_{?_B}\mathcal{E}_{?_A} m\right) &= \sum_m \operatorname{tr}\left(\mathcal{E}_m^{?_B}\mathcal{F}(\rho)\right)\delta_{mn} \\ &= \operatorname{tr}\left(\mathcal{E}_n^{?_B}\left(\mathcal{F}(\rho)\right)\right) \\ &=[\mathcal{F}(\rho)].\end{aligned}$$

Chapter 4

Game semantics for the measurement calculus

As a first application of the quantum strategy concept developed in the last chapter, we study in this chapter the low-level *measurement calculus*, presented in section 2.2.1. The measurement calculus lacks an explicit type system: this was not required when the measurement calculus language was first introduced, since the main purpose then was to simplify the presentation of the one-way model [RB01]. Because the construction of a game semantics is based upon games corresponding to types, we must begin by the introduction of a typed variant of the measurement calculus.

4.1 MCdata

The formalization of the measurement calculus that follow aims to construct a type system where commands are typed in a way that automatically enforces the three conditions defining patterns in section 2.2.1.

The approach chosen here to add types to the measurement calculus is not the unique possibility. We choose to consider quantum states as constants, as we do for Boolean and angle values. This forces commands to be operations taking quantum data to quantum data. Signals are considered as (classical) *stores* where measurements results are written when a quantum measurement is performed and read by dereferencing. Another possible approach would be to represent quantum data as the state of a quantum store. In that case, commands are considered as operations modifying the internal state of the store. While we do not develop this idea in the case of the measurement calculus, it is developed in chapter 5 in the case of a quantum λ-calculus.

4.1.1 Syntax

The language MCdata we define below uses *labelled types* and *labelled terms*. The labels identify qbits, and allow signals to be tied to specific measured qbits.

The terms of the language MCdata are constructed as follows:

$$\begin{array}{lrcl}
\text{Booleans} & B, B_1, B_2 & ::= & 0 \mid 1 \mid !s \mid B_1 \oplus B_2 \\
\text{Angles} & W, W_1, W_2 & ::= & \alpha \mid W_1 + W_2 \mid \mathsf{rot}\, W\, B_1\, B_2 \\
\text{qbits} & Q & ::= & x \mid |\phi\rangle^I \mid \mathsf{meas}^i_I\, s\, W\, Q \mid \mathsf{E}_{ij}\, Q \mid \mathsf{X}_i\, B\, Q \mid \ \mathsf{Z}_i\, B\, Q
\end{array}$$

where s, x are variable picked from an infinite set of variables Var, $\alpha \in [0, 2\pi)$, i, j are qbit labels, I is a finite set of labels and $|\phi\rangle^I$ is a state in the complex Hilbert space $H_I = \bigotimes_{i \in I} \mathbb{C}^2$ – since we use integers as labels, these products are always taken in some specific order. If $I = \emptyset$, we set $H_I = \mathbb{C}$. We assume there is an infinite number of different labels, i.e. that the number of qbits involved, while always finite, can be as large as desired. Most of the operations are explicit analogues of their counterparts in the measurement calculus. The only Boolean operation needed in the measurement calculus is the exclusive-or $\oplus$ operation. Boolean values are introduced as terms, either directly as the constants 0, 1, or indirectly as measurement results stored in signal variables which are accessed by the dereferenciation operation !. The angle operation + is the addition (modulo 2π) of two angles. The operation $\mathsf{rot}\, \alpha\, b_1\, b_2$ takes the angle α to $(-1)^{b_1} \alpha + b_2 \pi$, where b_1 and b_2 are two Boolean values. This operation is used in the measurement calculus when signals are used to modify a measurement angle, using the notation $[M^\alpha_i]^{s,t}$. Since we use conditional rotation operation, the qbit measurement operation does not need the signal input present in the measurement calculus. Finally, the above syntax introduces analogues of the conditional correction commands X and Z and of the entanglement command E, together with qbit constant terms and variables. Qbit variables are necessary to be able to represent terms with unspecified input.

The type system uses four base types:

$$T ::= \text{angle} \mid \text{bool} \mid \text{qbit}^I \mid \text{signal}^i_I$$

The types angle and bool are the classical types of angles and Boolean values, and the qbit^I type is the type of qbit states over H_I. Signals are stores for Boolean values. The labels associated to the type signal^i_I specify to which qbit i in I the signal is associated to, i.e. the qbit that can be measured to change the value of the signal.

A *context* Γ is a partial function assigning types to variables: it is written as a list of type assignments $x_1 \colon T_1, \ldots, x_n \colon T_n$. Note that such a list cannot refer more that once to a given variable. A typing judgement is a triple $\Gamma \vdash M \colon T$ consisting of a context Γ, a term M and a type T. We must give rules so that one can infer the type of a term M in a context Γ from basic type assumptions. Before doing this, note that measurements are destructive in the measurement calculus and thus that it should not be possible to reuse the label of a previously destroyed qbit. We enforce this formally by keeping track of the unused labels of a term M; the set unused labels in M is denoted UL(M). We consider the measurement and dereferencing operations binding on signal variables: a variable s is free in M if it does not occur in M as an argument to a meas or a ! operation. The set of free variables of M is denoted FV(M).

The typing rules of MCdata are described in table 4.1. Note that we only allow variables of type signal^i_I and qbit^I, since it is not possible in the measurement calculus to have an unspecified Boolean or angle.

Table 4.1 MCdata typing rules

Constants

$$\frac{}{\Gamma, x\colon T \vdash x\colon T}\ \begin{array}{l} x \in \mathrm{Var} \\ T = \mathrm{signal}^i_I \text{ or } \mathrm{qbit}^I \end{array} \qquad \frac{}{\Gamma \vdash \alpha\colon \mathrm{angle}}\ \alpha \in [0, 2\pi)$$

$$\frac{}{\Gamma \vdash b\colon \mathrm{bool}}\ b \in \{0,1\} \qquad \frac{}{\Gamma \vdash |\phi\rangle^I\colon \mathrm{qbit}^I}\ |\phi\rangle \in H_I$$

Classical operations

$$\frac{\Gamma \vdash s\colon \mathrm{signal}^i_I}{\Gamma \vdash !s\colon \mathrm{bool}} \qquad \frac{\Gamma \vdash W\colon \mathrm{angle} \quad \Gamma \vdash B_1\colon \mathrm{bool} \quad \Gamma \vdash B_2\colon \mathrm{bool}}{\Gamma \vdash \mathsf{rot}\, W\, B_1\, B_2\colon \mathrm{angle}}$$

$$\frac{\Gamma \vdash B_1\colon \mathrm{bool} \quad \Gamma \vdash B_2\colon \mathrm{bool}}{\Gamma \vdash B_1 \oplus B_2\colon \mathrm{bool}} \qquad \frac{\Gamma \vdash W_1\colon \mathrm{angle} \quad \Gamma \vdash W_2\colon \mathrm{angle}}{\Gamma \vdash W_1 + W_2\colon \mathrm{angle}}$$

Quantum operations

$$\frac{\Gamma \vdash B\colon \mathrm{bool} \quad \Gamma \vdash Q\colon \mathrm{qbit}^{I\cup\{i\}}}{\Gamma \vdash \mathsf{X}_i\, B\, Q\colon \mathrm{qbit}^{I\cup\{i\}}} \qquad \frac{\Gamma \vdash B\colon \mathrm{bool} \quad \Gamma \vdash Q\colon \mathrm{qbit}^{I\cup\{i\}}}{\Gamma \vdash \mathsf{Z}_i\, B\, Q\colon \mathrm{qbit}^{I\cup\{i\}}}$$

$$\frac{\Gamma \vdash s\colon \mathrm{signal}^i_I \quad \Gamma \vdash W\colon \mathrm{angle} \quad \Gamma \vdash Q\colon \mathrm{qbit}^{I\cup\{i\}}}{\Gamma \vdash \mathsf{meas}^i_I\, s\, W\, Q\colon \mathrm{qbit}^I}\ s \in \mathrm{FV}(W) \cap \mathrm{FV}(Q)$$

$$\frac{\Gamma \vdash Q\colon \mathrm{qbit}^{I\cup\{i,j\}}}{\Gamma \vdash \mathsf{E}_{ij}\, Q\colon \mathrm{qbit}^{I\cup\{i,j\}}} \qquad \frac{\Gamma \vdash Q : \mathrm{qbit}^I}{\Gamma \vdash \mathsf{prep}_i\, Q\colon \mathrm{qbit}^{I\cup\{i\}}}\ i \in \mathrm{UL}(Q)$$

An MCdata *pattern* is an MCdata term M for which we can derive a typing judgement of the form

$$s_1\colon \mathrm{signal}^{i_1}_{I_1}, \ldots, s_n\colon \mathrm{signal}^{i_n}_{I_n}, x\colon \mathrm{qbit}^{\mathrm{In}} \vdash M\colon \mathrm{qbit}^{\mathrm{Out}}.$$

where $\mathrm{In}, \mathrm{Out}, J_j \subseteq I$. To clarify the notation, we label signal variables with the label i of the associated measured qbit: s_i is the signal where is stored the measurement result of a qbit of type qbit^i. Note also that we need to use qbit variables in order to describe the unspecified input to a pattern. This is because we do not have access to higher-order types such as "$\mathrm{qbit}^{\mathrm{In}} \rightarrow \mathrm{qbit}^{\mathrm{Out}}$."

As the following examples show, it is easy to write a measurement calculus pattern as a MCdata term.

Example 4.1. The MCdata form of the measurement calculus pattern for the Hadamard operation is

$$\mathrm{Hadamard}_{12} = \mathsf{X}_2\, !s\, \mathsf{meas}^1_{12}\, s\, 0\, \mathsf{E}_{12}\, \mathsf{prep}_2\, x.$$

Note that we have left all parentheses implicit: the Hadamard term is

$$\mathsf{X}_2\, (!s)\, \left(\mathsf{meas}^1_{12}\, s\, 0\, (\mathsf{E}_{12}\, (\mathsf{prep}_2\, x))\right).$$

The type of this term is

$$s\colon \text{signal}^1_{12}, x\colon \text{qbit}^1 \vdash \text{Hadamard}_{12}\colon \text{qbit}^2$$

Example 4.2. The controlled-not operation $\wedge X$ is represented in MCdata as the pattern

$$\text{CNot}_{124} = \mathsf{X}_4\, !s_3\, \mathsf{Z}_4\, !s_2\, \mathsf{Z}_1\, !s_2\, \mathsf{meas}_3\, s_3\, 0\, \mathsf{meas}_2\, s_2\, 0\, \mathsf{E}_{34}\, \mathsf{E}_{23}\, \mathsf{E}_{13}\, \mathsf{prep}_3 \mathsf{prep}_4\, x$$

The type of this pattern is

$$x\colon \text{qbits}^{12}, s_2\colon \text{signal}^2_{1234}, s_3\colon \text{signal}^3_{134} \vdash \text{CNot}_{124}\colon \text{qbit}^{14}$$

Example 4.3. The *teleportation* pattern, which takes some state in H_1 and transfers it to H_3:

$$\text{Teleport}_{13} = \mathsf{X}_3\, !s_2\, \mathsf{Z}_3\, !s_1\, \mathsf{meas}^2_{23}\, s_2\, 0\, \mathsf{meas}^1_{123}\, s_1\, 0\, \mathsf{E}_{23}\, \mathsf{E}_{12}\, \mathsf{prep}_3\, \mathsf{prep}_2\, x$$

We can derive the following typing judgement using the above rules:

$$x : \text{qbit}^1, s_1 : \text{signal}^1_{123}, s_2\colon \text{signal}^2_{23} \vdash \text{Teleport}_{13}\colon \text{qbit}^3$$

The type system restricts MCdata terms to those corresponding to patterns.

Lemma 4.4. *If* $\Gamma \vdash M\colon T$ *is an MCdata valid typing judgement, then* Γ *contains at most one variable of type* qbit^I *which is used in* M

Proof. By induction on the derivation of $\Gamma \vdash M\colon T$.

If M is a constant or a variable term, the result is trivial.

By inspection of the other derivation rules, we can see that in all cases the hypothesis contains at most one term of type qbit^J, which, by induction hypothesis, contains at most one variable of type qbit^I. The term in the conclusion thus also has this property. □

We henceforth assume without loss of generality that there is at most one qbit variable in a context, i.e. that all contexts Γ are of the form

$$s_1\colon \text{signal}^{i_1}_{I_1}, \ldots, s_n\colon \text{signal}^{i_n}_{I_n}, x\colon \text{qbit}^{\text{In}}.$$

Consider a qbit MCdata term M:

$$s_1\colon \text{signal}^{i_1}_{I_1}, \ldots, s_n\colon \text{signal}^{i_n}_{I_n}, x\colon \text{qbit}^{\text{In}} \vdash M\colon \text{qbit}^{\text{Out}}.$$

Does the MCdata type system force M to correspond to a measurement calculus pattern? While we can construct from M a sequence of measurement calculus commands, we need to check that it will always satisfy the three defining conditions of patterns.

1. Assume a signal $s\colon \text{signal}^i_I$ is used in M. According to the typing rules, the only place where s can be used in either in a dereferencing operation or in a measurement operation. To satisfy the first defining condition of measurement calculus patterns, s must not be used in the arguments of a measurement operation that assigns a value to s. The measurement operation typing rule explicitly forbids this.

2. Since the type system does not allow to the introduction of a new qbit label i to a term N with prep_i, when i has been used previously in the construction of N, no command can be applied to i, if i is measured in the measurement calculus pattern associated to M.

3. For similar reasons, if $i \in \mathrm{Out}$, then i cannot be measured in M. If a qbit $i \notin \mathrm{Out}$ is used in M, then i must be measured at some point since the measurement typing rule is the only one where labels are removed from qbits.

4.1.2 Operational semantics

The operational semantics we give in this section is a direct adaptation to MCdata of the semantics given in [DKP07]. A **store** is a partial function $\Sigma : \mathrm{Vars} \to \{0, 1\}$ taking variables to Boolean values. Stores are modified as follows:

$$\Sigma[s \mapsto b](t) = \begin{cases} b \text{ if } t = s \\ \Sigma(t) \text{ otherwise.} \end{cases}$$

A **MCdata state** is a pair Σ, M where Σ is a store and M is a MCdata term. A **canonical form** Σ, V is a pair with a store Σ and a constant term V.

The operational semantics is given by a probabilistic reduction relation $\Sigma, M \Downarrow^p \Sigma', V$, where V is a canonical form and $p \in [0, 1]$ is the probability of the reduction occurrence. The parameter p is omitted when it is 1. The reduction rules are described in table 4.2.

Note that the only place where the reduction rules allow probabilistic branching is in the two rules for measurements.

4.2 Denotational semantics

We now turn to the problem of applying the ideas exposed in the last chapter to construct a game-based interpretation of MCdata terms. We want to define a map $[\![-]\!]$ on types and on typing judgements such that $[\![\Gamma \vdash M : T]\!]$ is a strategy in $[\![\Gamma]\!] \multimap [\![T]\!]$. Because of the presence of the preparation command, we need to allows POVM quantum strategies as introduced in section 3.6.2. Note that the fact that there can be only one POVM measurement per play is not problematic since in the measurement calculus each qbit is measured at most once.

Each type of MCdata is interpreted as a quantum arena:

$$[\![\mathrm{bool}]\!] = \mathbf{bool} \qquad [\![\mathrm{angle}]\!] = \mathbf{angle} \qquad \left[\!\left[\mathrm{qbit}^I\right]\!\right] = \mathbf{qbit}^I = [H_I]$$
$$\left[\!\left[\mathrm{signal}^i_I\right]\!\right] = \mathbf{signal}^i_I = \left(\mathbf{angle} \odot \mathbf{qbit}^{I \cup \{i\}} \multimap \mathbf{qbit}^I\right) \odot \mathbf{bool}$$

The **bool** arena is the classical flat arena defined in section 2.3.4. The **angle** arena is the flat arena over $[0, 2\pi)$, defined in a similar way as the **bool** arena. The interpretation of the qbit^I uses the quantum arena $[H_I]$ which is the tensor product of arenas $[\mathbb{C}^2]$, one for each index in I.

The interpretation the signal type is more complex. Since we consider signals as stores, we use an interpretation similar to what is used in the case of classical stores. A

Table 4.2 MCdata reduction rules

Constants

$$\frac{}{\Sigma, \alpha \Downarrow \Sigma, \alpha}\ \alpha \in [0, 2\pi) \qquad \frac{}{\Sigma, 0 \Downarrow \Sigma, 0} \qquad \frac{}{\Sigma, 1 \Downarrow \Sigma, 1}$$

$$\frac{}{\Sigma, x \Downarrow \Sigma, x}\ x \in \mathrm{Var} \qquad \frac{}{\Sigma, |\phi\rangle^I \Downarrow \Sigma, |\phi\rangle^I}$$

Classical operations

$$\frac{}{\Sigma, !s \Downarrow \Sigma, \Sigma(s)}$$

$$\frac{\Sigma, W \Downarrow^p \Sigma', \alpha \qquad \Sigma', B_1 \Downarrow^{q_1} \Sigma'', b_1 \qquad \Sigma'', B_2 \Downarrow^{q_2} \Sigma''', b_2}{\Sigma, q, \mathsf{rot}\ W\, B_1\, B_2 \Downarrow^{pq_1q_2} \Sigma''', \beta}$$

$$\alpha \in [0, 2\pi),\ b_1, b_2 \in \{0, 1\} \text{ and } \beta = \alpha^{b_1} + b_2\pi$$

$$\frac{\Sigma, B_1 \Downarrow^p \Sigma', b_1 \qquad \Sigma', B_2 \Downarrow^q \Sigma'', b_2}{\Sigma, B_1 \oplus B_2 \Downarrow^{pq} \Sigma'', b}\ b_1, b_2 \in \{0, 1\},\ b = b_1 \operatorname{xor} b_2$$

$$\frac{\Sigma, W_1 \Downarrow^p \Sigma', \alpha_1 \qquad \Sigma', W_2 \Downarrow^q \Sigma'', \alpha_2}{\Sigma, W_1 + W_2 \Downarrow^{pq} \Sigma'', \beta}\ \alpha_1, \alpha_2 \in [0, 2\pi), \beta = \alpha_1 + \alpha_2$$

Quantum operations

$$\frac{\Sigma, B \Downarrow^p \Sigma', b \qquad \Sigma', Q \Downarrow^q \Sigma'', |\phi\rangle^I}{\Sigma, \mathsf{X}_i\, B\, Q \Downarrow^{pq} \Sigma'', ([X_i]^b|\phi\rangle)^I} \qquad \frac{\Sigma, B \Downarrow^p \Sigma', b \qquad \Sigma', Q \Downarrow^q \Sigma'', |\phi\rangle^{I\cup\{i\}}}{\Sigma, \mathsf{Z}_i\, B\, Q \Downarrow^{pq} \Sigma'', ([Z_i]^b|\phi\rangle)^{I\cup\{i\}}}$$

$$\frac{\Sigma, W \Downarrow^p \Sigma', \alpha \qquad \Sigma', Q \Downarrow^q \Sigma'', |\phi\rangle^{I\cup\{i\}}}{\Sigma, \mathsf{meas}_I^i\ s\, W\, Q \Downarrow^{pqr} \Sigma[s \mapsto 1], \frac{1}{r}\langle +_\alpha|\phi\rangle^I}\ r = |\langle +_\alpha|\phi\rangle|^2$$

$$\frac{\Sigma, W \Downarrow^p \Sigma', \alpha \qquad \Sigma', Q \Downarrow^q \Sigma'', |\phi\rangle^{I\cup\{i\}}}{\Sigma, \mathsf{meas}_I^i\ s\, W\, Q \Downarrow^{pqr} \Sigma[s \mapsto 0], \frac{1}{r}\langle -_\alpha|\phi\rangle^I}\ r = |\langle -_\alpha|\phi\rangle|^2$$

$$\frac{\Sigma, Q \Downarrow^p \Sigma', |\phi\rangle^{I\cup\{i,j\}}}{\Sigma, \mathsf{E}_{ij}\, Q \Downarrow^p \Sigma', (\wedge Z_{ij}|\phi\rangle)^{I\cup\{i,j\}}} \qquad \frac{\Sigma, Q \Downarrow^p \Sigma', |\phi\rangle^I}{\Sigma, \mathsf{prep}_i\, Q \Downarrow^p \Sigma', (|+\rangle \otimes |\phi\rangle)^{I\cup\{i\}}}$$

classical store s comes equipped with two methods: one to write a value to the store, one to read a value from the store. The type of a classical store for Boolean values is taken to be

$$(\mathbf{bool} \multimap \mathbf{com}) \odot \mathbf{bool},$$

where **com** is the arena for command types, where the only possible interaction is of the form “run done”. Opponent asks that the command be run, and Player confirms that this has been done. With this definition, the write and read operations can be respectively interpreted as the projections on the **bool** ⊸ **com** and **bool** arenas. Intuitively, the write operation is a command that takes a value as input and is executed returning no value, and the read operation simply returns a Boolean value. The definition of the signal arena given below differs from this because we consider the measurement operation as a variant of the classical write command. Since the measurement operation takes an angle and a qbit$^{I\cup\{i\}}$ value and returns a qbit^I value, the write part must be replaced by the arena

$$\mathbf{angle} \odot \mathbf{qbit}^{I\cup\{i\}} \multimap \mathbf{qbit}^{I}.$$

We want the two projection strategies to correspond to measurement

$$\mathsf{meas}^{i}_{I} \colon \mathbf{signal}^{i}_{I} \multimap \left(\left(\mathbf{angle} \odot \mathbf{qbit}^{I\cup\{i\}}\right) \multimap \mathbf{qbit}^{I}\right)$$

and to dereferencing

$$\mathsf{deref} : \mathbf{signal}^{i}_{I} \to \mathbf{bool}.$$

We thus use the arena

$$\mathbf{signal}^{i}_{I} = \left(\left(\mathbf{angle} \odot \mathbf{qbit}^{I\cup\{i\}}\right) \multimap \mathbf{qbit}^{I}\right) \odot \mathbf{bool}$$

as the interpretation of the type signal^{i}_{I}.

A term M is said to be semi-closed if we can derived a typing judgement of the form $\Gamma \vdash M \colon T$ with Γ containing only signal variables. The interpretation of a semi-closed term $\Gamma \vdash M \colon T$ is a strategy $[\![M]\!] : [\![\Gamma]\!] \to [\![T]\!]$ which is defined by induction on the derivation of typing judgements.

All constants are interpreted as their corresponding strategies as described in the previous chapters: the Boolean values 0 and 1 are interpreted as the strategies $0, 1 \colon \mathbf{bool}$ and similarly an angle $\alpha \in [0, 2\pi)$ is interpreted as the strategy $\alpha \colon \mathbf{angle}$ with typical play

$$\begin{array}{c} \mathbf{angle} \\ ? \\ \alpha \end{array}$$

A quantum state term $\Gamma \vdash |\varphi\rangle^{I} \colon \mathrm{qbit}^{I}$ is interpreted as the strategy $\left[|\varphi\rangle^{I}\right]$ in $[\![\Gamma]\!] \multimap \mathbf{qbit}^{I}$ which ignore the $[\![\Gamma]\!]$ component.

The dereferencing operation $!s$ is interpreted as the projection strategy deref described above:

$$[\![\Gamma \vdash !s \colon \mathrm{bool}]\!] = [\![s]\!] \,; \mathsf{deref}.$$

The classical operations are interpreted using deterministic strategies rot, xor and $\mathsf{addAngle}$ with the obvious definition:

$$\begin{aligned} [\![\Gamma \vdash \mathsf{rot}\, W B_1 B_2 \colon \mathsf{angle}]\!] &= \langle [\![W]\!] , [\![B_1]\!] , [\![B_2]\!] \rangle ; \mathsf{rot} \\ [\![\Gamma \vdash B_1 \oplus B_2 \colon \mathsf{bool}]\!] &= \langle [\![B_1]\!] , [\![B_2]\!] \rangle ; \mathsf{xor} \\ [\![\Gamma \vdash W_1 + W_2 \colon \mathrm{angle}]\!] &= \langle [\![W_1]\!] , [\![W_2]\!] \rangle ; \mathsf{addAngle} \end{aligned}$$

In each case, Player queries Opponent about each required input datum and produces a final answer in the output component.

Conditional corrections are interpreted as follows:

$$\begin{aligned} \left[\!\!\left[\Gamma \vdash X_a B Q \colon \mathrm{qbit}^{I\cup\{i\}}\right]\!\!\right] &= \langle [\![B]\!] , [\![Q]\!] \rangle ; \mathsf{condX}_i \\ \left[\!\!\left[\Gamma \vdash Z_a B Q \colon \mathrm{qbit}^{I\cup\{i\}}\right]\!\!\right] &= \langle [\![B]\!] , [\![Q]\!] \rangle ; \mathsf{condZ}_i \end{aligned}$$

Table 4.3 MCdata denotational semantics

$$\left\llbracket \Gamma, x\colon T \vdash x\colon \mathrm{signal}^i_I \right\rrbracket : \ \llbracket \Gamma \rrbracket \odot \mathbf{signal}^i_I \xrightarrow{\pi_{\mathbf{signal}^i_I}} \mathbf{signal}^i_I$$

$$\left\llbracket \Gamma, x\colon T \vdash x\colon \mathrm{qbit}^I \right\rrbracket : \ \llbracket \Gamma \rrbracket \odot \mathbf{qbit}^I \xrightarrow{\pi_{\mathbf{qbit}^I}} \llbracket T \rrbracket$$

$$\llbracket \Gamma \vdash \alpha\colon \mathrm{angle} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\alpha} \mathbf{angle}$$

$$\llbracket \Gamma \vdash 0\colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{0} \mathbf{bool}$$

$$\llbracket \Gamma \vdash 1\colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{1} \mathbf{bool}$$

$$\left\llbracket \Gamma \vdash |\phi\rangle^I\colon \mathrm{qbit}^I \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{[|\phi\rangle^I]} \mathbf{qbit}^I$$

$$\llbracket \Gamma \vdash !s\colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\pi} \mathbf{signal}^i_I \xrightarrow{\mathsf{deref}} \mathbf{signal}^i_I$$

$$\llbracket \Gamma \vdash \mathsf{rot}\, W\, B_1\, B_2\colon \mathrm{angle} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket W \rrbracket, [[B_1]], [[B_2]] \rangle} \mathbf{angle} \odot \mathbf{bool} \odot \mathbf{bool} \xrightarrow{\mathsf{rot}} \mathbf{angle}$$

$$\llbracket \Gamma \vdash B_1 \oplus B_2\colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle [[B_1]], [[B_2]] \rangle} \mathbf{bool} \odot \mathbf{bool} \xrightarrow{\mathsf{xor}} \mathbf{bool}$$

$$\llbracket \Gamma \vdash W_1 + W_2\colon \mathrm{angle} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle [[W_1]], [[W_2]] \rangle} \mathbf{angle} \odot \mathbf{angle} \xrightarrow{\mathsf{addAngle}} \mathbf{angle}$$

$$\left\llbracket \Gamma \vdash \mathsf{X}_i\, B\, Q\colon \mathrm{qbit}^{I \cup \{i\}} \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket B \rrbracket, \llbracket Q \rrbracket \rangle} \mathbf{bool} \odot \mathbf{qbit}^{I \cup \{i\}} \xrightarrow{\mathsf{condX}_i} \mathbf{qbit}^{I \cup \{i\}}$$

$$\left\llbracket \Gamma \vdash \mathsf{Z}_i\, B\, Q\colon \mathrm{qbit}^{I \cup \{i\}} \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket B \rrbracket, \llbracket Q \rrbracket} \mathbf{bool} \odot \mathbf{qbit}^{I \cup \{i\}} \xrightarrow{\mathsf{condZ}_i} \mathbf{qbit}^{I \cup \{i\}}$$

$$\left\llbracket \Gamma \vdash \mathsf{meas}^i_I\, s\, W\, Q\colon \mathrm{qbit}^I \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket s \rrbracket, \llbracket W \rrbracket, \llbracket Q \rrbracket \rangle} \mathbf{signal}^i_I \odot \mathbf{angle} \odot \mathbf{qbit}^{I \cup \{i\}} \xrightarrow{\Lambda^{-1}(\mathsf{meas}^i_I)} \mathbf{qbit}^I$$

$$\left\llbracket \Gamma \vdash \mathsf{E}_{ij}\, Q\colon \mathrm{qbit}^{I \cup \{i,j\}} \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\llbracket Q \rrbracket} \mathbf{qbit}^{I \cup \{i,j\}} \xrightarrow{[\wedge Z_{ij}]} \mathbf{qbit}^{I \cup \{i,j\}}$$

$$\left\llbracket \Gamma \vdash \mathsf{prep}_i\, Q\colon \mathrm{qbit}^{I \cup \{i\}} \right\rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\llbracket Q \rrbracket} \mathbf{qbit}^I \xrightarrow{\mathsf{prep}_i} \mathbf{qbit}^{I \cup \{i\}}$$

Both strategies

$$\mathsf{condX}_i, \mathsf{condZ}_i \colon \mathbf{bool} \odot \mathbf{qbit}^{I\cup\{i\}} \to \mathbf{qbit}^{I\cup\{i\}}$$

are defined as follows: if Opponent begins with $\mathcal{A}_?$ in the output component, Player asks Opponent for a Boolean in the **bool** component, and then asks either $\left(X_i\mathcal{A}X_i^\dagger\right)_?$ or $\mathcal{A}_?$ when he is answered 1 or 0 respectively. He finally copies the final Opponent's answer to the output component.

$$\begin{array}{ccccc}
\mathbf{bool} & \odot & \mathbf{qbit}^{I\cup\{i\}} & \multimap & \mathbf{qbit}^{I\cup\{i\}} \\
 & & & & \mathcal{A}_? \\
? & & & & \\
b & & & & \\
 & & \left([X_i]^b\mathcal{A}[X_i^\dagger]^b\right)_? & & \\
 & & m & & \\
 & & & & m
\end{array}$$

Note that if we had the resources of higher order language, the conditional corrections could be interpreted using an "if then else" construct. In the case of a X correction, this would give:

$$\mathsf{if}\, B\, \mathsf{then}\, X\, \mathsf{else}\ \mathsf{id}\,.$$

In in classical higher-order game semantics, conditionals are interpreted using a strategy cond and pairing:

$$[\![\mathsf{if}\, B\, \mathsf{then}\, M\, \mathsf{else}\, N]\!] = \langle [\![B]\!]\,, [\![M]\!]\,, [\![N]\!]\rangle; \mathsf{cond}$$

where Player, when using the cond strategy, probes Opponent about the input bit, and then probes again in either the second or third component to copy Opponent's answers in the output component.

The measurement commands are interpreted using the adjunction bijection Λ between strategies in $A \odot B \to C$ and those in $A \to B \multimap C$. The projection meas^i_I has the adjoint

$$\Lambda^{-1}(\mathsf{meas}^i_I)\colon \mathbf{signal}^i_I \odot \mathbf{angle} \odot \mathbf{qbit}^{I\cup\{i\}} \to \mathbf{qbit}^I.$$

The denotation of the measurements commands is defined by

$$\left[\!\!\left[\Gamma \vdash \mathsf{meas}^i_I\, s\, W Q \colon \mathrm{qbit}^I\right]\!\!\right] = \langle [\![s]\!]\,, [\![W]\!]\,, [\![Q]\!]\rangle; \Lambda^{-1}\left(\mathsf{meas}^i_{I\cup\{i\}}\right).$$

Entanglement operations are interpreted using the unitary operation strategies defined in section 3.18:

$$\left[\!\!\left[\Gamma \vdash \mathsf{E}_{ij}\, Q \colon \mathrm{qbit}^{I\cup\{i,j\}}\right]\!\!\right] = [\![Q]\!]\,; [\wedge Z_{ij}]$$

The interpretation of typing judgements ending with the preparation of a new qbit is defined using the preparation strategy

$$\mathsf{prep}_i \colon \mathbf{qbit}^I \multimap \mathbf{qbit}^{I\cap\{i\}}$$

as described in section 3.6.2. Suppose we are given an interpretation $\left[\!\left[\Gamma \vdash Q\colon \mathrm{qbit}^{I}\right]\!\right]$ with

$$\Gamma \vdash Q\colon \mathrm{qbit}^{I \cup \{i\}}.$$

The strategy

$$\left[\!\left[\Gamma \vdash \mathsf{prep}_i\, Q\colon \mathrm{qbit}^{I \cup \{i\}}\right]\!\right] :\ [\![\Gamma]\!] \multimap \mathbf{qbit}^{I \cup \{i\}}$$

is defined as the composition $[\![Q]\!]\,; \mathsf{prep}_i$.

This completes the definition of the denotational semantics.

Example 4.5. Consider the interpretation of the Hadamard given in example 4.1, but for a fixed input qbit $|\varphi\rangle^1$:

$$\left[\!\left[s\colon \mathrm{signal}^1_{12} \vdash \mathsf{X}_2\, !s\, \mathsf{meas}^1_{12}\, s\, 0\, \mathsf{E}_{12}\, \mathsf{prep}_2\, |\varphi\rangle^1 : \mathrm{qbit}^2\right]\!\right]$$

is equal to the following composition:

$$\begin{tikzcd}
\mathbf{signal}^1_{12} \arrow[d, "\Delta"] \\
\mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \arrow[d, "\mathrm{id} \odot \Delta"] \\
\mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \arrow[d, "\mathrm{id} \odot \mathrm{id} \odot \Delta"] \\
\mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \arrow[d, "\mathrm{id} \odot \mathrm{id} \odot \mathrm{id} \odot {[\![|\varphi\rangle^1]\!]}"] \\
\mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{qbit}^1 \arrow[d, "\mathrm{id} \odot \mathrm{id} \odot {[\![0]\!]} \odot \mathsf{prep}_2"] \\
\mathbf{signal}^1_{12} \odot \mathbf{signal}^1_{12} \odot \mathbf{angle} \odot \mathbf{qbit}^{12} \arrow[d, "\mathsf{deref} \odot \Lambda^{-1}(\mathsf{meas}^i_I)"] \\
\mathbf{bool} \odot \mathbf{qbit}^2 \arrow[d, "\mathsf{condX}_2"] \\
\mathbf{qbit}^2
\end{tikzcd}$$

4.3 Soundness

Since our goal is to show that the denotational semantics matches the operational semantics of MCdata, we need to take stores into account in the denotational semantics.

To this end, we need to define a strategy $\mathsf{sig}\colon I \multimap \mathbf{signal}^i_I$ that behaves in a way that encodes the behaviour of a signal interacting with its environment.

Assume that a signal is initially set to $b_1 \in \{0,1\}$. A typical play in the signal arena where Player is using the deterministic strategy sig_{b_1} is

$$\begin{array}{cccccc}
(\mathbf{angle} & \odot & \mathbf{qbit}^{I\cup\{i\}} & \longrightarrow & \mathbf{qbit}^{I}) & \odot\ \mathbf{bool} \\
 & & & & & ? \\
 & & & & & b_1 \\
 & & & & \mathcal{A}_? & \\
? & & & & & \\
\alpha & & & & & \\
 & & \mathcal{A}_? \otimes \left\{[+_\alpha]^i_0, [-_\alpha]^i_1\right\}_? & & & \\
 & & m, b_2 & & & \\
 & & & & m & \\
 & & & & & ? \\
 & & & & & b_2
\end{array}$$

In the typical play, Player answers question in the Bool component using the initial value b_1 and answers any question $\mathcal{A}_?$ about the output qbits using the measurements results he gets from Opponent when she is asked to measure the input qbit at the required angle. New Opponent questions in the bool component are answered using the measurement result b_2 in for qbit i.

Let Γ be the context

$$s_1\colon \mathrm{signal}^1_{I_1}, \ldots, s_n\colon \mathrm{signal}^n_{I_n}.$$

A **Γ-store** is a store Σ defined exactly for the variables $s_1, \ldots, s_n$. If Σ is a Γ-store, we define $[\![\Sigma]\!]$ to be the product strategy

$$\langle \mathsf{sig}_{\Sigma(s_1)}, \ldots, \mathsf{sig}_{\Sigma(s_n)} \rangle \colon I \to [\![\Gamma]\!] \,.$$

A pair Σ, M, with $\Gamma \vdash M\colon T$ semi-closed and Σ a Γ-store, is interpreted as

$$[\![\Sigma, M]\!] = [\![\Sigma]\!] \,;\, [\![M]\!] \,.$$

The next proposition says that if some term M reduces to some value V with probability p when starting with a store Σ, then, if Player is using the strategies associated to M and V in their respective contexts, he behaves in the same way in both cases with probability p:

Proposition 4.6. *If $\Sigma, M \Downarrow^p \Sigma', V$, then for all well-opened $sab \in \mathcal{T}\,([\![\Sigma', V]\!])$ we have that*

$$[\![\Sigma, M]\!]\,(b \mid sa) = p\,[\![\Sigma', V]\!]\,(b \mid sa)$$

Let us consider a simple example of such an equality between probabilistic strategies in the arena **bool**. Consider the strategy true and the strategy coin which makes Player answer the initial question with 0 or 1 randomly with uniform probability. The fact that Player behaves the same way in both cases with probability 1/2 can be written as in the statement of the last proposition:

$$\mathsf{coin}(b \mid s?) = \frac{1}{2}\,\mathsf{true}(b \mid s?).$$

We prove proposition 4.6 using a stronger proposition which is proven by induction on the derivation of $\Sigma, M \Downarrow^p \Sigma', V$, using

$$\llbracket \Sigma, M \rrbracket' = \llbracket \Sigma \rrbracket \,;\Delta; (\llbracket M \rrbracket \odot \mathrm{id}_{\llbracket \Gamma \rrbracket})$$

instead of $\llbracket \Sigma, M \rrbracket$. This stronger proposition is that given $\Sigma, M \Downarrow^p \Sigma', V$, we have

$$\llbracket \Sigma, M \rrbracket' \,(b \mid sa) = p\,\llbracket \Sigma', V \rrbracket' \,(b \mid sa)$$

for any well-opened play in $\mathcal{T}(\llbracket \Sigma', V \rrbracket')$ starting in $\llbracket T \rrbracket$. Since these plays are the same as those of $\llbracket \Sigma, M \rrbracket$, the proposition follows directly.

Proof. We prove the result by induction on the derivation of $\Sigma, M \Downarrow^p \Sigma', V$, but we need to prove it using a stronger hypothesis. We define $\llbracket \Sigma, M \rrbracket'$ to be the morphism

$$\begin{tikzcd} I \arrow[r, "\llbracket \Sigma \rrbracket"] & \llbracket \Gamma \rrbracket \arrow[r, "\Delta"] & \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[r, "\llbracket M \rrbracket \odot \mathrm{id}_{\llbracket \Gamma \rrbracket}"] & \llbracket T \rrbracket \odot \llbracket \Gamma \rrbracket \end{tikzcd}$$

and we show by induction that given $\Sigma, M \Downarrow^p \Sigma', V$, we have

$$\llbracket \Sigma, M \rrbracket' \,(s) = p\,\llbracket \Sigma', V \rrbracket' \,(s)$$

for any well-opened play in $\mathcal{T}(\llbracket \Sigma', V \rrbracket')$ starting in $\llbracket T \rrbracket$. Using the variant $\llbracket \Sigma, M \rrbracket'$ allows one to access the $\llbracket \Sigma \rrbracket$ strategy from the output arena, which is not possible if we use $\llbracket \Sigma, M \rrbracket$.

For constants, the result is immediate.

We show how a typical induction case is handled. Suppose the proposition holds when $\Sigma, B_1 \Downarrow^p \Sigma', b_1$ and $\Sigma', B_2 \Downarrow^q \Sigma'', b_2$, and that we want to prove it when

$$\Sigma, B_1 \oplus B_2 \Downarrow^{pq} \Sigma'', b,$$

with $b = b_1 \oplus b_2$. Let $sab \in \mathcal{T}(\llbracket \Sigma'', b \rrbracket')$ with a being a move in **bool**. The fact that the diagram

$$\begin{tikzcd} \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[r, "\mathrm{id} \odot \Delta"] \arrow[d, "\Delta \odot \mathrm{id}"'] & \llbracket \Gamma \rrbracket \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket) \\ (\llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket) \odot \llbracket \Gamma \rrbracket \arrow[ur, "\alpha"'] & \end{tikzcd}$$

commutes and the functoriality of the $\odot$ arena operation imply that $\llbracket \Sigma, B_1 \oplus B_2 \rrbracket'$ is equal to

$$\begin{tikzcd} I \arrow[r, "\llbracket \Sigma \rrbracket"] & \llbracket \Gamma \rrbracket \arrow[d, "\Delta"'] & \mathbf{bool} \odot \llbracket \Gamma \rrbracket \\ & \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[d, "\llbracket B_1 \rrbracket \odot \mathrm{id}"'] & \mathbf{bool} \odot \mathbf{bool} \odot \llbracket \Gamma \rrbracket \arrow[u, "\mathrm{xor} \odot \mathrm{id}"'] \\ & \mathbf{bool} \odot \llbracket \Gamma \rrbracket \arrow[r, "\mathrm{id} \odot \Delta"'] & \mathbf{bool} \odot \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[u, "\mathrm{id} \odot \llbracket B_2 \rrbracket \odot \mathrm{id}"'] \end{tikzcd}$$

Consider a witness u of sa in the above composition. The first move of u must be a question in the final **bool** arena. The xor strategy copies this question to its left **bool** input arena, and the identity strategies copies it to a question in the output **bool** component of $\llbracket \Sigma, B_1 \rrbracket'$. By hypothesis, this strategy behave like $\llbracket \Sigma', b_1 \rrbracket$ with probability p and leave the Γ-store in the state Σ'. At this point, the xor strategy begins an interaction in its second **bool** input arena, and the play proceeds using $\llbracket \Sigma', B_2 \rrbracket'$. The induction hypothesis implies that $\llbracket \Sigma', B_2 \rrbracket'$ behaves like $\llbracket \Sigma'', b_2 \rrbracket'$ with probability q. Finally, Player following the xor strategy will answer $b = b_1$ xor b_2 in the final **bool** component with probability pq, leaving the Γ-store in state Σ''. This shows that plays starting in the last **bool** component $\llbracket \Sigma, B_1 \oplus B_2 \rrbracket$ behave as $\llbracket \Sigma'', b \rrbracket$ with probability p.

Consider the case of the measurement rule. Suppose that the proposition holds for $\Sigma', W \Downarrow^p \Sigma, \alpha$ and $\Sigma', Q \Downarrow^q \Sigma'', |\varphi\rangle^{I \cup \{i\}}$. We want to show that when

$$\Sigma, \mathsf{meas}^i_I s W Q \Downarrow^{pqr} \Sigma'', |\varphi\rangle^I$$

for all well-opened plays $sab \in \mathcal{T}\left(\left\llbracket \Sigma'', |\varphi\rangle^I \right\rrbracket'\right)$ with a in $\mathbf{qbit}^I$, we have that

$$\left\llbracket \Sigma, \mathsf{meas}^i_I\ s\ W\ Q \right\rrbracket'(b \mid sa) = pqr \left\llbracket \Sigma, |\varphi\rangle^I \right\rrbracket'(b \mid sa),$$

where $r = |\langle \alpha_+ | \varphi \rangle^i|^2$. Similarly to the previous case, we have that

$$\left\llbracket \Sigma, \mathsf{meas}^i_I\ s\ W\ Q \right\rrbracket'$$

is equal to the following composition:

$$\begin{tikzcd}
I \arrow[r, "\llbracket \Sigma \rrbracket"] & \llbracket \Gamma \rrbracket \arrow[d, "\llbracket \Delta \rrbracket"'] & \mathbf{signal}^i_I \odot \mathbf{angle} \odot \mathbf{qbit}^{I \cup \{i\}} \odot \llbracket \Gamma \rrbracket \arrow[r, "\Lambda^{-1}(\mathsf{meas}^i_I) \odot \mathrm{id}"] & \mathbf{qbit}^I \odot \llbracket \Gamma \rrbracket \\
& \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[d, "\pi \odot \mathrm{id}"'] & \mathbf{signal}^i_I \odot \mathbf{angle} \odot \llbracket \Gamma \rrbracket \odot \llbracket \Gamma \rrbracket \arrow[u, "\mathrm{id} \odot \llbracket Q \rrbracket \odot \mathrm{id}"'] & \\
& \mathbf{signal}^i_I \odot \llbracket \Gamma \rrbracket \arrow[r, "\mathrm{id} \odot \Delta"'] & \mathbf{signal}^i_I \odot \mathbf{angle} \odot \llbracket \Gamma \rrbracket \arrow[u, "\mathrm{id} \odot \Delta"'] &
\end{tikzcd}$$

Consider a witness u of sa in the above composition. The first move of u is a question $\mathcal{A}_?$ in the final $\mathbf{qbit}^I$ arena. This is copied to a question in the $\mathbf{qbit}^I$ component of the $\mathbf{signal}^i_I$ arena. This is copied by the identity, projection and diagonal strategies to the $\mathbf{signal}^i_I$ part of the initial Γ. Following $\llbracket \Sigma \rrbracket$, Player asks back a question in the **angle** component, which is copied back to the input signal arena of $\Lambda^{-1}\left(\mathsf{meas}^i_I\right)$. The question is then copied to the **angle** input arena and copied to the output arena of $\llbracket W \rrbracket$. By hypothesis, this question is answered with α with probability p after an interaction which changes the $\llbracket \Sigma \rrbracket$ strategy to a state where it behaves as the strategy $\llbracket \Sigma' \rrbracket$. Note that the s: $\mathbf{signal}^i_I$ part of Γ is not affected by this change, since s cannot be used before the measurement command meas^i_I is introduced. The answer is copied back to the $\mathbf{signal}^i_I$ part of the initial store strategy, now $\llbracket \Sigma \rrbracket'$, where Player uses it to ask for result of the measurement

$$\mathcal{A}_? \otimes \left\{ [+_\alpha]^i_0, [-_\alpha]^i_1 \right\}.$$

This question is copied to the input $\mathbf{qbit}^{I\cup\{i\}}$ of $\Lambda^{-1}\left(\mathsf{meas}_I^i\right)$, and answered using $[\![Q]\!]$. By induction hypothesis, this happens in the same way as using the strategy $\left[\!\left[|\varphi\rangle^{I\cup\{i\}}\right]\!\right]$ with probability qr, where r is the probability that the measurement on qbit i gives the result m. Note this is the point in the interaction where the quantum measurement is actually performed. By hypothesis, after this interaction, Player will use the strategy $[\![\Sigma'']\!]$ in the first part of the composition, and as in the previous angle step, the s part of Σ'' is unaffected.

The answer is copied back to the initial $\mathbf{signal}_I^i$, where afterwards any query using deref will be answered with the i part of the measurement result. This will leave the store strategy behaving as $[\![\Sigma[s \mapsto m]]\!]$, and the I part of the answer is copied to the final $\mathbf{qbit}^I$ arena.

The other cases are treated similarly. □

The next important result about the relation between the operational and denotational semantics of MCdata is *adequacy*, the converse of proposition 4.6.

Proposition 4.7. *(Adequacy for MCdata) Let M be a semi-closed term. If for all well-opened $sab \in \mathcal{T}([\![\Sigma', V]\!])$ we have that*

$$[\![\Sigma, M]\!](b \mid sa) = p\,[\![\Sigma', V]\!](b \mid sa),$$

then $\Sigma, M \Downarrow^p \Sigma', V$ holds.

Proof. By induction on the construction of terms. Assume $\Gamma \vdash M\colon \mathrm{qbit}^{\mathrm{Out}}$, where Γ contains only signal variables. We show how typical cases are dealt with, the other cases being similar.

For the base case, M is either a constant term or a signal variable. In both cases, the result is immediate since M is a value.

Suppose that the proposition holds for B_1 and B_2, two boolean semi-closed terms, i.e. that the strategy $[\![\Sigma, B_1]\!]$ makes Player play the same moves as $[\![\Gamma', b_1]\!]$ and that the strategy $[\![\Sigma', B_2]\!]$ makes Player play the same moves as $[\![\Gamma'', b_2]\!]$. We want to show that the proposition also holds for $\Gamma \vdash B_1 \oplus B_2\colon \mathrm{bool}$. Assume that when player uses the strategy $[\![\Sigma, B_1 \oplus B_2]\!]$ he makes the same choices as if playing using the strategy $[\![\Sigma'', b]\!]$ for some Boolean b. By the definition of $[\![B_1 \oplus B_2]\!]$, Player answer the initial question by starting interactions using successively the strategies $[\![B_1]\!]$ and $[\![B_2]\!]$. Suppose that in these interactions the initial questions are answered by b_1 and b_2 with probability p and q respectively and thus that the final answer Player gives using $[\![B_1 \oplus B_2]\!]$ is $b = b_1 \oplus b_2$. By induction hypothesis, this implies that $\Sigma, B_1 \Downarrow^p \Sigma', b_1$ and $\Sigma', B_2 \Downarrow^q \Sigma'', b_2$. If thus follows from the definition of the operational semantics that

$$\Sigma, B_1 \oplus B_2 \Downarrow^{pq} \Sigma'', b_1 \oplus b_2.$$

Most other cases follow using a similar argument.

The case for the two typing rules involving signals are a little different.

For a term of the form $\Gamma \vdash !s\colon \mathrm{bool}$, assume that $[\![\Sigma, !s]\!]$ makes Player behave in the same way as $[\![\Sigma', b]\!]$ for a Boolean value b. By definition of the dereferencing strategy, the initial question in $[\![\Sigma, !s]\!]$ is answered with the boolean $\Sigma(s)$. This entails that $b = \Sigma(s)$ and thus that $\Sigma, !s \Downarrow^p \Sigma, b$.

Suppose that the proposition holds for

$$\Gamma \vdash s\colon \mathrm{signal}^i_I,\ \Sigma \vdash W\colon \mathrm{angle}\ \text{and}\ \Gamma \vdash Q\colon \mathrm{qbit}^{I \cup \{i\}}.$$

We need to show that the proposition also holds for $\mathsf{meas}^i_I\ s\ W\ Q\colon \mathrm{qbit}^I$. Suppose that

$$\left[\!\left[\Sigma, \mathsf{meas}^i_I\ s\ \alpha\ Q\colon \mathrm{qbit}^I\right]\!\right]$$

makes Player play the same way as $[\![\Sigma'', |\varphi\rangle]\!]$ with probability p. By definition of the strategy meas^i_I, an initial question $\mathcal{A}_?$ about the output qbits is answered using first an interaction played using $[\![\Sigma', W]\!]$ to determine the angle W and second an interaction using $[\![\Sigma'', Q]\!]$ to determine the quantum state being measured. The first interaction will end with an answer which provides the measurement angle α with probability p; there may be some part of the interaction which uses $[\![\Sigma]\!]$ and leaves the game in a state where the next moves in the signal arenas are chosen by Player according to the strategy $[\![\Sigma']\!]$. Since this means that $[\![\Sigma, W]\!]$ makes Player behave as if he is using the strategy $[\![\Sigma', \alpha]\!]$. By induction hypothesis, this implies that $\Sigma, W \Downarrow^p \Sigma', \alpha$. The initial question $\mathcal{A}_?$ is transformed by the sig^i_I strategy into the question

$$\mathcal{A}_? \otimes \left\{[+_\alpha]^i_0, [-_\alpha]^i_1\right\}$$

in the arena $\mathbf{qbit}^{I \cup \{i\}}$. This question begins the second interaction and is answered using the strategy $[\![Q]\!]$. The answer to the initial question of this second interaction is a pair of measurement results m, b with probability q, where $b = \Sigma''(s)$ is the measurement results. The answer m is given by the state $|\varphi\rangle$ since meas^i_I makes Player answer the initial question $\mathcal{A}_m$ with this m and that by hypothesis the strategy

$$\left[\!\left[\Sigma, \mathsf{meas}^i_I\ s\ \alpha\ Q\colon \mathrm{qbit}^I\right]\!\right]$$

makes Player play the same way as $[\![\Sigma'', |\varphi\rangle]\!]$. It is not possible to infer from the measurement result the state being measured, but we can assume that this state is of the form $|\varphi\rangle|+_\alpha\rangle$ if b is 1 or $|\varphi\rangle|-_\alpha\rangle$ if $b = 0$ since all future interactions involving the measured qbit use this updated state. The measurement result b must be $\Sigma''(s)$. This implies that with probability q the strategy $[\![\Sigma', Q]\!]$ makes Player behave as if he is using the strategy $[\![\Sigma'', |\varphi\rangle|\pm_\alpha\rangle]\!]$. This implies by induction hypothesis that

$$\Sigma', Q \Downarrow^q \Sigma'', |\varphi\rangle|\pm_\alpha\rangle.$$

Using the reduction rule for measurement terms, we conclude that

$$\Sigma, \mathsf{meas}^i_I\ s\ W\ Q\colon \mathrm{qbit}^I \Downarrow^p qr\Sigma'', |\Phi\rangle$$

as required. □

A MCdata **context** $C[-]$ of type A with hole of type T is a constructed like a term with a free variable $-$ of type T.

Definition 4.8. *Let* $\Gamma \vdash M, M' : T$ *be two semi-closed terms.* M *and* M' *are* ***contextually equivalent*** $M \sim M'$ *if for all context* $C[-]$ *and* Γ*-store* Σ*,*

$$\Sigma, C[M] \Downarrow^p \Sigma', V \iff \Sigma, C[M'], \Downarrow^p \Sigma', V.$$

We can extend this definition to general open terms M, M' with a free qbit variable $x\colon \text{qbit}^{\text{In}}$ by asking that $M[Q/x] \sim M'[Q/x]$ for any qbit term $Q\colon \text{qbit}^{\text{In}}$.

We want to show that the denotational semantics defined in the last section captures contextual equivalence.

Proposition 4.9. *(Soundness for MCdata) If* $[\![M]\!] = [\![M']\!]$*, then* $M \sim M'$.

To prove this, we need the following lemma:

Lemma 4.10. *(Substitution for MCdata) Let* $\Gamma, x\colon T \vdash M\colon U$ *and* $\Gamma \vdash N\colon T$ *be two MCdata terms. Then*

$$\Gamma \vdash M[N/x]\colon T \textit{ and } [\![M[N/x]]\!] = \langle \text{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle ; [\![M]\!] .$$

Proof. By induction on the derivation of $\Gamma, x\colon T \vdash M\colon U$.

For the base cases where M is a constant term, so in

$$[\![M]\!] : \ [\![\Gamma]\!] \odot [\![T]\!] \to [\![U]\!]$$

all plays contain only moves from $[\![U]\!]$. The composition of $[\![M]\!]$ with $\langle \text{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle$ is thus $[\![M]\!]$. If M is a variable $y \neq x$, then $[\![y]\!]$ is the copy strategy between $[\![U]\!]$ and the $[\![U]\!]$ component of $[\![\Gamma]\!]$. The composed strategy $\langle \text{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle$ does not involve $[\![N]\!]$ and is thus equal to $[\![y]\!]$. If $M = x$, then this time $[\![x]\!]$ is the copy strategy between $[\![U]\!]$ and $[\![T]\!]$. Composing with $\langle \text{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle$ gives $[\![N]\!]$. In both case we get the desired result since $y[N/x] = y$ and $x[N/x] = N$.

We show how to deal with the induction step in the case of the measurement rule; the other cases are similar. Assume that the substitution lemma holds for $\Gamma, x\colon T \vdash s\colon \text{signal}^i_I$, $\Gamma, x\colon T \vdash W\colon \text{angle}$ and $\Gamma, x\colon T \vdash Q\colon \text{qbit}^{I\cup\{i\}}$. We want to show that it also holds for

$$\Gamma, x\colon T \vdash \mathsf{meas}^i_I\, s\, W\, Q\colon \text{qbit}^I.$$

On one hand, we have that

$$\left(\mathsf{meas}^i_I\, sWQ\colon \text{qbit}^I\right)[N/x] = \mathsf{meas}^i_I\, s[N/x]\, W[N/x]\, Q[N/x].$$

On the other hand, we have that by hypothesis

$$\begin{aligned} &\text{id}_{[\![\Gamma]\!]} \odot [\![N]\!] \,;\langle [\![s]\!] \,, [\![W]\!] \,, [\![Q]\!]\rangle \\ &\qquad = \langle \text{id}_{[\![\Gamma]\!]} \odot [\![N]\!] \,; [\![s]\!] \,, \text{id}_{[\![\Gamma]\!]} \odot [\![N]\!] \,; [\![W]\!] \,, \text{id}_{[\![\Gamma]\!]} \odot [\![N]\!] \,; [\![Q]\!]\rangle \\ &\qquad = \langle [\![s[N/x]]\!] \,, [\![W[N/x]]\!] \,, [\![Q[N/x]]\!]\rangle, \end{aligned}$$

and thus we get the desired result by composing with $\Lambda^{-1}\left(\mathsf{meas}^i_I\right)$. □

Proof of proposition 4.9. Suppose that we have $[\![M]\!] = [\![M']\!]$ for two semi-closed terms M, M' of type T. Take any context $C[-]$ with a hole of type T and any Γ-store Σ, and suppose that

$$\Sigma, C[M] \Downarrow^p \Sigma', V.$$

It follows by proposition 4.6 that for all well-opened $sab \in \mathcal{T}([\![\Sigma', V]\!])$ we have that

$$[\![\Sigma, C[M]]\!]\,(b \mid sa) = p\,[\![\Sigma', V]\!]\,(b \mid sa).$$

By hypothesis,

$$\begin{aligned}\llbracket \Sigma, C[M] \rrbracket &= \llbracket \Sigma \rrbracket \,; \langle \mathrm{id}_{\llbracket \Gamma \rrbracket}, \llbracket M \rrbracket \rangle ; \llbracket C[x] \rrbracket \\ &= \llbracket \Sigma \rrbracket \,; \langle \mathrm{id}_{\llbracket \Gamma \rrbracket}, \llbracket M' \rrbracket \rangle ; \llbracket C[x] \rrbracket \\ &= \llbracket \Sigma, C[M'] \rrbracket \,,\end{aligned}$$

and thus for all well-opened

$$sab \in \mathcal{T}(\llbracket \Sigma', V \rrbracket)$$

we have that

$$\llbracket \Sigma, C[M'] \rrbracket \, (b \mid sa) = p \, \llbracket \Sigma', V \rrbracket \, (b \mid sa) \,.$$

By adequacy we conclude that $C[M'], \Sigma \Downarrow^p \Sigma', V$.

By the symmetric argument, we get that $M \sim M'$. □

The main reason to introduce game semantics in classical programming language semantics is to be able to prove *full abstraction*. This property is the converse of soundness: it say that if two terms are contextually equivalent, then their denotations are the same.

Full abstraction is usually showed by proving the contrapositive proposition: if two terms M_1 and M_2 have different denotations $\llbracket M_1 \rrbracket$ and $\llbracket M_2 \rrbracket$, there must be a context $C[-]$ which can distinguish them. Proving this requires the construction of a context $C[-]$ associated to a given strategy. In game semantics, strategies are identified using an equivalence relation defined as follows. Let the test arena **test** be the arena with only one question q and one answer a. A **test** for an arena A is a strategy $\alpha \colon A \multimap$ **test**. A strategy σ in A passes the test α if $\sigma; \alpha = \top$, where $\top$ is the strategy where the question q is answered with the answer a. Two strateges σ and τ are equivalent if they both pass the same tests. It is shown in game semantics that when working with strategies up to this equivalence relation, every important property (such as proposition 4.6 and adequacy) stay valid. To prove full abstraction, one must produce a context $C[-]$ that distinguishes M_1 and M_2 when $\llbracket M_1 \rrbracket \neq \llbracket M_2 \rrbracket$. Using the equivalence relation, this last inequality means that there is a test $\alpha \colon A \multimap$ **test** which is passed by one of the two strategies but not by the other. The required context can be constructed from this test if the strategies of the category where the denotation is defined are characterised very tightly so that this construction is possible. We were not able to get such a result in the case of MCdata and for the other languages presented in this thesis because we do not have an appropriate characterisation of quantum strategies.

Chapter 5

λ-calculus with quantum stores

In the last chapter we used quantum arenas to define a denotational semantics for a typed variant of the measurement calculus. Based on attempts to construct a quantum arena based denotational semantics for Selinger and Valiron's language, which was presented in section 2.2.2, we developed two new quantum λ-calculi using different approaches to incorporate quantum states in classical languages. The first one uses quantum stores and is the topic of this chapter. In the second one, quantum states are used directly as data in the language; the description of this second language is the topic of the next chapter. We begin this chapter with a review of the main observations that lead us to introduce two new quantum λ-calculi. Then we present the first language, based on quantum stores, and its semantics.

5.1 Critique of the quantum λ-calculus

In the first presentations of the quantum λ-calculus developed by Selinger and Valiron [Val04, SV06a] no denotational semantics was given. They proposed in [SV06b] a denotational semantics for the linear part of the quantum λ-calculus; their interpretation is in the category **CPM** of completely positive maps on finite dimensional Hilbert spaces. The category **CPM** inherits a compact closed structure from the category of finite dimensional Hilbert spaces. By working in this category the difficulties of using trace non-increasing maps described in section 3.1.1 are avoided, but at the cost of having programs interpreted as trace-increasing completely positive maps because the interpretation of λ-abstraction can produce such maps (for example the term $\lambda x, y.\, x \otimes y$). This is incompatible with the expectation that terms of a language that described manipulation of quantum data should be interpreted as superoperators, which correspond to physically realisable operations.

We explored the possibility of using quantum arenas and strategies to construct a denotational semantics for the full language. The main difficulty encountered is with the tensor type operation of Selinger and Valiron's quantum λ-calculus: it can be used on both quantum and classical types. So if we want to inductively associate an arena $[\![A]\!]$ to each quantum λ-calculus type A, we need to define $[\![A \otimes B]\!]$ using the classical product of arenas $[\![A]\!] \odot [\![B]\!]$ in general, but by $[\![A]\!] \otimes [\![B]\!]$ when both A and B are qbit types. While with this idea we are able to deal with types, it creates difficulties for the definition of the denotation of terms. In particular, we need a strategy

$$[\![x\colon \text{qbit}, y\colon \text{qbit} \vdash x \otimes y\colon \text{qbit} \otimes \text{qbit}]\!]$$

which should intuitively take two qbit states and tensor them. This should be a strategy in the arena

$$[\![\text{qbit}]\!] \odot [\![\text{qbit}]\!] \multimap [\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!]\,,$$

but there is no natural strategy of this type with the required behavior. Such a strategy needs to specify how to answer a question in $[\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!]$ by measuring each qbit component separately. As we explained in section 3.4.2, this is not possible in general.

If instead of interpreting the type hypothesis $x\colon \text{qbit}, y\colon \text{qbit}$ in the above typing judgement as $[\![\text{qbit}]\!]\otimes[\![\text{qbit}]\!]$ instead of $[\![\text{qbit}]\!]\odot[\![\text{qbit}]\!]$, we run into a different difficulty. This time we have problems with abstraction. Suppose we want to define a strategy

$$[\![y\colon \text{qbit} \vdash \lambda x.\, x \otimes y\colon \text{qbit} \multimap (\text{qbit} \otimes \text{qbit})]\!]\,.$$

The typing judgement must be introduced using the abstraction rule

$$\frac{x\colon \text{qbit}, y\colon \text{qbit} \vdash x \otimes y\colon \text{qbit} \otimes \text{qbit}}{y\colon \text{qbit} \vdash \lambda x.\, x \otimes y\colon \text{qbit} \multimap (\text{qbit} \otimes \text{qbit})}$$

We thus need an adjunction between strategies in

$$[\![\text{qbit}]\!] \multimap ([\![\text{qbit}]\!] \multimap [\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!])$$

with those in the arena

$$[\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!] \multimap [\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!]\,.$$

This again requires that one constructs a strategy which tells how to answer measurement questions in $[\![\text{qbit}]\!] \otimes [\![\text{qbit}]\!]$ using separate measurements in the two qbit components.

There is also another issue with the quantum λ-calculus. The language does not allow quantum states to be introduced directly (as, for example, we allow in the language MCdata in the previous chapter): quantum states can only be *referred to* by using variables of type qbit. In the type system, quantum states are considered as data of type qbit which can't be duplicated, but at the same time the language only allows one to have *references* to qbits, which, intuitively, can be duplicated. There are two ways to introduce quantum data into programs: a direct way using a syntax to denote quantum states and an indirect way using references to external quantum stores. In the first case quantum data should be treated linearly to make it impossible to duplicate an unknown quantum state, but in the second case references to qbits should not have this restriction. We thus introduce two different languages, one for each of these approaches. We describe the second one in this chapter and the first one in the next chapter.

5.2 Simply typed λ-calculus with quantum stores

The λ-calculus with quantum stores language (**QSL**) introduced in this chapter avoids the difficulties described in the last section. The syntax of the language is built upon a simply typed λ-calculus with pairing and conditionals; quantum operations are added using *quantum stores* which have a syntax analogous to classical stores. In a classical higher-order programming language with stores, like idealised ALGOL [Rey81], stores are references to values. They are used through various operations like dereferencing and assignment. The quantum stores we use below are defined according to the following parallel between classical and quantum references:

Classical stores	**Quantum stores**
Dereferencing	Measurement
Assignment	Preparation
Command with side effects	Unitary transformation
Juxtaposition by products	Juxtaposition by tensor products

In this perspective, a quantum state is viewed as existing in an external store which can only be accessed indirectly. In this picture, the quantum counterpart of dereferencing, which classically returns the value stored, is quantum measurement. The counterpart of assignment is state preparation. Note that, while classically it is possible to assign a value to a store multiple times, this is not the case with quantum stores, as a quantum state cannot be destroyed. Instead, preparation creates a new quantum state. Classical stores can be equipped with commands with side effects, for example, an integer incrementation command. This role is played by unitary operations in the quantum counterpart. Finally, when many classical stores are used in some programs, they are simply juxtaposed using products. In the quantum case, juxtaposed quantum stores must be described by tensor products to allow them to hold entangled states.

5.2.1 Syntax

The syntax of QSL is that of a classical simply typed λ-calculus with pairing, conditionals and sequential composition, augmented with new constructs that permit manipulation of quantum stores. To accommodate these, we need to introduce a new syntactic device. When multiple quantum stores are combined, they can be measured by using a projective measurement on the whole space. Because of this, we must be able to refer to the combined store as a whole, while keeping the possibility to refer to a part of the system. To this end, we introduce tensor of variables in the syntax. An **extended variable** is an expression of the form $x_1 \otimes \cdots \otimes x_n$, where the x_i are variables such that $x_i \neq x_j$ if $i \neq j$. Two extended variables $x_1 \otimes \cdots \otimes x_n$ and $y_1 \otimes \cdots \otimes y_m$ are **disjoint** if $x_i \neq y_j$ for all i, j. Two such extended variables can be joined to form a new extended variable

$$x_1 \otimes \cdots \otimes x_n \otimes y_1 \otimes \cdots \otimes y_m.$$

Note that when we use $x_1 \otimes \cdots \otimes x_n$ to refer to an arbitrary extended variable, the case $n = 1$ is also possible. We use the notation

$$x_1 \otimes \cdots \otimes x_n \sqsubseteq y_1 \otimes \cdots \otimes y_m$$

when each of the variables $x_1, \ldots, x_n$ occurs in $y_1 \otimes \cdots \otimes y_m$ and the order of the occurrences is the same in both extended variables. We say in this case that $x_1 \otimes \cdots \otimes x_n$ is a **subvariable** of $y_1 \otimes \cdots \otimes y_m$. To simplify the notation, we use $\overline{x}$ instead of $x_1 \otimes \cdots \otimes x_n$, leaving the number n implicit.

The terms of QSL are defined by

$$\begin{aligned} M, N, P ::= {} & \overline{x} \mid 0 \mid 1 \mid \mathsf{skip} \mid \lambda\overline{x}.\, M \mid MN \mid \mathsf{if}\, M\, \mathsf{then}\, N\, \mathsf{else}\, P \mid \langle M, N\rangle \mid \\ & \mathsf{fst}\, M \mid \mathsf{snd}\, M \mid M; N \mid \mathsf{meas}^{x}\overline{x} \mid \mathsf{new}\, \overline{x}\, \mathsf{in}\, M \mid U^{\overline{z}}\, \overline{x} \mid \mathsf{prep}\, \overline{y}\, \mathsf{with}\, \overline{x}\, \mathsf{in}\, M \end{aligned}$$

where $\overline{x}$ and $\overline{y}$ can be any extended variables, $x, \overline{z} \sqsubseteq x$ and U can be any multiple-qbits unitary transformation. All the classical operations used are standard operations: $\langle M, N\rangle$ is pairing, fst and snd are the two associated projection operations, $M; N$ is sequential composition, and skip is the operation doing nothing. The quantum part of the language consists of operations to manipulate quantum stores: measurement, qbit creation, unitary modification and preparation of extra qbits. The unitary operation syntax $U^{\overline{z}}\,\overline{x}$ means that the unitary transformation U of rank n is applied to the qbits $\overline{z} = z_1 \otimes \cdots \otimes z_n$ of the quantum store $\overline{x}$. While $\overline{x}$ is an extended variable term, the extended variable $\overline{z}$ is used as a label and is not considered a free variable of $U^{\overline{z}}\,\overline{x}$. We will also use the notation $U\,\overline{z}$ to denote this operation when the quantum store $\overline{x}$ is being implicitly specified in the context. The measurement operation $\mathsf{meas}^{x}\,\overline{x}$ measures the qbit x in the quantum store $\overline{x}$ in the canonical basis and returns a boolean value corresponding to the measurement result. As for unitary operations, the variable x is only a label to point out which qbit of $\overline{x}$ is measured. We will also use the shorter notation $\mathsf{meas}\, x$ to denote $\mathsf{meas}^{x}\overline{x}$ when it is clear in the context which variable $\overline{x}$ is used. For the preparation operation, $\mathsf{prep}\,\overline{y}\,\mathsf{with}\,\overline{x}\,\mathsf{in}\, M$ means that a given quantum store $\overline{x}$ is extended to a larger store by adding extra qbits prepared in the $|0\rangle$ state. In M, the whole extended store is referred to as $\overline{x} \otimes \overline{y}$.

As in any λ-calculus, the λ operation is a binder. Observe that it can be used on extended variables, i.e. terms like $\lambda x \otimes y.\, \mathsf{meas}\, x$ are allowed. The preparation operation is also a binder: $\overline{x}$ is not free in the term $\mathsf{prep}\,\overline{y}\,\mathsf{with}\,\overline{x}\,\mathsf{in}\, M$. The set of free extended variables of M is denoted by $\mathrm{FV}(M)$. A term M is **closed** if it has no free extended variables. We use the notation $M[N/\overline{x}]$ to denote the capture-free substitution (no occurrence of a free variable in N is bound in M) of the term N for every occurrence of $\overline{x}$. Note that the syntax limits substitution in unitary and measurement operations to changes of variables. For example the substitution

$$U^{\overline{y}}\,\overline{x}[N/\overline{x}] = U^{\overline{y'}}\,\overline{x'}$$

is defined only when $N = \overline{x'}$ and $\overline{y'} \sqsubseteq \overline{x'}$ is the subvariable corresponding to $\overline{y} \sqsubseteq x$.

For clarity, we use the alternative notation $\mathsf{let}\, x = N\, \mathsf{in}\, M$ for $(\lambda x.\, M)N$. When multiple variables are bound in this manner successively, we use the notation

$$\mathsf{let}\, x_1 = N_1, \ldots, x_n = N_n\, \mathsf{in}\, M$$

Table 5.1 QSL typing rules

$$\frac{}{\Gamma, \overline{x}\colon A \vdash \overline{x}\colon A} \qquad \frac{}{\Gamma \vdash 0\colon \text{bool}} \qquad \frac{}{\Gamma \vdash 1\colon \text{bool}} \qquad \frac{}{\Gamma \vdash \mathsf{skip}\colon \text{com}}$$

$$\frac{\Gamma \vdash M\colon A \Rightarrow B \qquad \Gamma \vdash N\colon A}{\Gamma \vdash MN\colon B} \qquad \frac{\Gamma, \overline{x}\colon A \vdash M\colon B}{\Gamma \vdash \lambda\overline{x}.\, M\colon A \Rightarrow B} \qquad \frac{\Gamma \vdash M_1\colon A_1 \qquad \Gamma \vdash M_2\colon A_2}{\Gamma \vdash \langle M_1, M_2\rangle\colon A_1 \times A_2}$$

$$\frac{\Gamma \vdash M\colon A \times B}{\Gamma \vdash \mathsf{fst}\, M\colon A} \qquad \frac{\Gamma \vdash M\colon A \times B}{\Gamma \vdash \mathsf{snd}\, M\colon B} \qquad \frac{\Gamma \vdash P\colon \text{bool} \qquad \Gamma \vdash M\colon A \qquad \Gamma \vdash N\colon A}{\Gamma \vdash \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N\colon A}$$

$$\frac{\Gamma \vdash M\colon \text{com} \qquad \Gamma \vdash N\colon A}{\Gamma \vdash M; N\colon A}\ A = \text{com or bool} \qquad \frac{}{\Gamma, \overline{x}\colon \text{qstore} \vdash \mathsf{meas}^{x}\, \overline{x}\colon \text{bool}}$$

$$\frac{}{\Gamma, \overline{x}\colon \text{qstore} \vdash U^{\overline{y}}\, \overline{x}\colon \text{com}}$$

$$\frac{\Gamma, \overline{x}\colon \text{qstore} \vdash M\colon A}{\Gamma \vdash \mathsf{new}\, \overline{x}\, \mathsf{in}\, M\colon A} \qquad \frac{\Gamma, \overline{x} \otimes \overline{y}\colon \text{qstore} \vdash M\colon A}{\Gamma, \overline{x}\colon \text{qstore} \vdash \mathsf{prep}\, \overline{y}\, \mathsf{with}\, \overline{x}\, \mathsf{in}\, M\colon A}$$

for $(\lambda x_n. \ldots (\lambda x_1. M) N_1 \ldots) N_n$. Note that the terms $\lambda x_1(\lambda x_2 \ldots (\lambda x_n. M) \ldots)$ and $\lambda\overline{x}. M$ are different: in the first one the variables $x_1, \ldots x_n$ are considered separately while in the second case $\overline{x} = x_1 \otimes \cdots \otimes x_n$ is considered as a single variable.

5.2.2 Types

The types of QSL are the following:

$$A, B ::= \text{ bool} \mid \text{com} \mid A \times B \mid A \Rightarrow B \mid \text{qstore}.$$

The type bool is the type of boolean constants, $A \times B$ and $A \Rightarrow B$ are respectively the types of pairs and functions. The type com is the type of *commands* which can be composed using sequential composition. The type qstore is the type of a quantum store. A quantum store does not have a fixed dimension, as the number of qbits it holds can vary in the course of a computation if preparation operations are used.

The typing rules for the classical part are given in table 5.1. The rules for the classical part of the language are the standard rules of a simply typed λ-calculus where extended variables can be used. The rules for involving quantum operations encode the idea that the content of quantum stores can be measured, modified using unitary transformations and that quantum stores can be prepared or extended with an ancilla state. Note that the unitary operation rule allows unitary operations to be applied only to part of a quantum register. An important feature of QSL is that the typing rules do not forbid having multiple references to a quantum store. For example, the typing judgement $x\colon \text{qstore} \vdash \langle \text{meas}\, x, \text{meas}\, x\rangle\colon \text{bool} \times \text{bool}$ is valid. Copying a reference to a qbit is not the same thing as duplicating the qbit. Yet the language does not allow unknown qbit duplication: to duplicate the content of a quantum store x, one would need to prepare a new qbit y and apply an appropriate unitary transformation to the quantum store $x \otimes y$. There is no such unitary transformation.

5.2.3 Operational semantics

The operational semantics of the classical part of the quantum store language is standard. For the quantum part we use a quantum variant of stores. Note that we expect that the reduction relation of this language depends on reduction order, since, as we pointed out in section 2.2.2, it is the case in the presence of operations with side-effects like quantum measurements.

A **quantum store** Q is a function taking extended variables $x_1 \otimes \cdots \otimes x_n$ taken in a finite domain of extended variables $|Q|$ to a state $|x_1 \ldots x_n\rangle_Q \in \left(\mathbb{C}^2\right)^{\otimes n}$. The domain $|Q|$ is assumed to contain only disjoint extended variables. A quantum store holds the state of the quantum registers that are used in a quantum λ-calculus term. We drop the index Q when the context makes it clear to which quantum store a state belongs.

A quantum store Q can be modified in various ways. First, it can be extended by the addition of a new quantum register; since this is similar to the extension of a classical store we use the notation

$$Q[|x_1 \ldots x_n\rangle \mapsto |\varphi\rangle]$$

to denote the extension of Q to a store with domain $|Q| \cup \{x_1 \otimes \cdots \otimes x_n\}$ and associating to the new extended variable $x_1 \ldots x_n$ the state $|\varphi\rangle$.

Another important operation is *preparation* of extra qbits appended to a cell of a given quantum store Q. If $x_1 \otimes \cdots \otimes x_n \in |Q|$, then

$$Q[|x_1 \ldots x_n y_1 \ldots y_m\rangle \mapsto |x_1 \ldots x_n\rangle|0 \ldots 0\rangle]$$

is the quantum store with $x_1 \otimes \cdots \otimes x_n$ removed from $|Q|$ and

$$x_1 \otimes \cdots \otimes x_n \otimes y_1 \otimes \cdots \otimes y_m$$

added, and with associated state

$$|x_1 \ldots x_n y_1 \ldots y_n\rangle = |x_1 \ldots x_n\rangle|0 \ldots 0\rangle.$$

Note that by definition of quantum store, $\{x_1, \ldots x_n\}$ and $\{y_1, \ldots y_m\}$ are disjoint.

The final operation that we need is the modification of one register using a unitary operation or a projection. Given a quantum store Q and a linear map A over the Hilbert space associated to the extended variable $x_1 \otimes \cdots \otimes x_n \in |Q|$, we denote by

$$Q[|x_1 \ldots x_n\rangle \mapsto A|x_1 \ldots x_n\rangle]$$

the quantum store where $|x_1 \ldots x_n\rangle$ is replaced by $A|x_1 \ldots x_n\rangle$.

A QSL **program** is a pair $Q, \Gamma \vdash M \colon A$ where Q is a quantum store, $\Gamma \vdash M \colon A$ is a valid typing judgement such that all the qstore variables in Γ are in $|Q|$. We say that a program Q, M is **closed** if $|\Gamma| \subseteq |Q|$. To simplify the notation, we will often leave the types implicit and write Q, M instead of $Q, \Gamma \vdash M \colon A$.

A **value** for QSL is a term of the recursively defined form

$$V ::= \; x_1 \otimes \cdots \otimes x_n \mid 0 \mid 1 \mid * \mid \mathsf{skip} \mid \lambda\bar{y}.\, M \mid \langle M, N\rangle,$$

where $\bar{x}$ can be any extended variable and M is any term with $\bar{y} \in \mathrm{FV}(M)$.

Table 5.2 QSL probabilistic reduction rules

$$\frac{}{Q, V \Downarrow Q, V} \qquad \frac{Q, M \Downarrow^p Q', \lambda x.\, M' \qquad Q', M'[N/x] \Downarrow^q Q'', V}{Q, MN \Downarrow^{pq} Q'', V}$$

$$\frac{Q, M \Downarrow^p Q', V}{Q, \mathsf{fst}\,\langle M, N\rangle \Downarrow^p Q', V} \qquad \frac{Q, N \Downarrow^p Q', V}{Q, \mathsf{snd}\,\langle M, N\rangle \Downarrow^p Q', V}$$

$$\frac{Q, M \Downarrow^p Q', \mathsf{skip} \qquad Q', N \Downarrow^q Q'', V}{Q, M; N \Downarrow^{pq} Q'', V}$$

$$\frac{Q, P \Downarrow^p Q', 0 \qquad Q', N \Downarrow^q Q'', V}{Q, \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \Downarrow^{pq} Q'', V} \qquad \frac{Q, P \Downarrow^p Q', 1 \qquad Q', M \Downarrow^q Q'', V}{Q, \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \Downarrow^{pq} Q'', V}$$

$$\frac{}{Q, U^{y_1 \otimes \ldots \otimes y_k}\, x_1 \otimes \cdots \otimes x_n \Downarrow Q[|x_1 \ldots x_n\rangle \mapsto U^{y_1 \otimes \cdots \otimes y_k}|x_1 \ldots x_n\rangle], \mathsf{skip}}$$

$$\frac{}{Q, \mathsf{meas}\, x_j \Downarrow^{\|[0]^{x_j}|x_1 \ldots x_n\rangle\|} Q[|x_1 \ldots x_n\rangle \mapsto [0]^{x_j}|x_1 \ldots x_n\rangle / \|[0]^{x_j}|x_1 \ldots x_n\rangle\|], 0}$$

$$\frac{}{Q, \mathsf{meas}\, x_j \Downarrow^{\|[1]^{x_j}|x_1 \ldots x_n\rangle\|} Q[|x_1 \ldots x_n\rangle \mapsto [1]^{x_j}|x_1 \ldots x_n\rangle / \|[1]^{x_j}|x_1 \ldots x_n\rangle\|], 1}$$

$$\frac{Q[|x_1 \ldots x_n\rangle \mapsto |0 \ldots 0\rangle], M \Downarrow^p Q', V}{Q, \mathsf{new}\, x_1 \otimes \cdots \otimes x_n\, \mathsf{in}\, M \Downarrow^p Q', V} \; x_1 \otimes \cdots \otimes x_n \notin |Q|$$

$$\frac{Q[|x_1 \ldots x_n y_1 \ldots y_m\rangle \mapsto |\varphi\rangle|0\rangle], M \Downarrow^p Q', V}{Q[|x_1 \ldots x_n\rangle \mapsto |\varphi\rangle], \mathsf{prep}\, \overline{y}\, \mathsf{with}\, \overline{x}\, \mathsf{in}\, M \Downarrow^p Q', V}$$

We define the operational semantics of QSL as a big-step probabilistic reduction relation between programs. The notation

$$Q, M \Downarrow^p Q', V$$

means that when M is run with a quantum store in state Q, it reduces with probability p to the value V with the quantum store left in state Q'. When $p = 1$, we omit the probability argument and write simply $Q, M \Downarrow Q', V$. This relation is defined inductively by the rules in table 5.2. Most of these rules are the usual reduction rules for the simply typed λ-calculus with sequential composition, conditionals and pairing. The reduction rules for the classical part of the language do not affect the quantum stores. The rules involving measurements, preparations or unitary transformations change the quantum stores according to quantum mechanics. For example, the rule for measurement says that if x_i is measured with a quantum store in state Q, then the state $|x_1 \ldots x_n\rangle_Q$ where x occurs is projected with the projection $[0]^{x_i}$ or $[1]^{x_i}$, depending on the measurement result, and normalised. Note that this is the only place where there is a probabilistic branching in the reduction. For a unitary transformation operation U, the part of the quantum store Q affected by U is updated to $U|x_1 \ldots x_n\rangle$ and the term reduces to the command skip.

Figure 5.1 QSL teleportation

```
teleport$_{xz}$=
  prep $y \otimes z$ with $x$ in
    $H\,y$; $\wedge X y \otimes z$;
    $H\,x$; $\wedge X\,x \otimes y$;
    let $b_x$ = meas $x$, $b_y$ = meas $y$ in
      if $b_x$ then
          if $b_y$ then $U_{11}\,z$ else $U_{10}\,z$
        else
          if $b_y$ then $U_{01}\,z$ else $U_{00}\,z$
```

Example 5.1. Consider the following two terms M_1 and M_2 defined respectively by

$$M_1 : \ \wedge U x \otimes y \qquad M_2 : \text{ if meas } x \text{ then } (U\,y) \text{ else skip}$$

where $\wedge U$ denote the controlled version of a unitary operation U. This is defined by

$$\wedge U|b_1\rangle|b_2\rangle = |b_1\rangle|b_1 \oplus b_2\rangle,$$

where $\oplus$ is the exclusive-or operation. We have that

$$x \otimes y : \text{qstore} \vdash M_1 : \text{com and } x \otimes y : \text{qstore} \vdash M_2 : \text{com}.$$

In a quantum store state Q which assign $|\varphi\rangle$ to $x \otimes y$, M_1 reduce to skip and the state Q is modified by the unitary operation:

$$Q, M_1 \Downarrow Q\,[|xy\rangle \mapsto \wedge U|xy\rangle]\,, \text{skip}.$$

The term M_2 also reduces to skip but leaves the quantum store in a different state:

$$Q\,[|xy\rangle \mapsto |\varphi\rangle]\,, M_2 \Downarrow^{p} Q\,[|xy\rangle \mapsto [0]^x|xy\rangle]\,, \text{skip}$$
$$Q\,[|xy\rangle \mapsto |\varphi\rangle]\,, M_2 \Downarrow^{1-p} Q\,[|xy\rangle \mapsto U^y[1]^x|xy\rangle]\,, \text{skip}$$

where $p = \mathrm{tr}\,([0]^x|\varphi\rangle\langle\varphi|)$.

Example 5.2. It is possible to program the quantum teleportation protocol [BBC+93] in the quantum store language. It is represented as a term teleport$_{xz}$, defined in figure 5.1, which transfers an unknown state from some quantum store x to another quantum store z. In the definition of teleport$_{xz}$ the operation H is the Hadamard transformation and

$$U_{00} = I,\ U_{01} = X,\ U_{10} = Z,\ \text{and } U_{11} = ZX$$

are the four possible correction operations, one of which must be applied to z to change its state to that of the input quantum store x. If follows from the typing rules that

$$x : \text{qstore} \vdash \text{teleport}_{xz} : \text{com}$$

The command $\mathsf{teleport}_{xz}$ performs the teleportation protocol to transfer the state of the qbit register x to the qbit register z. This can be verified using the operational semantics rules: it is possible to derive that

$$Q, \mathsf{teleport}_{xz} \Downarrow Q\left[|xyz\rangle \mapsto U^z_{b_x b_y}[b_x]^x[b_y]^y \mathrm{cnot}^{xy} H^x |xyz\rangle\right], \mathsf{skip},$$

where we label each unitary transformation and projectors by the subspace associated to the label variables.

Example 5.3. Any quantum circuit can be represented as a QSL term. Suppose that the circuit takes a state $|x_1 \dots x_n\rangle$ as input which is initially tensored with the state

$$|y_1 \dots y_m\rangle = |0 \dots 0\rangle.$$

The unitary operations $U_1, \dots, U_k$ are applied to this state, and at the end the qbits

$$x_1, \dots x_n, y_1, \dots y_m$$

are measured. This is represented as the term M defined as follows:

$$\begin{array}{l}
\mathsf{prep}\, y_1 \otimes \cdots \otimes y_m \,\mathsf{with}\, x_1 \otimes \dots x_n \,\mathsf{in} \\
\quad U_1\, x_1 \otimes \cdots \otimes x_n \otimes y_1 \otimes \cdots \otimes y_m; \\
\qquad\qquad \vdots \\
\quad U_k\, x_1 \otimes \cdots \otimes x_n \otimes y_1 \otimes \cdots \otimes y_m; \\
\quad \langle \mathsf{meas}\, x_1, \dots, \mathsf{meas}\, x_n, \mathsf{meas}\, y_1, \dots, \mathsf{meas}\, y_m \rangle
\end{array}$$

We can derive that

$$x_1 \otimes \cdots \otimes x_n \colon \mathrm{qstore} \vdash M \colon \mathrm{bool} \odot \cdots \odot \mathrm{bool}.$$

Let Q be a quantum store with $|x_1 \dots x_n\rangle \mapsto |\varphi\rangle$. We have that

$$Q, M \Downarrow^p Q', \langle b_1, \dots, b_n \rangle,$$

where $b_1, \dots b_{n+m}$ and are the results of the final measurements operations and

$$Q' = Q\left[|x_1 \dots x_n\rangle \mapsto [b_{n+m}]^{y_m} \dots [b_{n+1}]^{y_1} [b_n]^{x_n} \dots [b_1]^{x_1} U_k \dots U_1 |\varphi\rangle |0 \dots 0\rangle\right].$$

5.2.4 Denotational semantics

We now use quantum strategies to construct a denotational semantics for the quantum store language. We want to define an arena $[\![A]\!]$ corresponding to each type A and a strategy $[\![M]\!] : [\![\Gamma]\!] \to [\![A]\!]$ corresponding to each term $\Gamma \vdash M \colon A$. We will use quantum strategies defined with intervention operators, as described in section 3.6.3.

The **qstore** arena is the arena with quantum interventions $\mathcal{E}_? = \left\{\mathcal{E}^?_m\right\}$ as questions and natural numbers m as answers. The question $\mathcal{E}_?$ enables its possible measurements results.

A play in this arena is a sequence of moves

$$\mathcal{E}_{?[1]} m_1 \cdots \mathcal{E}_{?[n]} m_n$$

where the quantum interventions $\mathcal{E}_{?[k]}$ may all be different. We need a strategy $[\rho]$ in **qstore** which describes a quantum state ρ.

The probabilistic strategy $[\rho]$ in **qstore** associated to a density matrix ρ is defined by $[\rho](\epsilon) = 1$ and

$$[\rho](\mathcal{E}_{?[1]}m_1 \dots \mathcal{E}_{?[n]}m_n) = \operatorname{tr}\left(\mathcal{E}^{?[n]}_{m_n} \dots \mathcal{E}^{?[1]}_{m_1}(\rho)\right).$$

Note that since we use the convention that impossible composition of superoperators yields the zero operator, the above definition assigns probability zero to plays which involve domain inconsistencies. For example, if Opponent asks another question $\mathcal{E}_{?[2]}$ after receiving an answer to $\mathcal{E}_{?[1]}$, all possible Player answers will have probability zero when the domain of $\mathcal{E}_{?[2]}$ is different than $\mathbf{SD}(H_{m_1})$. When the domain and $\mathbf{SD}(H_m)$ match, the question $\mathcal{E}_{?[2]}$ is answered using the normalised state

$$\mathcal{E}^{?[1]}_{m_1}(\rho) / \operatorname{tr}\left(\mathcal{E}^{?[1]}_{m_1}(\rho)\right).$$

It is easy to verify this satisfies the definition of probabilistic strategies. Note that the strategy $[\rho]$ is thread dependent: the first question is answered using the probabilities given by $p_{m_1} = \operatorname{tr}\left(\mathcal{E}^{?[1]}_{m_1}(\rho)\right)$, but a second question in a new thread will be answered with the probability distribution given by $\operatorname{tr}\left(\mathcal{E}^{?[2]}_{m_2}\mathcal{E}^{?[1]}_{m_1}(\rho)\right)/p_{m_1}$, i.e. using the updated state $\mathcal{E}^{?[1]}_{m_1}(\rho)/p_{m_1}$. Thus in general the probability distribution used is different in different threads, and is updated according to the laws of quantum mechanics.

Example 5.4. We can define a strategy which describes a unitary operation. This is a strategy $[U]$ in the arena **qstore** ⊸ **com**. Suppose that the superoperator corresponding to U is $\mathcal{U}$. A typical play using $[U]$ is "run $\{\mathcal{U}_0\}_?$ 0done". The $\{\mathcal{U}_0\}_?$ question in the **qstore** arena changes the state used to answer future questions in the arena. Notice that Player does not learn anything about the state in this interaction with Opponent because there is only one possible measurement result. The strategy $[U]$ really describes the effect of U since one can verify that $[\rho]; [U] = [\mathcal{U}(\rho)]; \text{skip}$ using the definition of composition of strategies.

Example 5.5. We define a strategy which represents performing a projective measurement of the state of a quantum store as follows.

$$\begin{array}{ccc} \mathbf{qstore} & \overset{\text{meas}}{\multimap} & \mathbf{bool} \\ & & ? \\ C_? & & \\ m & & \\ & & m \end{array}$$

The measurement strategy makes Player answer the first question in the output Boolean component by asking about the result of a measurement in the computational basis of the input qbit with the quantum intervention $C = \{\mathcal{P}_0, \mathcal{P}_1\}$, where $\mathcal{P}_m$ is the projective measurement superoperator defined by $\mathcal{P}_m(\rho) = [m]\rho[m]$. Player then copies the answer m to the output component. In contrast to the case of unitary transformations,

Table 5.3 QSL denotational semantics

$$\llbracket \Gamma, \overline{x} \colon A \vdash \overline{x} \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \otimes \llbracket A \rrbracket \xrightarrow{\pi_A} \llbracket A \rrbracket \qquad \llbracket \Gamma \vdash \mathsf{skip} \colon \mathrm{com} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\mathrm{skip}} \mathbf{com}$$

$$\llbracket \Gamma \vdash 0 \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{0} \mathbf{bool} \qquad \llbracket \Gamma \vdash 1 \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{1} \mathbf{bool}$$

$$\llbracket \Gamma \vdash MN \colon B \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle} (\llbracket A \rrbracket \multimap \llbracket B \rrbracket) \odot \llbracket A \rrbracket \xrightarrow{\mathsf{eval}} \llbracket B \rrbracket$$

$$\llbracket \Gamma \vdash \lambda \overline{x}. M \colon A \Rightarrow B \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\Lambda(\llbracket M \rrbracket)} \llbracket A \multimap B \rrbracket$$

$$\llbracket \Gamma \vdash \langle M_1, M_2 \rangle \colon A_1 \times A_2 \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket M_1 \rrbracket, \llbracket M_2 \rrbracket \rangle} \llbracket A_1 \rrbracket \odot \llbracket A_2 \rrbracket$$

$$\llbracket \Gamma \vdash \mathsf{fst}\, M \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \odot \llbracket B \rrbracket \xrightarrow{\pi_{\llbracket A \rrbracket}} \llbracket A \rrbracket$$

$$\llbracket \Gamma \vdash \mathsf{snd}\, M \colon B \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \odot \llbracket B \rrbracket \xrightarrow{\pi_{\llbracket B \rrbracket}} \llbracket B \rrbracket$$

$$\llbracket \Gamma \vdash \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket P \rrbracket, \llbracket M \rrbracket, \llbracket N \rrbracket \rangle} \mathbf{bool} \odot \llbracket A \rrbracket \odot \llbracket A \rrbracket \xrightarrow{\mathsf{cond}} \llbracket A \rrbracket$$

$$\llbracket \Gamma \vdash M; N \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket M_1 \rrbracket, \llbracket M_2 \rrbracket \rangle} \mathbf{com} \odot \llbracket A \rrbracket \xrightarrow{\mathrm{seq}_A} \llbracket A \rrbracket \ , \ A = \mathrm{com\ or\ bool}$$

$$\llbracket \Gamma, \overline{x} \colon \mathrm{qstore} \vdash \mathsf{meas}^x\, \overline{x} \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \mathbf{qstore} \xrightarrow{\pi_{\mathbf{qstore}}} \mathbf{qstore} \xrightarrow{\mathsf{meas}} \mathbf{bool}$$

$$\left\llbracket \Gamma, \overline{x} \colon \mathrm{qstore} \vdash U^{\overline{y}}\, \overline{x} \colon \mathrm{com} \right\rrbracket : \ \llbracket \Gamma \rrbracket \odot \mathbf{qstore} \xrightarrow{\pi_{\mathbf{qstore}}} \mathbf{qstore} \xrightarrow{[U]} \mathbf{com}$$

$$\llbracket \Gamma \vdash \mathsf{new}\, \overline{x}\, \mathsf{in}\, M \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket \mathrm{id}_{\llbracket \Gamma \rrbracket} \rrbracket, [|0 \ldots 0\rangle\langle 0 \ldots 0|] \rangle} \llbracket \Gamma \rrbracket \odot \mathbf{qstore} \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket$$

$$\llbracket \Gamma, \overline{x} \colon \mathrm{qstore} \vdash \mathsf{prep}\, \overline{y}\, \mathsf{with}\, \overline{x}\, \mathsf{in}\, M \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \odot \mathbf{qstore} \xrightarrow{\mathsf{prep}(\llbracket M \rrbracket)} \llbracket A \rrbracket$$

Player does learn some information about the input state in the part of the exchange happening in the **qstore** arena, and this information is used to provide an answer in the **bool** arena.

We now use quantum strategies to construct a denotational semantics for the quantum store language. For each type A, we define an arena $\llbracket A \rrbracket$, and given a term $\Gamma \vdash M \colon A$, we define a strategy $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket$.

For types, the definition is given by the following inductive construction :

$$\llbracket \mathrm{bool} \rrbracket = \mathbf{bool} \quad \llbracket \mathrm{com} \rrbracket = \mathbf{com} \quad \llbracket \mathrm{qstore} \rrbracket = \mathbf{qstore}$$
$$\llbracket A \times B \rrbracket = \llbracket A \rrbracket \odot \llbracket B \rrbracket \quad \llbracket A \Rightarrow B \rrbracket = \llbracket A \rrbracket \multimap \llbracket B \rrbracket$$

The arena **com** is defined with the moves “run” and “done” as in chapter 4. The quantum store type is interpreted using the arena **qstore**.

Given a context $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, we set $\llbracket \Gamma \rrbracket$ to be $\llbracket A_1 \rrbracket \odot \cdots \odot \llbracket A_n \rrbracket$. The interpretation $\llbracket \Gamma \vdash M \colon A \rrbracket$ is defined by induction on the derivation of $\Gamma \vdash M \colon A$ in what follows.

We begin the definition of $[\![\Gamma \vdash M \colon A]\!]$ with the base cases of variables and constant terms. The interpretation of $\Gamma, x \colon A \vdash x \colon A$ uses the projection strategy π_A. The Boolean constants 0, 1 are interpreted as their corresponding deterministic strategies in **bool**. The constant skip is interpreted as the unique non-trivial deterministic strategy skip in **com**.

The strategy

$$\left[\!\!\left[U^{y_1 \otimes \cdots \otimes y_m}\, x_1 \otimes \cdots \otimes x_n \right]\!\!\right]$$

corresponding to a unitary transformation is defined as the strategy $[U] \colon \mathbf{qstore} \multimap \mathbf{com}$. In the case of measurements, $[\![\mathsf{meas}\, x_i]\!]$ is interpreted using the meas strategy.

We now turn to the inductive cases. The definition of $[\![M_1 ; M_2]\!]$ follows the standard idea in game semantics: it is defined as the composition $\langle [\![M_1]\!] , [\![M_2]\!] \rangle ; \mathsf{seq}$, where seq is the strategy $\mathbf{com} \odot \mathbf{com} \multimap \mathbf{com}$ defined with the following typical play:

$$\begin{array}{ccccc}
\mathbf{com} & \odot & \mathbf{com} & \overset{\mathsf{seq}_{\mathrm{com}}}{\xrightarrow{\qquad\qquad}\!\!\circ} & \mathbf{com} \\
 & & & & \mathrm{run} \\
\mathrm{run} & & & & \\
\mathrm{done} & & & & \\
 & & \mathrm{run} & & \\
 & & \mathrm{done} & & \\
 & & & & \mathrm{done}
\end{array}$$

Using this scheme, the commands M_1 and M_2 are successively ran when seq is composed with $\langle [\![M_1]\!] , [\![M_2]\!] \rangle$.

For terms of the form $\Gamma \vdash \lambda \overline{x}.\, N \colon B$, where $\Gamma, \overline{x} \colon A \vdash N \colon B$, we define $[\![\lambda \overline{x}.\, N]\!]$ to be $\Lambda\,([\![N]\!])$, using the adjunction

$$\frac{[\![N]\!] :\ [\![\Gamma]\!] \odot [\![A]\!] \to [\![B]\!]}{\Lambda\,([\![N]\!]) :\ [\![\Gamma]\!] \to [\![A]\!] \multimap [\![B]\!]}$$

The other classical operations are also interpreted using the usual game semantics ideas. We refer the reader to [Har99] for a detailed account.

For quantum store creation using new, suppose that the denotation of

$$\Gamma, x_1 \otimes \cdots \otimes x_n \colon \text{qstore} \vdash M \colon A$$

is already defined. The term $\mathsf{new}\, x_1 \otimes \cdots \otimes x_n\ \mathsf{in}\ M$ is interpreted as the composition $\langle \mathrm{id}_{[\![\Gamma]\!]}, [|0 \ldots 0\rangle\langle 0 \ldots 0|] \rangle ; [\![M]\!]$. The strategy $[|0\rangle\langle 0|]$ is used to initiate the state of the new quantum store.

The last case is for the preparation typing rule. The strategy

$$[\![\mathsf{prep}\, \overline{y}\ \mathsf{with}\ \overline{x}\ \mathsf{in}\ M]\!]$$

is defined as the strategy

$$\mathrm{prep}\,([\![M]\!]) :\ [\![\Gamma]\!] \odot \mathbf{qstore} \multimap [\![A]\!]$$

defined with the following idea. Let $\mathcal{F}_0$ be the preparation superoperator taking ρ to $\rho \otimes |0 \dots 0\rangle\langle 0 \dots 0|$. Player plays using prep $([\![M]\!])$ by making the moves prescribed by $[\![M]\!]$ except that before playing his first move in the qstore arena, he must initiate an exchange in this arena which forces Opponent to add the $|0 \cdots 0\rangle$ state to the state ρ she uses to answer Player's questions about the state of the quantum store. This is achieved by playing a $\{\mathcal{F}_0\}_?$ quantum intervention question in the **qstore** arena before any other move is played there:

$$\begin{array}{ccccc} [\![\Gamma]\!] & \odot & \mathbf{qstore} & \overset{\mathrm{prep}([\![M]\!])}{\multimap} & [\![A]\!] \\ &&& \vdots & \\ && \{\mathcal{F}_0\}_? && \\ && 0 && \\ && \mathcal{F}_{?[1]} && \\ && m && \\ &&& \vdots & \end{array}$$

This completes the definition of the denotational semantics.

5.3 Soundness

To study the relation between the operational and denotational semantics, we need to take quantum stores into account. We use the standard approach used in game semantics of classical stores, described in the last chapter for the language MCdata: we define a strategy

$$[\![Q, M]\!] : I\colon\ [\![A]\!]$$

for each pair Q, M where M is of type A. This strategy is defined as the composition of $[\![M]\!]$ with a strategy $[\![Q]\!]$ representing the state of the quantum registers in Q. For each extended variable $x_1 \otimes \cdots \otimes x_n \in |Q|$, the state $|x_1 \dots x_n\rangle_Q$ can be described as a strategy $[|x_1 \dots x_n\rangle]$ in $I \multimap$ **qstore**. The strategy $[\![Q]\!]$ associated to the quantum store Q is defined as the $\odot$-product of all the strategies $[|x_1 \dots x_n\rangle]$, $x_1 \otimes \cdots \otimes x_n \in |Q|$.

Lemma 5.6. *(Substitution for QSL) For any QSL terms $\Gamma, x\colon A \vdash M\colon B$ and $\Gamma \vdash N\colon A$ with $x \in \mathrm{FV}(M)$, we have that $\Gamma \vdash M\,[N/x] : B$ and $[\![M[N/x]]\!] = \langle \mathrm{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle ; [\![M]\!]$.*

Proof. By structural induction on the construction of M. Since the proof for the classical cases is well-known, we show how some typical classical cases are dealt with and then focus on the cases involving quantum operations.

Suppose $\Gamma, x\colon A \vdash M\colon B$.

If M is a variable $\overline{x}$, then $\overline{x}[N/\overline{x}] = N$ and it is immediate that $\Gamma \vdash \overline{x}[N/\overline{x}]\colon A$. Moreover, $[\![\overline{x}[N/\overline{x}]]\!] = [\![N]\!] :\ [\![\Gamma]\!] \multimap [\![A]\!]$ which is equal to $\langle \mathrm{id}_{[\![\Gamma]\!]}, [\![N]\!]\rangle ; [\![\overline{x}]\!] = [\![N]\!]$ since $[\![\overline{x}]\!]$ is the projection strategy on $[\![A]\!]$.

For a term M of the form $\lambda\overline{y}.\, M'$, $\overline{y}$ must be different than $\overline{x}$ because we suppose that $\overline{x} \in \mathrm{FV}(\lambda\overline{y}.\, M')$. The induction hypothesis is that the proposition holds for

$$\Gamma, \overline{x} \colon A, \overline{y} \colon C \vdash M \colon B,$$

i.e. that $\Gamma \vdash M'[N/\overline{x}] \colon B$ and $\llbracket M'[N/\overline{x}] \rrbracket = \langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \llbracket M' \rrbracket$. Since

$$(\lambda\overline{y}.\, M')[N/\overline{x}] = \lambda\overline{y}.\ (M'[N/\overline{x}])\,,$$

we have that $\Gamma \vdash \lambda\overline{y}.\, M'[N/\overline{x}] \colon B$ and that

$$\begin{aligned}
\llbracket (\lambda\overline{y}.\, M')[N/\overline{x}] \rrbracket &= \Lambda\,(\llbracket M'[N/\overline{x}] \rrbracket) \\
&= \Lambda\,(\langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \llbracket M \rrbracket) \\
&= \langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \Lambda\,(\llbracket M' \rrbracket) \\
&= \langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \llbracket \lambda\overline{y}.\, M' \rrbracket\,,
\end{aligned}$$

where the third equality follows from the naturality of the adjunction Λ.

The quantum cases are dealt with in a similar manner. Consider the unitary operation case. Suppose that $M = U^{\overline{y}}\,\overline{x}$, with $\overline{y} \sqsubseteq \overline{x}$. Since $\overline{x} \colon \mathrm{qstore}$, N must be a qstore variable $\overline{x'}$. Let $\overline{y'}$ be the subvariable of $\overline{x'}$ corresponding to the same qbits as $\overline{y}$ in $\overline{x}$. We have that

$$\Gamma \vdash U^{\overline{y}}\,\overline{x}\,[N/\overline{x}] = U^{\overline{y'}}\,\overline{x'} \colon \mathrm{com}$$

and that

$$\left\llbracket U^{\overline{y}}\,\overline{x}\,[N/\overline{x}] \right\rrbracket = \left\llbracket U^{\overline{y'}}\,\overline{x'} \right\rrbracket = \left\langle \mathrm{id}_{\Gamma}, \left\llbracket \overline{x'} \right\rrbracket \right\rangle ; \llbracket M \rrbracket\,.$$

The measurement case is similar to the unitary transformation case.

Finally, consider that M is a preparation term

$$\Gamma, \overline{x} \colon A, \overline{y} \colon \mathrm{qstore} \vdash \mathsf{prep}\,\overline{z}\,\mathsf{with}\,\overline{y}\,\mathsf{in}\,M' \colon B$$

and assume that the lemma holds for

$$\Gamma, \overline{x} \colon A, \overline{y} \otimes \overline{z} \colon \mathrm{qstore} \vdash M' \colon A.$$

Substitution of N for $\overline{x}$ in M yields

$$(\mathsf{prep}\,\overline{z}\,\mathsf{with}\,\overline{y}\mathsf{in}\,M')\,[N/\overline{x}] = \mathsf{prep}\,\overline{z}\,\mathsf{with}\,\overline{y}\,\mathsf{in}\ (M\,[N/\overline{x}])\,,$$

and thus by induction hypothesis $\Gamma, \overline{y} \colon \mathrm{qstore} \vdash M[N/\overline{x}] \colon B$. Furthermore, we have by definition of the preparation strategy that

$$\begin{aligned}
\llbracket (\mathsf{prep}\,\overline{z}\,\mathsf{with}\,\overline{y}\,\mathsf{in}\,M')\,[N/\overline{x}] \rrbracket &= \llbracket \mathsf{prep}\,\overline{z}\,\mathsf{with}\,\overline{y}\mathsf{in}\ (M'\,[N/\overline{x}]) \rrbracket \\
&= \mathrm{prep}\,(\llbracket M'\,[N/\overline{x}] \rrbracket) \\
&= \mathrm{prep}\,(\langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \llbracket M' \rrbracket) \\
&= \langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \mathrm{prep}\,(\llbracket M' \rrbracket) \\
&= \langle \mathrm{id}_{\llbracket\Gamma\rrbracket}, \llbracket N \rrbracket \rangle ; \llbracket M \rrbracket
\end{aligned} \qquad \square$$

It is now possible to state and prove the following result.

Proposition 5.7. *Let M and V be two terms of ground type. If $Q, M \Downarrow^p Q', V$, then for all well-opened $sab \in \mathcal{T}(\llbracket Q', V \rrbracket)$ we have that*

$$\llbracket Q, M \rrbracket\,(\, b \mid sa\,) = p\,\llbracket Q, V \rrbracket\,(\, b \mid sa\,).$$

Proof. The proof is a structural induction on the derivation of $Q, M \Downarrow^p Q', V$.

We show how to deal with some typical classical cases. First, we deal with the base cases.

For the cases of the form $Q, V \Downarrow Q, V$, the proposition is trivial.

In the case of a unitary transformation operation U, suppose that

$$Q, U\, x_1 \otimes \cdots \otimes x_n \Downarrow Q\,[|x_1 \ldots x_n\rangle \mapsto U|x_1 \ldots x_n\rangle]\,, \mathsf{skip}$$

holds. By definition of the denotational semantics, we have that $\left\llbracket Q, U^{\bar{y}}\,\overline{x} \right\rrbracket$ is the composition

$$I \xrightarrow{\llbracket Q \rrbracket} \llbracket \Gamma \rrbracket \xrightarrow{\llbracket x_1 \otimes \cdots \otimes x_n \rrbracket} \mathbf{qstore} \xrightarrow{[U]} \mathbf{com}$$

A run move in the final **com** arena is answered with the question $\{\mathcal{U}_0\}_?$ in the **qstore** arena and then copied by the projection strategy to the $\llbracket \Gamma \rrbracket$ arena, where an interaction begins with $\llbracket Q \rrbracket$ in which the unitary transformation move $\{\mathcal{U}_0\}_?$ is made, affecting all subsequent interactions in the **qstore** component. The 0 answers that Opponent gives back to Player is copied back to the initial **qstore** arena, and then a "done″ move is made in the **com** arena. In any further interaction with the quantum store strategy $\llbracket Q \rrbracket$ Player will behave as if he is using the strategy

$$\left\llbracket Q[|x_1 \ldots x_n\rangle \mapsto U^{\bar{y}}|x_1 \ldots x_n\rangle] \right\rrbracket .$$

If Player uses the strategy

$$\left\llbracket Q[|x_1 \ldots x_n\rangle \mapsto U^{\bar{y}}|x_1 \ldots x_n\rangle], \mathsf{skip} \right\rrbracket ,$$

then the behaviour is the same: the initial "run" move is answered with "done" without interacting with the strategy

$$\llbracket Q\,[|x'_1 \ldots x'_{n'}\rangle \mapsto U|x'_1 \ldots x'_{n'}\rangle] \rrbracket .$$

The two rules for quantum measurement operations are dealt with similarly. Suppose that

$$Q, \mathsf{meas}\, x_i \Downarrow^{\mathrm{tr}([0]^{x_i}[x_1 \ldots x_n])} Q\,[|x_1 \ldots x_n\rangle \mapsto [0]^{x_i}|x_1 \ldots x_n\rangle / ||[0]^{x_i}|x_1 \ldots x_n\rangle]\,, 0.$$

By definition we have that $\llbracket Q, \mathsf{meas}\, x_i \rrbracket$ is the strategy $\llbracket Q \rrbracket\,;\llbracket x_i \rrbracket\,;\mathrm{meas}$ in the arena

$$I \multimap \llbracket \Gamma \rrbracket \multimap \mathbf{qstore} \multimap \mathbf{bool}.$$

Any interaction starting with the question ? in **bool** is answered by measuring in the canonical basis the qbit of the arena **qstore**. The answer to this is given according to

$[\![Q]\!]$ and is 0 with probability $\|[0]^{x_i}|x_1 \dots x_n\rangle\|$. Any further interaction with $[\![Q]\!]$ will be made according to

$$[\![Q[|x_1 \dots x_n\rangle \mapsto [0]^{x_i}|x_1 \dots x_n\rangle]]\!]\,,$$

and the answer to the initial question in **bool** is 0. This amounts to saying that $[\![Q, \mathsf{meas}\ x_i]\!]$ behaves like

$$\left[\!\!\left[Q\left[|x_1 \dots x_n\rangle \mapsto [0]^0|x_1 \dots x_n\rangle\right], 0\right]\!\!\right]$$

with probability $\|[0]^{x_i}|x_1 \dots x_n\rangle\|$. The other measurement case is similar.

We now deal with some typical induction cases.

For conditionals, suppose that the proposition holds when

$$Q, P \Downarrow^p Q', 0 \text{ and } Q', N \Downarrow^q Q'', V.$$

Assume that $Q, \mathsf{if}\ P\ \mathsf{then}\ M\ \mathsf{else}\ N \Downarrow^{pq} Q'', V$. By definition, we have that

$$[\![Q, \mathsf{if}\ P\ \mathsf{then}\ M\ \mathsf{else}\ N]\!]$$

is the composition

$$\begin{tikzcd} I \arrow[r, "{[\![Q]\!]}"] & {[\![\Gamma]\!]} \arrow[r, "{\langle [\![P]\!], [\![M]\!], [\![N]\!] \rangle}"] & \mathbf{bool} \odot [\![A]\!] \odot [\![A]\!] \arrow[r, "\mathsf{cond}"] & {[\![A]\!]} \end{tikzcd}$$

An initial move in the final $[\![A]\!]$ arena will make Player ask for a Boolean in the bool input of cond. Opponent will answer using $[\![Q]\!] ; [\![P]\!]$, which, by hypothesis, with probability p will make her answer as if using the strategy $[\![Q']\!] ; [\![0]\!]$. After that Player will play according to the strategy $[\![Q']\!] ; [\![N]\!]$, which with probability q makes him behave as if using $[\![Q'']\!] ; [\![V]\!]$. After hiding, we see that using $[\![Q]\!] ; [\![\mathsf{if}\ P\ \mathsf{then}\ M\ \mathsf{else}\ N]\!]$, Player will play as using the strategy $[\![Q]\!] ; [\![V]\!]$ with probability pq. The other conditional case is treated similarly.

In the case of application, suppose that the proposition is true when

$$Q, M \Downarrow^p Q', \lambda\overline{x}.\ M' \text{ and } Q', M'[N/\overline{x}] \Downarrow^q Q'', V.$$

Assume that $Q, MN \Downarrow^{pq} Q'', V$. By definition we have that $[\![Q, MN]\!]$ is

$$\begin{tikzcd} I \arrow[r, "{[\![Q]\!]}"] & {[\![\Gamma]\!]} \arrow[r, "{\langle [\![M]\!], [\![N]\!] \rangle}"] & ([\![A]\!] \multimap [\![B]\!]) \odot [\![A]\!] \arrow[r, "\mathsf{eval}"] & {[\![B]\!]}\ . \end{tikzcd}$$

A move in the final $[\![B]\!]$ is copied to $[\![A]\!] \multimap [\![B]\!]$, where it is answered using the strategy $[\![Q, M]\!]$. By induction hypothesis, with probability p this answer is given as if using the strategy $[\![Q', \lambda\overline{x}.\ M']\!]$. So with probability p the plays of $[\![Q, MN]\!]$ are the same as those of $[\![Q', (\lambda\overline{x}.\ M')N]\!]$. By lemma 5.6, this is the same as $[\![Q', M[N/\overline{x}]]\!]$, which by induction hypothesis is the same as $[\![Q'', V]\!]$ with probability q. So the plays of $[\![Q, MN]\!]$ are the same as those of $[\![Q'', V]\!]$ with probability pq.

For the new operation, assume that the proposition holds when

$$Q[|x_1 \dots x_n\rangle \mapsto |0 \dots 0\rangle], M \Downarrow^p Q', V.$$

Suppose that

$$Q, \mathsf{new}\, x_1 \otimes \cdots \otimes x_n \,\mathsf{in}\, M \Downarrow Q', V.$$

By definition we have that $[\![\mathsf{new}\, x_1 \otimes \cdots \otimes x_n \,\mathsf{in}\, M]\!]$ is the composition

$$\begin{tikzcd} I \arrow[r, multimap, "{[\![Q]\!]}"] & {[\![\Gamma]\!]} \arrow[r, multimap, "{\langle I,[|0...0\rangle\langle 0...0|]\rangle}"] & {[\![\Gamma]\!] \odot \textbf{qstore}} \arrow[r, multimap, "{[\![M]\!]}"] & {[\![A]\!]} \end{tikzcd}$$

Since $[\![Q[|x_1 \ldots x_n\rangle \mapsto |0 \ldots 0\rangle]]\!]$ is equal to $[\![Q]\!]\,; I \odot [|0 \ldots 0\rangle\langle 0 \ldots 0|]$, the above composition is equal to $[\![Q[|x_1 \ldots x_n\rangle \mapsto |0 \ldots 0\rangle], M]\!]$, which by induction hypothesis is $[\![Q', V]\!]$.

The most interesting induction case is the preparation case. Suppose that the proposition holds when

$$Q\,[|x_1 \ldots x_n\rangle|y_1 \ldots y_m\rangle \mapsto |\varphi\rangle|0 \ldots 0\rangle] \Downarrow^p Q', V.$$

Assume that

$$Q\,[|x_1 \ldots x_n\rangle \mapsto |\varphi\rangle]\,, \mathsf{prep}\,\overline{y}\,\mathsf{with}\,\overline{x}\,\mathsf{in}\, M \Downarrow^p Q', V.$$

By definition of

$$[\![\Gamma, \overline{x} \colon \text{qstore} \vdash \mathsf{prep}\,\overline{y}\,\mathsf{with}\,\overline{x}\,\mathsf{in}\, M \colon A]\!]\,,$$

any play in $[\![\Gamma]\!] \odot$ **qstore** $\multimap$ $[\![A]\!]$ will be played with player using the strategy $[\![M]\!]$, except that a preparation move is made in **qstore**. This preparation move is answered by Opponent using the strategy

$$[\![Q\,[|x_1 \ldots x_n\rangle \mapsto |\varphi\rangle]]\!]\,,$$

which make her pick her answers using the strategy $[|\varphi\rangle\langle\varphi|]$. After the preparation move, Opponent will play as if she is using the strategy $[|\varphi\rangle\langle\varphi||0 \ldots 0\rangle\langle 0 \ldots 0|]$, which is

$$[\![Q\,[|x_1 \ldots x_n\rangle|y_1 \ldots y_m\rangle \mapsto |\varphi\rangle|0 \ldots 0\rangle]]\!]\,.$$

The overall play is thus just like what would happen if Player uses $[\![M]\!]$ composed with this last strategy. We get the desired result because the induction hypothesis implies that composed strategy dictates the same moves to Players as the strategy $[\![Q', V]\!]$. □

We now turn to the converse problem: proving adequacy for the QSL.

A term $\Gamma \vdash M \colon A$ is said to be **semi-closed** if $\mathrm{FV}(M)$ contains only variables to type qstore. The **ground types** are all the constants types.

Proposition 5.8. *(Adequacy for QSL) Let M be a semi-closed term of ground type. If for all well opened sab* $\in \mathcal{T}([\![Q', V]\!])$ *we have that*

$$[\![Q, M]\!]\,(b \mid sa) = p\,[\![Q, V]\!]\,(b \mid sa),$$

then we must also have that

$$Q, M \Downarrow^p Q', V.$$

We use the standard proof technique that uses a computability predicate. We refer the reader to [Gun92] for an expository account of adequacy proofs for the language PCF which uses this technique. The usual definition of computability predicate is adapted to quantum stores as follows.

Definition 5.9. *(Computability for QSL) Let $\Gamma_1, \Gamma_2 \vdash M\colon A$, with Γ_1 containing only variable of type* qstore*. We say M is* ***computable*** *if*

1. *$\Gamma_1 \vdash M\colon A$, $A =$* bool, qstore, $\top$ *or* com *and if for all $sab \in \mathcal{T}$ ($[\![Q', V]\!]$) we have that $[\![Q, M]\!]\,(b \mid sa) = p\,[\![Q', V]\!]\,(b \mid sa)$, then $Q, M \Downarrow^p Q', V$,*
2. *$\Gamma_1, \overline{x}_1\colon A_1, \ldots, \overline{x}_n\colon A_n \vdash M\colon A$ is $\Gamma_1 \vdash M[N_1/\overline{x}_1, \ldots, N_n/\overline{x}_n]\colon A$ is computable for all computable semi-closed terms $\Gamma_1 \vdash N_1\colon A_1, \ldots, \Gamma_1 \vdash N_n\colon A_n$,*
3. *$\Gamma_1 \vdash M\colon A \multimap B$, M semi-closed and for all semi-closed $\Gamma_1 \vdash N\colon A$ we have that $\Gamma_1 \vdash MN\colon B$ is computable,*
4. *$M = \overline{x}$ with $\Gamma_1 \vdash \overline{x}\colon$* qstore *and both $\Gamma_1 \vdash \mathsf{meas}\, x_i\colon$* bool *and $\Gamma_1 \vdash U\,\overline{y}\colon$* com *with $\overline{y} \sqsubseteq \overline{x}$ are computable.*

Proposition 5.8 is a direct consequence of the following lemma.

Lemma 5.10. *All QSL terms are computable.*

In order to prove this lemma, we need the following result:

Lemma 5.11. *For any type A, there exist a semi-closed term M such that $\Gamma \vdash M\colon A$.*

Proof. By induction on the construction of A. If A is bool or com, we can take M to be the constant true or respectively skip. If $A =$ qstore, then taking $M = x$ we have the semi-closed term $x\colon A \vdash x\colon A$.

If A is a product $B_1 \times B_2$, assume inductively that there are terms $\Gamma_1 \vdash M_1\colon B_1$ and $\Gamma_2 \vdash M_1\colon B_2$. Without loss of generality, we can also assume that $|\Gamma_1| \cap |\Gamma_2| =$, renaming variables if necessary. Then $M = \langle M_1, M_2\rangle$ is a term such that $\Gamma_1, \Gamma_2 \vdash M\colon A$. Similarly, if $A = B_1 \Rightarrow B_2$, assume that there is a term $\Gamma \vdash N\colon B_2$. Then if $x \notin |\Gamma|$, we can take the term $\Gamma \vdash \lambda x. N\colon B_1 \Rightarrow B_2$. □

Proof of lemma 5.10. By induction on the construction of M. By the second and third clauses of the definition of computability, we can assume that M is constructed out of semi-closed terms. We explain the most interesting part of the proof, leaving out the cases which are standard classical cases.

For the base case, M must be a constant, a classical variable x or a quantum store variable $\overline{x}$. If $M = \overline{x}$ is a quantum store variable, we must apply the last clause of the definition of computability. We need to check that both

$$\Gamma_1 \vdash \mathsf{meas}\, x_i\colon \text{bool and } \Gamma_1 \vdash U\,\overline{y}\colon \text{com}, \overline{y} \sqsubseteq \overline{x}$$

are computable. In the first case, suppose that $[\![Q, \mathsf{meas}\, x_i]\!]$ makes Player behave as $[\![Q', V]\!]$ for some boolean value V. This means that measuring the qbit i of the quantum store $\overline{x}$ with the quantum store in some state Q gives the boolean result V (without loss of generality, suppose that $V = 0$) with probability p and a quantum store left in state $Q\,[|x_1 \ldots x_n\rangle \mapsto [0]^{x_i}|x_1 \ldots x_n\rangle]$. This implies that $Q, \mathsf{meas}\, x_i \Downarrow^p Q', V$. A similar argument is used to show that $\Gamma_1 \vdash U\,\overline{y}\colon$ com is computable.

For the induction step, we assume that M is constructed out of semi-closed computable terms.

For example, to show that $\lambda\overline{x}.\,M$ is computable, we assume that M is a semi-closed computable term. Since $\lambda\overline{x}.\,M$ is of type $A \Rightarrow B$, we have to use the third clause of the definition of computability. Using lemma 5.11, we can take a computable semi-closed $N\colon A$ and consider $(\lambda\overline{x}.\,M)\,N$. Assume that $[\![Q,(\lambda\overline{x}.\,M)N]\!] = p\,[\![Q',V]\!]$. We have to show that $Q,(\lambda\overline{x}.\,M)\,N \Downarrow^p Q',V$. By the properties of adjunctions and the definition of $[\![\lambda\overline{x}\,M]\!]$ we have

$$\begin{aligned}[\![(\lambda\overline{x}.\,M)N]\!] &= \langle[\![\lambda\overline{x}.\,M]\!]\,,[\![M]\!]\rangle\,;\mathrm{eval}\\ &= \langle\Lambda\,([\![M]\!])\,,[\![M]\!]\rangle\,;\mathrm{eval}\\ &= \langle\mathrm{id},[\![N]\!]\rangle\,;[\![M]\!]\,;\mathrm{eval}\\ &= [\![M[N/\overline{x}]]\!]\,.\end{aligned}$$

This implies that $[\![Q,(\lambda\overline{x}.\,M)N]\!]$ is the same as $[\![Q,M[N/\overline{x}]]\!]$, which with probability p makes Player behave as if he is using $[\![Q',V]\!]$. By induction hypothesis, this implies that

$$Q,M[N/\overline{x}] \Downarrow^p Q',V.$$

Using the operational semantics derivation rules, we get that

$$Q,(\lambda\overline{x}.\,M)\,N \Downarrow^p Q'V,$$

which is the desired result.

The quantum measurement and unitary operations cases are dealt with in the same way as in the previous case of qstore variables.

In the case of local preparation, consider that

$$M = \mathsf{prep}\,\overline{y}\,\mathsf{with}\,\overline{x}\,\mathsf{in}\,N$$

is a semi-closed term. Assume that $[\![Q,M]\!]$ makes Player behave as if he was using the strategy $[\![Q',V]\!]$, with probability p. Since in the definition of $[\![Q,M]\!]$ Player plays a preparation move before the first question about the state held by $\overline{x}\otimes\overline{y}$ in Q, the answer to this question is given using

$$[\![Q\,[|x_1\ldots x_ny_1\ldots y_m\rangle \mapsto |x_1\ldots x_n\rangle|0\ldots 0\rangle]]\!]\,.$$

Thus the strategy $[\![Q',V]\!]$ make player behave as

$$[\![Q\,[|x_1\ldots x_ny_1\ldots y_m\rangle \mapsto |x_1\ldots x_n\rangle|0\ldots 0\rangle]\,,M]\!]\,.$$

By induction hypothesis, this implies that

$$Q\,[|x_1\ldots x_ny_1\ldots y_m\rangle \mapsto |x_1\ldots x_n\rangle|0\ldots 0\rangle]\,,M \Downarrow^p Q',V.$$

Using the operational semantics derivations rules, we get that $Q,M \Downarrow^p Q',V$, which is the desired result. □

Contexts for QSL are defined similarly as in the case of MCdata: a **context** with a hole of type B is a term $C[-]$ with a special free variable "$-$" of type B, i.e. it is possible to derive that $\Gamma,-\colon B \vdash C[-]\colon A$. Capture-free substitution of a term $\Gamma \vdash M\colon B$ in the context $C[-]$ is denoted by $C[M]$.

Definition 5.12. *Two semi-closed terms $\Gamma \vdash M_1 : A$ and $\Gamma \vdash M_2 : A$ are* ***contextually equivalent*** *if for all quantum stores Q and ground type context $C[-]$*

$$Q, C[M_1] \Downarrow^p Q', V \iff Q, C[M_2] \Downarrow^p Q', V.$$

This relation is denoted by $M_1 \sim M_2$.

Proposition 5.13. *(Soundness for QSL) Let M_1 and M_2 be two semi-closed QSL terms. If $[\![M_1]\!] = [\![M_1]\!]$, then $M_1 \sim M_2$.*

Proof. Suppose that $\Gamma \vdash M_1 : A$ and $\Gamma \vdash M_2 : A$ are two semi-closed terms with $[\![M_1]\!] = [\![M_2]\!]$. Take any ground type context $C[-]$ with a hole of type A and Γ-quantum store Q. Suppose that $Q, C[M_1] \Downarrow^p Q', V$. By proposition 5.7, we have that for any well-opened $sab \in \mathcal{T}([\![Q', V]\!])$

$$[\![Q, C[M_1]]\!](\, b \mid sa\,) = p\,[\![Q', V]\!](\, b \mid sa\,).$$

Using the hypothesis and the substitution lemma and naturality of adjunction, we have that

$$\begin{aligned} [\![Q, C[M_1]]\!] &= [\![Q]\!]\,;[\![C[M_1]]\!] \\ &= [\![Q]\!]\,;\langle\, \mathrm{id}_{[\![\Gamma]\!]}, [\![M_1]\!]\rangle; [\![C[-]]\!] \\ &= [\![Q]\!]\,;\langle\, \mathrm{id}_{[\![\Gamma]\!]}, [\![M_2]\!]\rangle; [\![C[-]]\!] \\ &= [\![Q]\!]\,;[\![C[M_2]]\!] \\ &= [\![Q, C[M_2]]\!]\,. \end{aligned}$$

We thus have that for all well-opened $sab \in \mathcal{T}([\![Q', V]\!])$

$$[\![Q, C[M_2]]\!](\, b \mid sa\,) = p\,[\![Q, V]\!](\, b \mid sa\,),$$

which implies by adequacy that $Q, C[M_2] \Downarrow^p Q', V$.

The other implication being proved with a similar argument, we get that $M_1 \sim M_2$. □

Chapter 6

λ-calculus with quantum data

In the quantum store λ-calculus presented in the last chapter, quantum states can only be accessed indirectly through references. We now introduce another quantum λ-calculus in which a quantum state can be represented and manipulated directly in the language. We want to be able to apply unitary transformations to quantum data, to prepare quantum states, to tensor and measure them, and to refer to parts of a quantum state. Since quantum states cannot be duplicated, we must make the λ-calculus with quantum data *linear*, as in the case of the quantum λ-calculus of Selinger and Valiron [SV06a]. The denotational semantics presented in this chapter will validate this choice using a different argument: quantum measurements have side effects, which forces us to use thread dependent strategies that cannot be duplicated using the duplicating strategy Δ.

The problems pointed out in section 5.1 force us to be careful when introducing the qbit tensor operation. Because of this, we also use extended variables in the quantum data language. While QSL extended variables are used as references to quantum stores, in this chapter they are used to represent quantum data.

6.1 Syntax

The syntax of the λ-calculus with quantum data language (**QDL**) is that of a classical simply typed λ-calculus with pairing and conditionals, with extra constructs that give the language enough expressiveness to encode the usual manipulations of quantum data as can be described with the low level formalism of quantum circuits.

6.1.1 Terms

The terms of QDL are defined recursively as follows:

$$\begin{aligned} M, N, P ::= {} & \overline{x} \mid * \mid 0 \mid 1 \mid \rho \mid \langle M, N\rangle \mid \mathsf{fst}\, M \mid \mathsf{snd}\, M \mid \\ & MN \mid \lambda\overline{x}.\, M \mid \mathsf{if}\, M\, \mathsf{then}\, N\, \mathsf{else}\, P \mid \\ & \mathcal{U}\, M \mid M \otimes N \mid \mathsf{let}\, b, \overline{x} = \mathsf{meas}_i\, M\, \mathsf{in}\, N \mid \mathsf{meas}\, M, \end{aligned}$$

where b, $\overline{x}$ and $\overline{y}$ are extended variables defined as in section 5.2.1, $i > 0$ is a natural number, ρ can be any density matrix and $\mathcal{U}$ is a superoperator corresponding to a unitary transformation U. Most of the syntax consists of standard λ-calculus operations. The term $\mathcal{U}\,M$ is the operation that corresponds to applying a unitary transformation to the state described by the term M. The measurement operation syntax, $\mathsf{let}\, b, \overline{x} = \mathsf{meas}_i\, M \,\mathsf{in}\, N$, means that the qbit i of the term M is measured and thereafter the measurement result is accessible in N as b and the resulting state is accessible as $\overline{x}$. Note that the variables b and $\overline{x}$ are bound in N. To measure a single qbit, we use instead the simpler syntax $\mathsf{meas}\, Q$. The set of free variables in M is denoted $\mathrm{FV}(M)$.

6.1.2 Types

The types of QDL are the following:

$$A, B ::= \text{ bool} \mid \top \mid \text{qbit}^{\otimes n} \mid A \times B \mid A \Rightarrow B.$$

where $n > 0$. The type bool is the type of boolean constants, $A \times B$ and $A \Rightarrow B$ are respectively the types of pairs and functions. The type $\text{qbit}^{\otimes n}$ is the type of quantum states on n qbits. The notation $\text{qbit}^{\otimes n}$ stands implicitly for the product $\text{qbit} \otimes \cdots \otimes \text{qbit}$; we use the notation $\text{qbit}^{\otimes n} \otimes \text{qbit}^{\otimes m}$ to denote $\text{qbit}^{\otimes(n+m)}$, although there is no $\otimes$ type operation.

The typing rules of QDL are given in table 6.1. We assume that contexts Γ contain no qbit variables and contexts Δ_k contain only qbit variables. This convention will be used throughout this chapter. Rules involving classical operations are direct adaptation of the standard typing rules of a typed λ-calculus. The rules for quantum constants, quantum measurements and unitary operations are straightforward. The three tensor rules allow one to take two terms of type $\text{qbit}^{\otimes n}$ and $\text{qbit}^{\otimes m}$ and create a term of type $\text{qbit}^{\otimes(n+m)}$. The distinction between the three cases is due to the fact that known or unknown qbits must be dealt with differently. If $\Gamma, \Delta \vdash M \colon \text{qbit}^{\otimes n}$, M is a known qbit when it has no dependency on some quantum state variable in Δ, i.e. if $\mathrm{FV}(M) \cap |\Delta| = \emptyset$. If instead $\mathrm{FV}(M) \cap |\Delta|$ contains only an extended variable $\overline{x}$, then the quantum state represented by M depends on the value of the quantum variable $\overline{x}$ and is thus unknown. The typing rules do not allow an unknown quantum state to depend upon more than one other quantum state.

Example 6.1. The term $\rho \otimes \rho$ has type $\vdash \rho \otimes \rho \colon \text{qbit}^{\otimes 2}$.

The term $x \otimes x$ is not allowed since extended variables cannot contain duplicate variables. It follows from this that there is no duplicating function $\lambda x.\, x \otimes x$ either. In the λ-calculus with quantum data, duplicating a known state ρ is possible but duplicating an unknown state x isn't.

Note that $x \colon \text{qbit} \vdash \langle x, x \rangle \colon \text{qbit} \times \text{qbit}$ is not a valid typing judgement either. This is forbidden by the pairing typing rules: to derive that

$$x \colon \text{qbit} \vdash \langle x, x \rangle \colon \text{qbit} \times \text{qbit},$$

one must start with the assumption $x \colon \text{qbit} \vdash x \colon \text{qbit}$ and then use the derivation rule

Table 6.1 QDL typing rules

$$\frac{}{\Gamma,\Delta,\overline{x}\colon A\vdash\overline{x}\colon A}\qquad\frac{}{\Gamma,\Delta\vdash *\colon\top}\qquad\frac{}{\Gamma,\Delta\vdash 0\colon\text{bool}}\qquad\frac{}{\Gamma,\Delta\vdash 1\colon\text{bool}}$$

$$\frac{\Gamma,\Delta,\overline{x}\colon A\vdash M\colon B}{\Gamma,\Delta\vdash\lambda\overline{x}.M\colon A\Rightarrow B}\qquad\frac{\Gamma,\Delta_1\vdash M\colon A\Rightarrow B\qquad\Gamma,\Delta_2\vdash N\colon A}{\Gamma,\Delta_1,\Delta_2\vdash MN\colon B}$$

$$\frac{\Gamma,\Delta_1\vdash M_1\colon A_1\qquad\Gamma,\Delta_2\vdash M_2\colon A_2}{\Gamma,\Delta_1,\Delta_2\vdash\langle M_1,M_2\rangle\colon A_1\times A_2}\qquad\frac{\Gamma,\Delta\vdash M\colon A\times B}{\Gamma,\Delta\vdash\mathsf{fst}\,M\colon A}\qquad\frac{\Gamma,\Delta\vdash M\colon A\times B}{\Gamma,\Delta\vdash\mathsf{snd}\,M\colon B}$$

$$\frac{\Gamma,\Delta_1\vdash P\colon\text{bool}\qquad\Gamma,\Delta_2\vdash M\colon A\qquad\Gamma,\Delta_2\vdash N\colon A}{\Gamma,\Delta_1,\Delta_2\vdash\mathsf{if}\,P\,\mathsf{then}\,M\,\mathsf{else}\,N\colon A}\qquad\frac{}{\Gamma,\Delta\vdash\rho\colon\text{qbit}^{\otimes n}}$$

$$\frac{\Gamma,\Delta_1\vdash Q\colon\text{qbit}^{\otimes(n+1)}\qquad\Gamma,\Delta_2,b\colon\text{bool},\overline{x}\colon\text{qbit}^{n}\vdash M\colon A}{\Gamma,\Delta_1,\Delta_2\vdash\mathsf{let}\,b,\overline{x}=\mathsf{meas}_i\,Q\,\mathsf{in}\,M\colon A}\qquad\frac{\Gamma,\Delta\vdash M\colon\text{qbit}^{\otimes n}}{\Gamma,\Delta\vdash\mathcal{U}\,M\colon\text{qbit}^{\otimes n}}$$

$$\frac{\Gamma,\Delta\vdash Q\colon\text{qbit}}{\Gamma,\Delta\vdash\mathsf{meas}\,Q\colon\text{bool}}\qquad\frac{\Gamma,\Delta_1\vdash M_1\colon\text{qbit}^{\otimes n}\qquad\Gamma,\Delta_2\vdash M_2\colon\text{qbit}^{\otimes m}}{\Gamma,\Delta_1,\Delta_2\vdash M_1\otimes M_2\colon\text{qbit}^{\otimes n}\otimes\text{qbit}^{\otimes m}}\ \text{FV}(M_i)\cap|\Delta_i|=\emptyset$$

$$\frac{\Gamma,\Delta_1,\overline{x_1}\colon\text{qbit}^{\otimes n}\vdash M_1\colon\text{qbit}^{\otimes n}\qquad\Gamma_2,\Delta_2,\overline{x_2}\colon\text{qbit}^{\otimes m}\vdash M_2\colon\text{qbit}^{\otimes m}}{\Gamma,\Delta_1,\Delta_2,\overline{x_1}\otimes\overline{x_2}\colon\text{qbit}^{\otimes n}\otimes\text{qbit}^{\otimes m}\vdash M_1\otimes M_2\colon\text{qbit}^{\otimes n}\otimes\text{qbit}^{\otimes m}}\ \text{FV}(M_i)\setminus|\Delta_i|=\{\overline{x}_i\}$$

$$\frac{\Gamma,\Delta_1,\overline{x}\colon\text{qbit}^{\otimes n}\vdash M_1\colon\text{qbit}^{\otimes n}\qquad\Gamma,\Delta_2\vdash M_2\colon\text{qbit}^{\otimes m}}{\Gamma,\Delta_1,\Delta_2,\overline{x}\colon\text{qbit}^{\otimes n}\vdash M_1\otimes M_2\colon\text{qbit}^{\otimes n}\otimes\text{qbit}^{\otimes m}}\ \begin{array}{l}\text{FV}(M_1)\setminus|\Delta_1|=\{\overline{x}_1\}\\\text{FV}(M_2)\cap|\Delta_2|=\emptyset\end{array}$$

$$\frac{x\colon\text{qbit}\vdash x\colon\text{qbit}\qquad x\colon\text{qbit}\vdash x\colon\text{qbit}}{x\colon\text{qbit},x\colon\text{qbit}\vdash\langle x,x\rangle\colon\text{qbit}\times\text{qbit}}$$

This is forbidden because contexts can only refer once to a given variable.

Example 6.2. Quantum teleportation can be implemented in the quantum data λ-calculus. Consider the term teleportation defined in figure 6.1, where the unitary superoperators $\mathcal{U}_{b_x b_y}$ are the usual correction unitary operations of the teleportation protocol. Using the type inference rules, we can derive that $\vdash$ teleport: qbit $\Rightarrow$ qbit.

Figure 6.1 QDL Teleportation

```
teleport:
  $\lambda x.$ let $b_x, y \otimes z$ = meas$_1$cnot$^{12}$ (($\mathcal{H}\,x$) $\otimes$ [$\beta_{00}$]) in
      let $b_y, z'$ = meas$_1$ $y \otimes z$ in
        if $b_x$ then
          if $b_y$ then $\mathcal{U}_{00}\,z'$ else $\mathcal{U}_{01}\,z'$
        else
          if $b_y$ then $\mathcal{U}_{10}\,z'$ else $\mathcal{U}_{11}\,z'$
```

Example 6.3. Any quantum circuit can be implemented as a QDL term. The input qbits are represented as a $\mathrm{qbit}^{\otimes n}$ variable $\overline{x}$. If some ancilla state $|\varphi\rangle$ is used, $\overline{x}$ is then tensored with $|\varphi\rangle\langle\varphi|$. The unitary transformations $U_1, \ldots, U_N$ corresponding to the quantum gates applied in the circuit are then applied. Finally, measurement operations are used to measure the qbits i_1 to i_k and return a tuple containing the measurement results.

$$\begin{array}{l} \mathsf{let}\, b_1, \overline{y_1} = \mathrm{meas}_{i_1}\, \mathcal{U}_N \ldots \mathcal{U}_1\, (x_1 \otimes \cdots \otimes x_n \otimes |\varphi\rangle\langle\varphi|)\ \mathsf{in} \\ \quad \vdots \\ \quad \mathsf{let}\, b_k, \overline{y_k} = \mathrm{meas}_{i_k}\, \overline{y_{k-1}}\ \mathsf{in} \\ \qquad \langle b_1, \ldots b_k \rangle \end{array}$$

6.2 Operational semantics

The operational semantics of the λ-calculus with quantum data is given as a big-step probabilistic reduction relation $M \Downarrow^p V$ between terms and values. **Values** are the terms defined recursively by

$$V, W ::= \ 0 \mid 1 \mid * \mid \rho \mid \lambda\overline{x}.\, M \mid \langle V, W \rangle \mid V \otimes W.$$

The reduction relation is defined by the rules given in table 6.2.

Example 6.4. Consider the term

$$M = \mathsf{if}\,(\mathsf{meas}\,|+\rangle\langle+|)\,\mathsf{then}\,\rho_1\,\mathsf{else}\,\rho_2 : \mathrm{qbit}.$$

Since $\mathsf{meas}\,|+\rangle\langle+| \Downarrow^{1/2} 0$, we have that $M \Downarrow^{1/2} \rho_1$. Similarly, $M \Downarrow^{1/2} \rho_2$.

Example 6.5. The term $\mathsf{teleport}\,\rho$ reduces with probability 1 to ρ.

6.3 Denotational semantics

We now define a denotational semantics for QDL. The first problem to solve is to find the right arena to model the type $\mathrm{qbit}^{\otimes n}$. We use the arena $\mathbf{qbit}^{\otimes n}$ defined in the same way as **qstore**, but where the quantum intervention question $\mathcal{E}_? = \left\{\mathcal{E}^?_m\right\}$ uses only quantum operations

$$\mathcal{E}^?_m : \mathrm{SD}\left(\mathbb{C}^{2n}\right) \to \mathrm{SD}\left(H_m\right),$$

i.e. all operations must take their input in the state Hilbert space $\mathbb{C}^{2n}$ for n qbits. In the case of the **qstore** arena, the dimension of the input space of the operations $\mathcal{E}^?_m$ could be any natural number $n \geq 2$ since the dimension of the state stored in a quantum store can vary in the course of a computation. For a given piece of quantum data, this dimension is fixed.

Table 6.2 QDL probabilistic reduction rules

$$\frac{}{V \Downarrow V} \qquad \frac{M \Downarrow^{p} \lambda \overline{x}.\, M' \qquad N \Downarrow^{q} V}{MN \Downarrow^{pq} M[V/\overline{x}]} \qquad \frac{M_1 \Downarrow^{p} V_1 \qquad M_2 \Downarrow^{q} V_2}{\langle M_1, M_2\rangle \Downarrow^{pq} \langle V_1, V_2\rangle}$$

$$\frac{M \Downarrow^{p} \langle V_1, V_2\rangle}{\mathsf{fst}\, M \Downarrow^{p} V_1} \qquad \frac{M \Downarrow^{p} \langle V_1, V_2\rangle}{\mathsf{snd}\, M \Downarrow^{p} V_2}$$

$$\frac{P \Downarrow^{p} 0 \qquad M \Downarrow^{q} V}{\mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \Downarrow^{pq} V} \qquad \frac{P \Downarrow^{p} 1 \qquad N \Downarrow^{q} V}{\mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \Downarrow^{pq} V}$$

$$\frac{Q \Downarrow^{q} \rho \qquad M\left[b/m, \overline{x}/\tfrac{1}{p_m}\, \mathrm{tr}^{i}([m]\rho[m])\right] \Downarrow^{r} V}{\mathsf{let}\, b, \overline{x} = \mathsf{meas}_i\, Q\, \mathsf{in}\, M \Downarrow^{p_m q r} V} \quad p_m = \mathrm{tr}\left([m]^{i}\rho\right),\ m = 0, 1$$

$$\frac{Q \Downarrow^{q} \rho}{\mathsf{meas}\, Q \Downarrow^{p_m} m} \quad p_m = \mathrm{tr}\left([m]^{i}\rho\right),\ m = 0, 1$$

$$\frac{M_1 \Downarrow^{p} V_1 \qquad M_2 \Downarrow^{q} V_2}{M_1 \otimes M_2 \Downarrow^{pq} V_1 \otimes V_2} \qquad \frac{M \Downarrow^{p} \rho}{\mathcal{U}\, M \Downarrow^{p} \mathcal{U}(\rho)}$$

With the arena **qbit**, we can define the interpretation of the QDL types recursively as follows:

$$[\![\mathrm{bool}]\!] = \mathbf{bool} \qquad [\![\top]\!] = \top \qquad \left[\!\left[\mathrm{qbit}^{\otimes n}\right]\!\right] = \mathbf{qbit}^{\otimes n}$$
$$[\![A \Rightarrow B]\!] = [\![A]\!] \multimap [\![B]\!] \qquad [\![A \times B]\!] = [\![A]\!] \odot [\![B]\!]$$

Apart from the definition of $\left[\!\left[\mathrm{qbit}^{\otimes n}\right]\!\right]$, this definition is similar to the corresponding definition for QSL. Given a context

$$\Gamma = x_1 \colon A_1, \ldots, x_n \colon A_n,$$

we set $[\![\Gamma]\!]$ to be $[\![A_1]\!] \odot \cdots \odot [\![A_n]\!]$.

We now turn to the definition of the interpretation $[\![M]\!]$ of a term $\Gamma \vdash M \colon A$. The definition is by induction on the derivation of $\Gamma \vdash M \colon A$; it is summarised in table 6.3.

In the base case we must deal with variable and constant terms. For variables, the interpretation of $\Gamma, \overline{x} \colon A \vdash \overline{x} \colon A$ is defined using the projection strategies

$$\pi_A \colon\ [\![\Gamma]\!] \odot [\![A]\!] \to [\![A]\!]\,.$$

As for QSL, the denotations of the constants 0, 1, and $*$ are the constant strategies. A quantum state constant $\rho \colon \mathrm{qbit}^{\otimes n}$ is interpreted as the quantum strategy $[\rho]$ in $\mathbf{qbit}^{\otimes n}$.

We describe the interesting inductive cases. The other cases are interpreted using the same ideas used for QSL in the last chapter.

The definition of the strategy

$$[\![\Gamma, \Delta_1, \Delta_2 \vdash \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \colon A]\!]$$

Table 6.3 QDL denotational semantics

$$\llbracket \Gamma, \Delta, \overline{x} \colon A \vdash \overline{x} \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \odot \llbracket A \rrbracket \xrightarrow{\pi_{\llbracket A \rrbracket}} \llbracket A \rrbracket \qquad \llbracket \Gamma, \Delta \vdash * \colon \top \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{*} \top$$

$$\llbracket \Gamma, \Delta \vdash 0 \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{0} \mathbf{bool} \qquad \llbracket \Gamma, \Delta \vdash 1 \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{1} \mathbf{bool}$$

$$\llbracket \Gamma, \Delta \vdash \lambda \overline{x}.\, M \colon A \Rightarrow B \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{\Lambda(\llbracket M \rrbracket)} \llbracket A \rrbracket \multimap \llbracket B \rrbracket$$

$$\llbracket \Gamma, \Delta_1, \Delta_2 \vdash MN \colon B \rrbracket : \quad \begin{tikzcd} \llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \arrow[d, "r"'] & (\llbracket A \rrbracket \multimap \llbracket B \rrbracket) \odot \llbracket A \rrbracket \arrow[r, "\mathsf{eval}"] & \llbracket B \rrbracket \\ (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[ur, "\llbracket M \rrbracket \odot \llbracket N \rrbracket"'] & & \end{tikzcd}$$

$$\llbracket \Gamma, \Delta_1, \Delta_2 \vdash \langle M_1, M_2 \rangle \colon A_1 \times A_2 \rrbracket :$$

$$\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \xrightarrow{r} (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \xrightarrow{\llbracket M_1 \rrbracket \odot \llbracket M_2 \rrbracket} \llbracket A_1 \rrbracket \odot \llbracket A_2 \rrbracket$$

$$\llbracket \Gamma, \Delta \vdash \mathsf{fst}\, M \colon A \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \odot \llbracket B \rrbracket \xrightarrow{\pi_{\llbracket A \rrbracket}} \llbracket A \rrbracket$$

$$\llbracket \Gamma, \Delta \vdash \mathsf{snd}\, M \colon B \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \odot \llbracket B \rrbracket \xrightarrow{\pi_{\llbracket B \rrbracket}} \llbracket B \rrbracket$$

$$\llbracket \Gamma, \Delta_1, \Delta_2 \vdash \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \colon A \rrbracket :$$

$$\begin{tikzcd} \llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \arrow[d, "r"] & \mathbf{bool} \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[r, "\mathsf{cond}(\llbracket M \rrbracket, \llbracket N \rrbracket)"] & \llbracket A \rrbracket \\ (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[ur, "\llbracket P \rrbracket \odot \mathrm{id}"'] & & \end{tikzcd}$$

$$\llbracket \Gamma, \Delta \vdash \rho \colon \mathrm{qbit}^{\otimes n} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{[\rho]} \mathbf{qbit}^{\otimes n}$$

$$\llbracket \Gamma, \Delta_1, \Delta_2 \vdash \mathsf{let}\, b, \overline{x} = \mathsf{meas}_i\, Q\, \mathsf{in}\, M \colon A \rrbracket :$$

$$\begin{tikzcd} \llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \arrow[d, "r"] & \left(\mathbf{bool} \odot \mathbf{qbit}^{\otimes n}\right) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[r, "\llbracket M \rrbracket"] & \llbracket A \rrbracket \\ (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[r, "\llbracket Q \rrbracket \odot \mathrm{id}"'] & \mathbf{qbit}^{\otimes n+1} \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[u, "\mathsf{meas}_i \odot \mathrm{id}"'] & \end{tikzcd}$$

$$\llbracket \Gamma, \Delta \vdash \mathcal{U}\, M \colon \mathrm{qbit}^{\otimes n} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{\llbracket M \rrbracket} \mathbf{qbit}^{\otimes n} \xrightarrow{[\mathcal{U}]} \mathbf{qbit}^{\otimes n}$$

$$\llbracket \Gamma, \Delta \vdash \mathsf{meas}\, Q \colon \mathrm{bool} \rrbracket : \ \llbracket \Gamma \rrbracket \odot \llbracket \Delta \rrbracket \xrightarrow{\llbracket Q \rrbracket} \mathbf{qbit}^{\otimes n} \xrightarrow{\mathsf{meas}} \mathbf{bool}$$

$\llbracket \Gamma, \Delta_1, \Delta_2 \vdash M_1 \otimes M_2 \colon \mathrm{qbit}^{\otimes n} \otimes \mathrm{qbit}^{\otimes m} \rrbracket$:

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \arrow[r, "r"] & (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[r, "\llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket"] & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m}
\end{tikzcd}$$

$\llbracket \Gamma, \Delta_1, \Delta_2, \overline{x_1} \otimes \overline{x_2} \colon \mathrm{qbit}^{\otimes n} \otimes \mathrm{qbit}^{\otimes m} \vdash M_1 \otimes M_2 \colon \mathrm{qbit}^{\otimes n} \otimes \mathrm{qbit}^{\otimes m} \rrbracket$:

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \odot \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \arrow[d, "r"'] & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \\
(\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \odot \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \arrow[ur, "\llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket"'] &
\end{tikzcd}$$

$\llbracket \Gamma, \Delta_1, \Delta_2, \overline{x} \colon \mathrm{qbit}^{\otimes n} \vdash M_1 \otimes M_2 \colon \mathrm{qbit}^{\otimes n} \otimes \mathrm{qbit}^{\otimes m} \rrbracket$:

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket \arrow[d, "r"'] & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \\
(\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot \mathbf{qbit}^{\otimes n} \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \arrow[ur, "\llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket"'] &
\end{tikzcd}$$

differs from the usual definition of conditional strategies used in game semantics because of the linearity constraint. Assume that

$$\llbracket P \rrbracket : \llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \multimap \mathbf{bool} \qquad \llbracket M \rrbracket , \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket \multimap \llbracket A \rrbracket$$

are already defined. Using the symmetry strategy associated to $\odot$ and the duplicating strategy Δ, we can define a strategy

$$r\colon (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \odot \llbracket \Delta_2 \rrbracket) \multimap (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket) \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket)$$

which reorganizes the input arena. With this strategy, we can define $\llbracket \mathsf{if}\, P\, \mathsf{then}\, M\, \mathsf{else}\, N \rrbracket$ to be the composition

$$r; \llbracket P \rrbracket \odot \mathrm{id}; \mathsf{cond}(\llbracket M \rrbracket , \llbracket N \rrbracket),$$

where

$$\mathsf{cond}(\llbracket M \rrbracket , \llbracket N \rrbracket)\colon \mathbf{bool} \odot (\llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket) \multimap \llbracket A \rrbracket$$

is defined using a conditional strategy operation defined in general by the following idea. Given any two arenas A and B and two strategies $\sigma, \tau\colon A \to B$, the strategy

$$\mathsf{cond}(\sigma, \tau)\colon (\mathbf{bool} \odot A) \multimap B$$

is the strategy that makes Player answer an initial move in B by asking for a Boolean b in the **bool** component and then makes Player play in the components A and B using the strategy σ if $b = 1$ and τ if $b = 0$.

The first quantum operation we deal with is unitary transformations. In this case we assume that the strategy $\left\llbracket \Gamma, \Delta \vdash M\colon \mathrm{qbit}^{\otimes n} \right\rrbracket$ is already defined. The strategy $\llbracket \mathcal{U}\, M \rrbracket$ is defined to be $\llbracket M \rrbracket ; [\mathcal{U}]$, where $[\mathcal{U}]$ is the strategy corresponding to the superoperator $\mathcal{U}$.

For the measurement case, suppose that

$$\llbracket Q \rrbracket : \llbracket \Gamma \rrbracket \odot \llbracket \Delta_1 \rrbracket \multimap \mathbf{qbit}^{\otimes(n+1)} \text{ and } \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \odot \llbracket \Delta_2 \rrbracket \odot \mathbf{bool} \odot \mathbf{qbit}^{\otimes n}$$

are already defined. We can define

$$[\![\mathsf{let}\, b, \overline{x} = \mathsf{meas}_i\, Q\, \mathsf{in}\, M]\!]$$

as the composition

$$r; ([\![Q]\!] \odot \mathrm{id}); (\mathrm{meas}_i \odot \mathrm{id}); [\![M]\!]$$

where $\mathrm{meas}_i \colon \mathbf{qbit}^{\otimes n+1} \to \mathbf{bool} \odot \mathbf{qbit}^{\otimes n}$ is the strategy described as follows. Let C be the quantum intervention corresponding to a projective measurement in the canonical basis and $\mathcal{I}$ be the identity quantum intervention. If the first move is a question in the $\mathbf{qbit}^{\otimes n}$ arena, Player uses the left-hand scheme of figure 6.2 and if the first move is in the **bool** arena, then Player uses the right-hand scheme. In these schemes, $\mathcal{E} \otimes \mathcal{F}$ stands for the quantum intervention $\{\mathcal{E}_{m_1} \otimes \mathcal{F}_{m_2}\}_{(m_1,m_2)}$. It is important to point out that in the right-hand scheme, Player must question Opponent two times. Since the first intervention $\mathcal{I} \otimes C$ alters the state, Opponent's answer to the second question $\mathcal{E}_? \otimes \mathcal{I}^i$ depends on the first answer given. This is the only instance in the semantics described in this chapter where more than one thread is necessary in the $\mathrm{qbit}^{\otimes n}$ arena. Because of the side effects of measurements, we are forced to use thread dependent strategies to describe quantum states. This is the point where we are forced to assume that qbit types are linear, since thread dependent strategies cannot be duplicated using the usual Δ duplicating strategy. In contrast, previous work on quantum λ-calculi justified the need of the linearity hypothesis by invoking the no-cloning theorem.

Figure 6.2 Strategy for the QDL measurement rules

$\mathbf{qbit}^{\otimes(n+1)} \xrightarrow{\text{meas}_i}$	**bool**	$\odot$	$\mathbf{qbit}^{\otimes n}$
			$\mathcal{E}_?$
$\mathcal{E}_? \otimes C$			
(m, b)			
			m
	$?$		
	b		

$\mathbf{qbit}^{\otimes(n+1)} \xrightarrow{\text{meas}_i}$	**bool**	$\odot$	$\mathbf{qbit}^{\otimes n}$
	$?$		
$\mathcal{I} \otimes C$			
b			
	b		
			$\mathcal{E}_?$
$\mathcal{E}_? \otimes \mathcal{I}^i$			
m			
			m

There are three tensor cases to deal with. In the first case, we tensor two known qbits. Suppose that the strategies

$$\left[\!\!\left[\Gamma, \Delta_1, \overline{x}_1 \colon \mathrm{qbit}^{\otimes n} \vdash M_1 \colon \mathrm{qbit}^{\otimes n}\right]\!\!\right] \text{ and } \left[\!\!\left[\Gamma, \Delta_2, \overline{x}_2 \colon \mathrm{qbit}^{\otimes m} \vdash M_2 \colon \mathrm{qbit}^{\otimes m}\right]\!\!\right]$$

are already defined, where $\mathrm{FV}(M_i) \setminus |\Delta_i| = \emptyset$ for $i = 1, 2$. The strategy $[\![M_1 \otimes M_2]\!]$ is defined as the composition $r; [\![M_1]\!] \otimes [\![M_2]\!]$, where the strategy $[\![M_1]\!] \otimes [\![M_2]\!]$ is defined by the scheme described in figure 6.3. In this scheme, the probability that Player answers m to $\mathcal{E}_?$ after the interactions $s = a_1 \ldots a_n$ and $t = b_1 \ldots b_k$ is $\operatorname{tr}(\mathcal{E}_m(\rho_s \otimes \rho_t))$. Note that while we take the tensor product of the two output quantum arenas, we must take the classical game product of the classical input arenas.

Figure 6.3 Strategy for the first QDL tensor rule

$$\begin{array}{ccccc}
(\llbracket\Gamma\rrbracket \odot \llbracket\Delta_1\rrbracket) & \odot & (\llbracket\Gamma\rrbracket \odot \llbracket\Delta_2\rrbracket) & \xrightarrow{\llbracket M_1\rrbracket\otimes\llbracket M_2\rrbracket} & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \\
 & & & & \mathcal{E}_? \\
a_1 & & & & \\
\vdots & & & & \\
a_n & & & & \\
 & & b_1 & & \\
 & & \vdots & & \\
 & & b_k & & \\
 & & & & m
\end{array}$$

In the second case, we tensor two qbits each constructed from unknown qbits. This case is similar to the first one: suppose that $\left\llbracket \Gamma, \Delta_1 \vdash M_1 \colon \mathrm{qbit}^{\otimes n} \right\rrbracket$ and $\left\llbracket \Gamma, \Delta_2 \vdash M_2 \colon \mathrm{qbit}^{\otimes m} \right\rrbracket$ are already defined and that $\mathrm{FV}(M_i) \cap |\Delta_i| = \{\overline{x}_i\}$. The strategy $\llbracket M_1 \otimes M_2 \rrbracket$ is defined to be the composition $r \odot \mathrm{id}; \llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket$, but this time the strategy $\llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket$ must be defined using the scheme of figure 6.4. In this figure $\mathcal{F}_s$ and $\mathcal{G}_t$ are the two trace-preserving superoperators used by Player respectively in $\llbracket M_1 \rrbracket$ and $\llbracket M_2 \rrbracket$.

The third tensor rule is for cases where known and unknown states are tensored. In this case we have to use a conditional preparation strategy defined using a combination of schemes used in the first two cases. Assume that $\left\llbracket \Gamma, \Delta_1, \overline{x} \colon \mathrm{qbit}^{\otimes n} \vdash M_1 \colon \mathrm{qbit}^{\otimes n} \right\rrbracket$ and $\left\llbracket \Gamma, \Delta_2 \vdash M_2 \colon \mathrm{qbit}^{\otimes m} \right\rrbracket$ are already defined and that $\mathrm{FV}(M_1) \setminus |\Delta_1| = \{\overline{x}\}$ and $\mathrm{FV}(M_2) \cap |\Delta_2| = \emptyset$. The strategy $\llbracket M_1 \otimes M_2 \rrbracket$ is defined as the composition $r; \llbracket M_1 \rrbracket \otimes \llbracket M_1 \rrbracket$ where this time the tensor strategy $\llbracket M_1 \rrbracket \otimes \llbracket M_2 \rrbracket$ is defined with the scheme given in figure 6.5. Using that scheme, Player determines how to answer the initial question $\mathcal{E}_?$ by first playing in the $\llbracket\Gamma\rrbracket \odot \llbracket\Delta_2\rrbracket$ arena to determine which state ρ_s, $s = a_1 \ldots a_k$, to prepare; we assume this state is prepared by a superoperator $\mathcal{F}_s$. After this, Player will start an interaction in $\llbracket\Gamma\rrbracket$ in order to learn how the state represented by the term M_1 is built from its input. In this case, we assume that this construction corresponds to a superoperator $\mathcal{G}_t$, where $t = b_1 \ldots b_l$ is the interaction in the $\llbracket\Gamma\rrbracket$ part. The initial question is then transformed into the question $(\mathcal{F}_s \otimes \mathcal{G}_t)\,\mathcal{E}_?$ in the input arena $\mathbf{qbit}^{\otimes n}$, and the answer is copied back to the output arena.

This completes the definition of the denotational semantics.

Example 6.6. Consider the two QDL terms

$$M_1 = x \otimes (\mathsf{if}\, b\, \mathsf{then}\, \rho_1\, \mathsf{else}\, \rho_2)$$
$$M_2 = \mathsf{if}\, b\, \mathsf{then}\, x \otimes \rho_1\, \mathsf{else}\, x \otimes \rho_2,$$

Figure 6.4 Strategy for the second QDL tensor rule

$$\begin{array}{ccccccc}
(\llbracket\Gamma\rrbracket \odot \llbracket\Delta_1\rrbracket) & \odot & (\llbracket\Gamma\rrbracket \odot \llbracket\Delta_2\rrbracket) & \odot & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} & \xrightarrow{\llbracket M_1\rrbracket \otimes \llbracket M_2\rrbracket} & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \\
 & & & & & & \mathcal{E}_? \\
a_1 & & & & & & \\
\vdots & & & & & & \\
a_n & & & & & & \\
 & & b_1 & & & & \\
 & & \vdots & & & & \\
 & & b_m & & & & \\
 & & & & \mathcal{E}_?\,(\mathcal{F}_s \otimes \mathcal{G}_t) & & \\
 & & & & m & & \\
 & & & & & & m
\end{array}$$

Figure 6.5 Strategy for the third QDL tensor rule

$$\begin{array}{ccccccc}
(\llbracket\Gamma\rrbracket \odot \llbracket\Delta_1\rrbracket) & \odot & \mathbf{qbit}^{\otimes n} & \odot & (\llbracket\Gamma\rrbracket \odot \llbracket\Delta_2\rrbracket) & \xrightarrow{\llbracket M_1\rrbracket \otimes \llbracket M_2\rrbracket} & \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \\
 & & & & & & \mathcal{E}_? \\
 & & & & a_1 & & \\
 & & & & \vdots & & \\
 & & & & a_k & & \\
b_1 & & & & & & \\
\vdots & & & & & & \\
b_l & & & & & & \\
 & & \mathcal{E}_?\,(\mathcal{F}_s \otimes \mathcal{G}_t) & & & & \\
 & & m & & & & \\
 & & & & & & m
\end{array}$$

where ρ_1 and ρ_2 are two one qbit states. Intuitively, both terms produce the state $x \otimes \rho_1$ or $x \otimes \rho_2$ depending on the value of B. We can derive that

$$x\colon \text{qbit}, b\colon \text{bool} \vdash M_i\colon \text{qbit} \otimes \text{qbit}, \; i = 1, 2.$$

Let us compare the associated strategies $\llbracket M_1 \rrbracket$ and $\llbracket M_2 \rrbracket$.

In the first case, $\llbracket M_1 \rrbracket$ is defined as a preparation strategy with typical play

$$\begin{array}{ccccc}
\mathbf{qbit} & \odot & \mathbf{bool} & \multimap & \mathbf{qbit} \otimes \mathbf{qbit} \\
 & & & & \mathcal{E}_? \\
 & & ? & & \\
 & & b & & \\
\mathcal{E}_?\mathcal{F}_b & & & & \\
m & & & & \\
 & & & & m
\end{array}$$

where $\mathcal{F}_b$ is the superoperator that tensors its input with the state ρ_b.

In the second case, $\llbracket M_2 \rrbracket$ is the strategy in the same arena using which Player will first query for the boolean value in the **bool** input arena, then play according to either $\llbracket x \otimes \rho_1 \rrbracket$ or $\llbracket x \otimes \rho_2 \rrbracket$ conditionally on the given answer. A typical play is thus exactly the same as in the case of $\llbracket M_1 \rrbracket$ and thus $\llbracket M_1 \rrbracket = \llbracket M_2 \rrbracket$, as can be expected from the intuitive meaning of both terms.

6.4 Soundness

We now turn to the problem of proving a soundness result for the denotational semantics defined in the last section. First, we need a substitution lemma.

Lemma 6.7. *(Substitution for QDL) For any QDL terms $\Gamma, \Delta_1, \overline{x}\colon A \vdash M\colon B$ and $\Gamma, \Delta_2 \vdash N\colon A$ with $\overline{x} \in \mathrm{FV}(M)$, we have that*

$$\Gamma, \Delta_1, \Delta_2 \vdash M\,[N/\overline{x}] : B \quad \textit{and} \quad \llbracket M[N/\overline{x}] \rrbracket = r; \mathrm{id} \odot \llbracket N \rrbracket \, ; \llbracket M \rrbracket \, .$$

Proof. This is proven by structural induction on the construction of M. □

The following proposition states that when a term M reduces to some value V with probability p, the corresponding strategies $\llbracket M \rrbracket$ and $\llbracket V \rrbracket$ makes Player play in the same way with probability p.

Proposition 6.8. *If $M \Downarrow^p V$, then for all well-opened $sab \in \mathcal{T}(\llbracket V \rrbracket)$ we have that*

$$\llbracket M \rrbracket \, (b \mid sa) = p \, \llbracket V \rrbracket \, (b \mid sa).$$

Proof. By structural induction on the derivation of $M \Downarrow^p V$. Most of the proof follows an argument similar to the QSL case in section 5.3. We skip these to focus on the cases involving quantum operations.

For measurement operations, consider first the single qbit case. Suppose that $[\![M]\!]$ behaves as $[\![\rho]\!]$ with probability p. Assume that $\mathsf{meas}\, M$ reduces to 0 with probability $p\,\mathrm{tr}(|0\rangle\langle 0|\,\rho)$. The strategy $[\![\mathsf{meas}\, M]\!]$ is the composition $[\![M]\!]\,;\mathrm{meas}$ and, by induction hypothesis, any interaction using this strategy will behave as an interaction using the strategy $[\rho];\mathrm{meas}$. By definition of $[\rho]$, this strategy behaves as the constant strategy 0 in **bool** with probability $\mathrm{tr}\,(|0\rangle\langle 0|\,\rho)$, and thus $[\![\mathsf{meas}\, M]\!]$ behaves as $[\![0]\!]$ with probability $p\,\mathrm{tr}\,(|0\rangle\langle 0|\,\rho)$.

The general measurement case is similar.

To deal with the tensor operation reduction rule, suppose that the proposition holds when $M_1 \Downarrow^p V_1$ and $M_2 \Downarrow^q V_2$ and assume that

$$M_1 \otimes M_2 \Downarrow^{pq} V_1 \otimes V_2.$$

Since the definition of $[\![M_1 \otimes M_2]\!]$ is in three cases, these must be considered separately. In the first case, M_1 and M_1 are both terms with no free variables of type qbit appearing in the type context. By definition

$$[\![M_1 \otimes M_2]\!] = r \odot \mathrm{id}; [\![M_1]\!] \otimes [\![M_2]\!]$$

and by the induction hypothesis this will behaves as

$$[\![M_1 \otimes M_2]\!] = r \odot \mathrm{id}; [\![V_1]\!] \otimes [\![V_2]\!]$$

with probability pq. The other two cases are similar, except that the definition of the strategy $[\![M_1]\!] \otimes [\![M_2]\!]$ is different in each case. □

The next result is adequacy, the converse of the previous one. As for classical λ-calculus and QSL, we use a computability predicate to prove adequacy for QDL. The main difference between the following definition and the usual definition of computability is the use of extended variables. Note that neither the presence of extended variables or the linearity constraint on qbits have any significant impact on this definition.

Definition 6.9. *A QDL term M is **computable** if*

1. *M is closed with $M\colon A$ and $A = \mathrm{bool}$, $\top$ or qbit, and if for all $sab \in \mathcal{T}\,(\,b \mid sa)$ we have that $[\![M]\!]\,(\,b \mid sa\,) = p\,[\![V]\!]\,(\,b \mid sa\,)$, then $M \Downarrow^p V$,*

2. *$\overline{x_1}\colon A_1, \ldots, \overline{x_n}\colon A_n \vdash M\colon A$ and for all computable closed terms*

$$\Gamma \vdash N_1 : A_1,\ \ldots\ , \Gamma \vdash N_n : A_n$$

   we have that $M[N_1/\overline{x_1},\ \ldots\ , N_n/\overline{x_n}]$ is computable,

3. *M is closed with $\vdash M\colon A \Rightarrow B$ and for all closed N with $\vdash N\colon A$ the term MN is computable.*

Lemma 6.10. *All QDL terms are computable.*

We need the following lemma which is proved by induction on the construction of the type A.

Lemma 6.11. *For any type A, there exist a closed term M of type A.*

Proof of lemma 6.10. By induction on the construction of M. The part of the proof involving classical constructs follows the usual pattern as in classical game semantics, using lemma 6.11 for abstraction as explained in the proof of lemma 5.10, so we focus here on the quantum operations. Using the definition of computability, we can assume that the components of M are computable closed terms.

The most interesting case is measurement since it involves an argument specific to QDL. We begin with the one qbit measurement case. Suppose that $M = \mathsf{meas}\, N$ where N is a closed computable term of type qbit. Assume that V is a boolean value and that

$$[\![M]\!]\,(\,b \mid sa\,) = p\,[\![V]\!]\,(\,b \mid sa\,)$$

for all well-opened $sab \in \mathcal{T}([\![V]\!])$.

When Player uses $[\![M]\!]$, a typical play is

$$\begin{array}{ccccc} I & \overset{[\![N]\!]}{\multimap} & \mathbf{qbit} & \overset{\mathsf{meas}}{\multimap} & \mathbf{bool} \\ &&&& ? \\ && C_? && \\ && m && \\ &&&& m \end{array}$$

where $C_?$ is the quantum intervention corresponding to a projective measurement in the canonical basis. Let p be the probability that using $[\![N]\!]$ the answer is 0 and $1 - p$ the probability that the answer is 1. Although it is not possible to infer which state ρ is used to answer $C_?$ using these probabilities, we know that if player was using

$$\rho' = p|0\rangle\langle 0| + (1 - p)|1\rangle\langle 1|$$

instead of ρ, we would get the same play as above. Since $\mathsf{meas}\,\rho' \Downarrow^p 0$, we get that $\mathsf{meas}\,\rho \Downarrow^p 0$ as required.

We use a similar argument to deal with the general measurement case. For unitary operations, the above problem does not occur since the strategy $[\![\mathcal{U}M]\!] = [\![M]\!]\,;[\mathcal{U}]$ provides the measurement probabilities for all quantum interventions $\mathcal{E}_?$. This allows one to find, via the Gleason theorem, a state ρ such that $[\![M]\!]$ behaves like $[\rho]$ with probability p. Using this and the induction hypothesis on M, we get the desired result. □

Adequacy is a direct corollary of lemma 6.10.

Proposition 6.12. *(Adequacy for QDL) Let M be a closed term of type* bool, $\top$ *or* qbit$^{\otimes n}$. *If for all well-opened $sab \in \mathcal{T}([\![V]\!])$ we have that $[\![M]\!]\,(\,b \mid sa\,) = p\,[\![V]\!]\,(\,b \mid sa\,)$, then we have that $M \Downarrow^p V$.*

To give the final result, we need to introduce the necessary concept of contextual equivalence for QDL. A **context** $C[-]$ of type B with a hole of type A is a term $C[-]$ with a special variable "$-$" (possibly an extended variable) such that $- \colon A \vdash C[-] \colon B$. Capture-free substitution of a term N in a context $C[-]$ is denoted $C[N]$.

Definition 6.13. *Two closed terms $\vdash M_1 : A$ and $\vdash M_2 : A$ are **contextually equivalent** (denoted $M_1 \sim M_2$) if for every ground-type context $C[-]$ with a hole of type A we have that*

$$C[M_1] \Downarrow^p V \iff C[M_2] \Downarrow^p V.$$

The following soundness result follows from proposition 6.8 and adequacy using the same standard argument used to prove proposition 5.13 in the last chapter.

Proposition 6.14. *(Soundness for QDL) Let M_1 and M_2 by two closed QDL terms. If $[\![M_1]\!] = [\![M_2]\!]$, then $M_1 \sim M_2$.*

Chapter 7

Conclusion

7.1 Recapitulation

We introduced a notion of *quantum arena* and of *quantum strategy* derived from the concept of probabilistic strategy of Danos and Harmer [DH02] and based on the vision of quantum knowledge proposed by D'Hondt and Panangaden [DP05] and quantum consistent history theory. This notion was illustrated by many examples of quantum strategies that describe quantum states and important quantum operations. To justify the use of these strategies, a criterion was given to identify the probabilistic strategies that correspond to quantum states. Since the usual classical game semantics operations on arenas are insufficient to represent tensor product spaces adequately, we introduced a new tensor operation for quantum arenas. As a last contribution to the topic of quantum strategies, we gave an overview of the various generalisations of quantum plays that can be obtained by considering other kinds of quantum measurements than projective measurements. The two important cases are quantum plays using POVM measurements and those using intervention operators.

The rest of this thesis was devoted to the use of quantum strategies to analyse three different quantum programming languages. We first gave a denotational semantics for a typed variant of the measurement calculus of Danos et al. [DKP07]. We obtained a soundness result for this semantics.

We then introduced two new higher-order quantum programming languages. While the syntax of both languages was derived from the work of Selinger and Valiron [SV06a], two different views of the interaction between quantum and classical parts of a quantum λ-calculus were developed. In one case, quantum states are represented as states of quantum stores on which various commands can be applied. A new syntactic device, *extended variables*, was used to allow various qbits of a store to be entangled. In the other case, quantum states can be used directly in the language as quantum data, forcing the λ-calculus to be linear to avoid duplication of unknown states. In game semantics of classical languages, this difference between a reference and the data itself is reflected in the semantics as the difference between thread dependent and thread independent strategies. The work presented in this thesis clarifies the impact this has in the quantum case: measuring quantum states has side effects which can only be represented using thread dependent strategies.

The fact that there are two different products (quantum and classical) of quantum arenas led us to separate the $\otimes$ type operation of the quantum λ-calculus of Selinger and Valiron into the classical arena product operation and the quantum tensor product arenas operation.

A denotational semantics using quantum arenas was given for both the quantum store and the quantum data λ-calculi. In both cases the classical segment of the interpretation uses known constructs from game semantics. For the quantum store λ-calculus, new quantum arenas and strategies were required to take into account the fact that the internal state of a quantum store is affected by unitary transformation, measurements and preparation commands. For the quantum data λ-calculus, qbit variables can only be used linearly because the semantics requires thread dependent strategies to account for quantum measurements side effects. We proved soundness results for both languages.

7.2 Discussion

The main goal of this thesis was to present quantum games and strategies as a new framework to understand the relation between classical and quantum data in quantum programming languages. The applications we have given show that it is possible to use this framework to define semantics of various typed quantum programming languages including higher order languages. It inherits one important general feature of game semantics: it can be adapted to deal with different kinds of quantum programming languages constructions. Let us point out some features of quantum strategies that had to be taken into account. These features played an important role in this thesis, as we took them as guides for the design of the two λ-calculi introduced instead of seeing them as defects of the model.

I Quantum strategies $[\rho]$ in $[H]$ are not thread independent. This is pointing out that quantum strategies behave like classical strategies for constructions with side-effects. This feature was obviously important for the λ-calculus with quantum stores, since thread dependence is a general feature of stores. It is also important in the case of the λ-calculus with quantum data since it entails that quantum data must be used linearly: a strategy representing a state ρ can't be duplicated using a Δ strategy as any interaction with it will change the state it represent.

II The usual game semantics tensor operation $\odot$ cannot produce quantum arenas where general quantum measurements can be made, thus making it impossible to deal with entanglement. This forced the introduction of a tensor operation $\otimes$ which can only be used on quantum strategies. The syntax of both λ-calculi we introduced reflect this: we used classical pairing and a purely quantum tensor operation. Having two different products also has consequences for abstraction because it depends on the existence of a closed structure, i.e. on the existence of an adjoint to the product. The quantum tensor product of arenas does not have an adjoint, and this makes it impossible to use λ-abstraction over a qbit variable which is part of a tensor product. There is, thus, no such abstraction in the syntax. This can be seen as a consequence of the principle we adopted in section 3.1.1: all choices are classical, and thus we cannot abstract over part of a tensor product.

III The quantum tensor product of strategies can be defined for either two strategies representing known states, or two strategies representing unknown states. To deal with the case of the tensor product of a known and an unknown state, we used instead a preparation strategy. These three cases are distinguished in the type system of the λ-calculus with quantum data. This feature of the quantum tensor product suggests that in quantum languages we must distinguish between the cases where quantum states are known and those where quantum states are unknown.

IV There is a strategy that allows one to consider locally the components of a state on a joint space as independent states. This strategy

$$C\colon \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes m} \multimap \mathbf{qbit}^{\otimes n} \odot \mathbf{qbit}^{\otimes m}$$

is used in the interpretation of the $\mathsf{let} \dots \mathsf{in} \dots$ operations of the λ-calculus with quantum data. There is no strategy doing the reverse operation that takes independent states to tensor product states. This is intuitively impossible because there are ways to operate on the resulting tensor state that cannot be described as separate operation on each independent qbits.

These features of quantum arenas and strategies have their roots in the approach taken to define quantum strategies for quantum states. We adopted a point of view close to that of the quantum consistent histories interpretation of quantum mechanics: agents can only interact with quantum data through measurements. This determined the structure of quantum arenas where quantum states can be represented using the standard approach of game semantics to represent states of systems. We then built more complex quantum arenas using the usual product and arrow arena operations of game semantics. As the applications presented in this thesis showed, it is possible to represent enough important quantum operations in these arenas to be able to construct denotational semantics for quantum programming languages. The various properties of quantum strategies representing quantum operations are due to a feature specific to them. A typical strategy represents a classical operation in the arena $A \multimap B$ as the way Player uses Opponent's answers in the input A to give an answer in the output B. In contrast, in all the examples of quantum strategies given in this thesis, a quantum operation is represented as the relation between the initial question $\mathcal{P}_{?_B}$ asked by Opponent with the counter-question $\mathcal{P}_{?_A}$ asked by Player. This constraint explains the last of the three features listed above.

7.3 Future work

We conclude this thesis with possible developments of the ideas it presents.

While we gave enough results on quantum strategies to be able to define denotational semantics for three quantum languages, there are many questions remaining to be answered. A central one is to characterise, using a condition on plays, the quantum strategies in $[H_A] \multimap [H_B]$ among all probabilistic strategies in that arena. In chapter 3 we defined the quantum strategies as those that send, via composition, quantum states to quantum states. We would like to identify these strategies directly, since strategies with that property correspond to superoperators. The link with consistent histories may prove useful to solve this problem. By contrast, the approach used to define quantum arenas could be used in consistent history theory to describe processes; as far as the author is aware, there is no such development in that theory. Note that the characterisation of the quantum strategies of the form $[\rho]$ we gave in chapter 3 relies on Gleason's characterisation of density operators in terms of probabilities assigned to projectors. To get a similar result for quantum strategies describing quantum operations, we would need a result characterising a quantum operation $\mathcal{E}$ as a function taking quantum measurements on the output to quantum measurements on the input. The author is not aware of any such result either. As explained in the conclusion of chapter 4, the absence of such a characterisation explains why we did not give any full abstraction results.

A closely related problem is to understand better the structure of the category **Qstrat** of quantum strategies. This category was defined as a first step toward the construction of a dagger compact-closed category of quantum arenas and strategy. Its relation with the larger category of probabilistic strategies should be investigated further. This categorical investigation should probably take into account the features of quantum strategies enumerated in the last section. One possible goal for such an investigation would be to get a factorisation result which would allow one to split a probabilistic strategy into a quantum and a classical strategy. Factorisation results are used in games semantics as a way to reduce a full abstraction proof for a given language to a full abstraction result for a simpler language.

Another possible research development is to improve our understanding of the structure of quantum strategies extended to use intervention operators and the arena **qstore**, as described in chapter 3. The author proposed an alternative formalism to describe strategies which use a structure related to Petri-nets to describe information flow in classical game semantics [Del05]. This information flow framework can be adapted to the use of quantum interventions.

Finally, the concepts of quantum arena and strategy presented in this thesis could be used to analyse quantum protocols used in quantum information and cryptography theory. Quite often these protocols are already presented informally as games and furthermore some work has been done to use game semantics tools in classical cryptography [Jür05]. The relation between our approach and the results mentioned on non-locality without entanglement – results that have antecedents in the literature on quantum key distribution – is another indication that the quantitative approach of quantum information theory could be used to analyse quantum strategies.

Bibliography

[AC03] S. Abramsky and B. Coecke. Physical traces: Quantum vs. classical information processing. In R. Blute and P. Selinger, editors, *Electronic Notes in Theoretical Computer Science*, volume 69. Elsevier, 2003.

[AC04] S. Abramsky and B. Coecke. A categorical semantics of quantum protocol. In *Proceedings of the 19th IEEE conference on Logic in Computer Science: LICS 2004*, pages 415–425. IEEE Computer Society, 2004.

[AD06] S. Abramsky and R. Duncan. A categorical quantum logic. *Mathematical Structures in Computer Science*, 16:469–489, 2006.

[AHM98] S. Abramsky, K. Honda, and G. McCusker. A fully abstract game semantics for general references. In *Proceedings of the Thirteenth International Symposium on Logic in Computer Science*, pages 334–344. Computer Society Press of the IEEE, 1998.

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

[AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (extended abstract). In M. Hagiya and J. C. Mitchell, editors, *Theoretical Aspects of Computer Software*, volume 789 of *Lecture notes in computer sciences*, pages 1–15. Springer, 1994.

[AL04] P. Aliferis and D. W. Leung. Computation by measurements: a unifying picture, 2004.

[AM99] S. Abramsky and G. McCusker. Game semantics. In H. Schwichtenberg and U. Berger, editors, *Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School*, pages 1–56. Springer, 1999.

[AMJ00] S. Abramsky, P. Malacaria, and R. Jagadeesan. Full abstraction for PCF. *Information and Computation*, 163:409–470, 2000.

[BBC$^+$93] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70:1895–1899, 1993.

[BDF$^+$99] C. H. Bennett, D. P. DiVincenzo, C. A. Fuchs, T. Mor, E. Rains, P. W. Shor, J. A. Smolin, and W. K. Wootters. Quantum nonlocality without entanglement. *Physical Review A*, 59(2):1070–1091, 1999.

[Bla92] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:151–166, 1992.

[BW99] M. Barr and C. Wells. *Category Theory for Computing Science*. Centre de recherche en mathématiques, third edition, 1999.

[Cho75] M. Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, pages 285–290, 1975.

[CHTW04] R. Cleve, P. Hoyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th Annual IEEE Conference on Computational Complexity (2004)*, pages 236–249, 2004.

[Coe04] B. Coecke. The logic of entanglement. an invitation. Technical report, Oxford University Computing Laboratory, 2004.

[Coe07] B. Coecke. De-linearizing linearity: projective quantum axiomatics from strong compact closure. In Peter Selinger, editor, *Proceedings of the 3rd International Workshop on Quantum Programming Languages*, volume 170 of *Electronic Notes in Theoretical Computer Science*. Springer, 2007.

[CP06a] B. Coecke and É. O. Paquette. POVMs and naimark's theorem without sums. In *Proceedings of the 4th International Workshop on Quantum Programming Languages (QPL'06)*, 2006.

[CP06b] B. Coecke and D. Pavlovic. Quantum measurements without sums, 2006.

[Del05] Y. Delbecque. Information and information flow in game semantics. In Ghica and McCusker [GM05], pages 226–240.

[Del08a] Y. Delbecque. Game semantics for quantum data. In *Proceedings of the joint 5th Quantum Physics and Logic and 4th Development of Computational Models Workshops*, Electronic Notes in Theoretical Computer Science, 2008.

[Del08b] Y. Delbecque. A quantum game semantics for the measurement calculus. In *Prooceedings of the fourth International Workshop on Quantum Programming Languages*, volume 210, pages 33–48. Elsevier, 2008.

[Deu85] D. Deutsch. Quantum theory, the church-turing principle, and the universal quantum computer. In *Proceedings of the Royal Society of London*, volume A400, pages 97–117. Royal Society, 1985.

[DH02] V. Danos and R. Harmer. Probabilistic game semantics. In *ACM Transactions On Computational Logic, Special Issue for LICS'00*. Association For Computing Machinery, ACM Press, 2002.

[D'H05] E. D'Hondt. *Distributed quantum computation - a measurement-based approach*. PhD thesis, Vrije Universiteit Brussel, 2005.

[DJ92] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. In *Proceedings of the Royal Society of London A*, number 439 in A. Royal Society, 1992.

[DKP07] V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *Journal of the Association of Computing Machinery*, 2007.

[DL70] E. B. Davies and J. T. Lewis. An operational approach to quantum probability. *Communications in Mathematical Physics*, 17(3):239–260, 1970.

[DP05] E. D'Hondt and P. Panangaden. Reasoning about quantum knowledge. In R. Ramanujam and S. Sen, editors, *Foundations of Software Technology and Theoretical Computer Science*, number 3821 in Lecture Notes in Computer Science, pages 553–564. Springer, 2005.

[DP08] Y. Delbecque and P. Panangaden. Game semantics for quantum stores. In *Proceedings of the XXIV Symposium on Mathematical Foundations of Programming Semantics*, Electronic Notes in Theoretical Computer Science, 2008.

[EWL99] J. Eisert, M. Wilkens, and M. Lewenstein. Quantum games and quantum strategies. *Physical Reviews Letters*, 83(15):3077–3080, 1999.

[Gay05] S. Gay. Quantum programming languages: Survey and bibliography. *Bulletin of the European Association for Theoretical computer science*, 2005.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gle57] A. M. Gleason. Measures on the closed subspaces of a hilbert space. *Journal of Mathematics and Mechanics*, 6:885–893, 1957.

[GM05] D. R. Ghica and G. McCusker, editors. *Games for Logic and Programming Languages (GALOP 2005), University of Edinburgh, 2-3 April 2005,*, 2005.

[GMH93] M. Gell-Mann and J.B. Hartle. Classical equations for quantum systems. *Physical Review D*, 47:3345–3382, 1993.

[Gri84] R. B. Griffiths. Consistent histories and the interpretation of quantum mechanics. *Journal of Statistical Physics*, 36(219), 1984.

[Gri03] R. B. Griffiths. *Consistent Quantum Theory*. Cambridge University Press, 2003.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceeding of the 28th Annual Symposium on Theory of Computing*, volume 69, 1996.

[Gun92] C. A. Gunter. *Semantics of programming languages*. MIT Press, 1992.

[GW07] G. Gutoski and J. Watrous. Toward a general theory of quantum games. In Jon M. Kleinberg, editor, *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 565–574. ACM, 2007.

[Har99] R. Harmer. *Games and Full Abstraction For Nondeterministic Languages*. PhD thesis, Imperial College, 1999.

[HO00] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I. models, observables and the full abstraction problem, II. dialogue games and innocent strategies, III. a fully abstract and universal game model. *Information and Computation*, 163:285–408, 2000.

[Hol73] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973.

[Jac94] B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69:73–106, 1994.

[Jür05] J. Jürjens. Towards using game semantics for crypto protocol verification: Lorenzen games. In Ghica and McCusker [GM05], pages 241–257.

[Kni96] E. Knill. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

[Kra83] K. Kraus. *States, Effects and Operations: Fundamental Notions of Quantum Theory*. Number 190 in Lecture notes in physics. Springer, 1983.

[Löw34] K. Löwner. Uber monotone matrixfunktionen. *Mathematische Zeitschrift*, 38:177–216, 1934.

[LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[Mac71] S. MacLane. *Categories for the Working Mathematician*. Springer, 1971.

[May96] P. Maymin. Extending the lambda calculus to express randomized and quantumized algorithms. arXiv:quant-ph/9612052, 1996.

[May97] P. Maymin. The lambda-q calculus can efficiently simulate quantum computers. arXiv:quant-ph/9702057, 1997.

[Mey99] D. A. Meyer. Quantum strategies. *Physical Review Letters*, 82:1052–1055, 1999.

[Mey00] D. A. Meyer. Why quantum strategies are quantum mechanical. *Physical Review Letters*, 84, 2000.

[MvN47] O. Morgenstern and J. von Neumann. *The Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Neu43] M. A. Neumark. On a representation of additive operator set functions. *Doklady Akademii Nauk SSSR*, 41:359–361, 1943.

[Omn88a] R. Omnès. Logical reformulation of quantum mechanics. I. Foundations. *Journal of Statistical Physics*, 53(1-2):893–932, 1988.

[Omn88b] R. Omnès. Logical reformulation of quantum mechanics. II. Interferences and the Einstein-Podolsky-Rosen experiment. *Journal of Statistical Physics*, 53(1-2):933–955, 1988.

[Omn88c] R. Omnès. Logical reformulation of quantum mechanics. III. Classical limit and irreversibility. *Journal of Statistical Physics*, 53:957–975, 1988.

[Omn88d] R. Omnès. Logical reformulation of quantum mechanics. IV. Projectors in semiclassical physics. *Journal of Statistical Physics*, 57(1–2):357–382, 1988.

[Omn92] R. Omnès. Consistent interpretations of quantum mechanics. *Reviews of Modern Physics*, 64(2):339–382, 1992.

[OR94] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT press, 1994.

[Per00] A. Peres. Classical interventions in quantum systems. I. The measuring process. *Physical Review A*, 61, 2000.

[RB01] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letter A*, 86(22):5188–5191, 2001.

[Rey81] J. C. Reynolds. The essence of ALGOL. In J. W. de Bakker and J. C. van Vliet, editors, *Algorithmic Languages, Proceedings of the International Symposium on Algorithmic Languages*, pages 345–372. North-Holland, 1981.

[Sel04a] P. Selinger. A brief survey of quantum programming languages. In *Proceedings of the 7th International Symposium on Functional and Logic Programming, Nara, Japan.*, number 2998 in Lecture Notes in Computer Science, pages 1–6. Springer, 2004.

[Sel04b] P. Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14:527–586, 2004.

[Sel04c] P. Selinger. Towards a semantics for higher-order quantum computation. *Proceedings of the 2nd International Workshop On Quantum Programming Languages, Turku, Finland*, pages 127–143, 2004.

[Sel06] P. Selinger. Idempotents in dagger categories. In *Proceedings of the 4th International Workshop on Quantum Programming Languages*, 2006.

[Sel07] P. Selinger. Dagger compact closed categories and completely positive maps. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages*, number 170 in Electronic Notes in Theoretical Computer Science, pages 139–163, 2007.

[Sho94] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceeding of the 35th Annual Symposium on Foundation of Computer Science*, volume 69, pages 20–22, 1994.

[SV06a] P. Selinger and B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006.

[SV06b] P. Selinger and B. Valiron. On a fully abstract model for a quantum linear functional language. In *Proceedings of the 4th International Workshop on Quantum Programming Languages, Oxford, July 17-19*, 2006.

[SV08] P. Selinger and B. Valiron. A linear-non-linear model for a computational call-by-value lambda calculus. In *Proceedings of the Eleventh International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2008), Budapest*, number 4962 in Lectures Notes in Computer Sciences, pages 81–96. Springer, 2008.

[Val04] B. Valiron. A functional programming language for quantum computation with classical control. Master's thesis, Departement of Mathematics, University of Ottawa, 2004.

[vdMP03] R. van der Meyden and M. Patra. Knowledge in quantum systems. In *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pages 104–117. ACM press, 2003.

[vT04] A. van Tonder. A lambda calculus for quantum computation. *Siam Journal on Computing*, 33(5):1109–1135, 2004.