Puthoor, Ittoop Vergheese (2015) *Theory and applications of quantum process calculus.* PhD thesis.

http://theses.gla.ac.uk/5986/

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Glasgow Theses Service
http://theses.gla.ac.uk/
theses@gla.ac.uk

Theory and applications of quantum process calculus

Ittoop Vergheese Puthoor

A thesis submitted in partial fulfillment for the degree of

Doctor of Philosophy

Supervisors: Dr. Simon Gay and Dr. Sonja Franke-Arnold

School of Computing Science
University of Glasgow
Glasgow
United Kingdom

Wednesday 28th January, 2015

Declaration of Authorship

I, Ittoop Vergheese Puthoor, declare that this thesis titled, 'Theory and applications of quantum process calculus' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"But he said to me, My grace is sufficient for you, for my power is made perfect in weakness. Therefore I will boast all the more gladly about my weaknesses, so that Christ's power may rest on me."

The Holy Bible (2 Corinthians 12:9)

Abstract

Formal methods is an area in theoretical computer science that provides the theories and tools for describing and verifying the correctness of computing systems. Usually, such systems comprise of concurrent and communicating components. The success of this field led to the development of quantum formal methods by transferring the ideas of *formal methods* to quantum systems. In particular, *formal methods* provides a systematic methodology for verification of systems. *Quantum process calculus* is a specialised field in quantum formal methods that helps to describe and analyse the behaviour of systems that combine quantum and classical elements.

We focus on the theory and applications of quantum process calculus in particular to use *Communicating Quantum Processes (CQP)*, a quantum process calculus, to model and analyse quantum information processing (QIP) systems. Previous work on CQP defined labelled transition relations for CQP in order to describe external interactions and also established the theory of behavioural equivalence in CQP based on probabilistic branching bisimilarity. This theory formalizes the idea of observational indistinguishability in order to prove or verify the correctness of a system, and an important property of the equivalence is the *congruence* property. We use the theory to analyse two versions of a quantum error correcting code system. We use the equational theory of CQP from the previous work and define an additional three new axioms in order to analyse quantum protocols comprising quantum secret-sharing, quantum error correction, remote-CNOT and superdense coding.

We have expanded the framework of modelling in CQP from providing an abstract view of the quantum system to describe a realistic QIP system such as linear optical quantum computing (LOQC) and its associated experimental processes. By extending the theory of behavioural equivalence of CQP, we have formally verified two models of an LOQC CNOT gate using CQP. The two models use different measurement semantics in order to work at different levels of abstraction. This flexibility of the process calculus approach allows descriptions from detailed hardware implementations up to more abstract specifications.

The orbital angular momentum (OAM) property of light allows us to perform experiments in studying higher dimensional quantum systems and their applications to quantum technologies. In relation to this work, we have extended CQP to model higher dimensional quantum protocols.

Acknowledgements

I am indebted to so many people for their help and support, without which I could not complete my research. Firstly, I would like to thank my supervisors, Dr. Simon Gay and Dr. Sonja Franke-Arnold, for all that you have taught me and for the opportunities that you have provided me. Simon, introduced me to the intriguing field of quantum formal methods and quantum computing and the challenging problems in this new perplexing area. When I began my research, I never had any background in Computer Science but Simon was very patient with my slow progress and helped me to overcome various obstacles during my PhD research. Sonja provided me all the invaluable help and guidance not only during my research but also before the beginning of my work. I am very thankful and consider myself as very lucky to have Simon and Sonja for not only being my mentors but also providing an excellent environment to carry out my research.

I would also like to thank the Lord Kelvin Adam Smith (LKAS) Scholarship of the University of Glasgow for providing me the funding to do my PhD. The LKAS scheme not only provided me the best facilities to do my research but also provided me the funds to present my research at various conferences.

Special thanks to Prof. Scott Walck with whom I have had many discussions in both Computer Science and Physics. His suggestions and ideas provided a lot of encouragement during my research.

I would also like to thank Dr. Oana Andrei for her valuable suggestions and comments which has been crucial in writing this thesis. I have benefited from useful discussions, both technical and non technical, with Dr. Tim Davidson, Dr. Ebrahim Ardeshir-Larijani, Dr. Ryan Kirwan, Ms. Sarah Sharp and the optics group.

I am grateful to my examiners Dr. Elham Kashefi and Dr. Sarah Croke for their invaluable comments and suggestions that significantly improved my thesis.

I would also like to thank Dr. Lakshminarayan Nariangadu, my teacher for being the inspiration to pursue my PhD.

I wish to thank my family and friends, for their continual love and support, and prayers, which manifests itself in many different ways.

My beloved wife Anna, who has stood by my side every step of the way, thank you for everything.

And finally, to my baby daughter Elsa, who provided me joy and cheers all the time of my research.

Contents

List of Figures

List of Tables

Abbreviations

CCS	**C**alculus of **C**ommunicating **S**ystems.
CSP	**C**ommunicating **S**equential **P**rocesses.
CQP	**C**ommunicating **Q**uantum **P**rocesses.
LTS	**L**abelled **T**ransition **S**ystem.
qCCS	**Q**uantum **CCS**.
QPAlg	**Q**uantum **P**rocess **A**lgebra.
QRAM	**Q**uantum **R**andom **A**ccess **M**achine.
QMC	**Q**uantum **M**odel **C**hecker.
QCTL	**Q**uantum **C**omputation **T**ree **L**ogic.
QKD	**Q**uantum **K**ey **D**istribution.
QIP	**Q**uantum **I**nformation **P**rocessing.
CWB-NC	**C**oncurrency **W**ork **B**ench - **N**ew **C**entury.
QSS	**Q**uantum **S**ecret **S**haring.
QECC	**Q**uantum **E**rror **C**ode **C**orrection.
LOQC	**L**inear **O**ptical **Q**uantum **C**omputing.
OAM	**O**rbital **A**ngular **M**omentum.
KLM	**K**nill **L**aflamme **M**ilburn.
EPR	**E**instein **P**odolsky **R**osen.
EDP	**E**ntanglement **D**istillation **P**rotocol.
QFT	**Q**uantum **F**ourier **T**ransform.

Dedicated to my family

Chapter 1

Introduction

As predicted by Moore [127] in 1965, computers have become faster and more powerful and, in the same time, decreased in size. This is explained by the exponential growth in the number of transistors on a microprocessor while the size of the processor remains constant. Keyes [99] extrapolates that the constant decrease in size of the computer circuits will reach the atomic level at a stage where a bit will be represented by a single atom in the year 2020: at this point *quantum effects* begin to play. This is an important reason for the significant research advancement in quantum computing, which is believed to be the next computing revolution.

A quantum computer is a computation device that uses certain quantum mechanical properties such as *superposition* and *entanglement* to perform computations on data. The technology promises to offer a very high degree of improvement over its classical counterpart. Some of the *potential* improvements provided by quantum computing over classical computing are:

- A quantum computer is defined in terms of fundamental microscopic systems which can be implemented by using the smallest known states.
- Quantum algorithms are much more efficient and outperform the classical algorithms for very specific tasks. For instance, Shor's algorithm [154] for prime factorisation, Grover's algorithm [83] for searching unstructured databases and the quantum Fourier transform [132] for performing Fourier transforms provide a significant improvement in complexity than the best known classical algorithms performing the same tasks.
- Quantum cryptography has already provided secure communication systems. The key distribution network (i.e. the process by which two or more users agree on a

shared secret, referred to as the key) has been tested [64, 143]. Quantum cryptographic systems are already commercially available from several companies like MagiQ Technologies [116], ID Quantique [93], NEC, Toshiba and so on. Protocols for quantum key distribution, such as BB84 [23] offer unconditional security, a result which is not yet achieved in classical computation.

Thanks to the above potential factors, quantum computing is already an advanced field of research involving computer science, physics, mathematics, chemistry and engineering, with the eventual aim of making a computer that works on the principles of quantum mechanics. D-Wave Systems [43] claimed to have built the first commercial quantum device, i.e quantum annealer, although it is yet to perform Shor's Algorithm. Recently, the scientists of D-Wave showed that their computer exhibits the quantum phenomenon called *entanglement* [111].

In the context of hardware and software technologies, formal verification is the method of proving the correctness of software programs or algorithms underlying a system. These techniques play a major role in proving the correctness of systems such as cryptographic protocols, digital circuits, and provide us an in depth understanding of interactive and complicated distributed systems. This approach has been successful in verifying mission critical or safety critical software and is considered as an alternative to testing. SPARK a formally defined computer language based on the Ada programming language is used in the systems that are safety and security related [15].

With the emergence of automated formal verification techniques over the past few years, the field is promising and has several industrial applications, including microprocessor design, automated business processes. The French railway company (SNCS) uses the B-Software [13], an automated verification tool, for modelling and verifying the automatic train protection system. Peugeot automobiles uses the B-System [13] for formally modelling the functioning of subsystems such as lightings, airbags, engine, and so on for their after sales service.

The aim of this thesis is to describe the theory and applications of the formal techniques for modelling and verifying quantum information processing systems. This chapter provides the context by giving a brief account of the research field; the motivation for our work and the contribution are described. Finally, the contents of the remaining chapters are outlined.

1.1 Context

In this section, we review some of the important aspects of quantum information and quantum computation.

1.1.1 Quantum Information

Information in general is referred to as the data that is contained in a *physical system*. Classical information theory provides the mathematical foundation for the storage, transmission and processing of information. Quantum information theory is the study of the same tasks using quantum mechanical systems.

In quantum mechanics, the state of the system is described by a *wave function*. The following six properties are the most important properties of quantum states for differentiating the quantum information from classical information [164]:

- Superposition
- Non-determinism
- Interference
- Uncertainty
- Non-cloneability
- Entanglement

Superposition. The *superposition* [30, 41] principle states that if a quantum particle can be in one of several given states, then it can also be in a state that is a linear combination of any two or more allowable states. This principle is due to the linearity of the quantum theory and is fundamental in distinguishing qubits (i.e. quantum bits) from classical bits, which can only ever be in one of the two states 0 and 1, but not in both. This principle gives rise to the notion that a particle can exist in one location and another at the same time. There are different interpretations of the meaning of this principle, but we will be concentrating on a few in the later part of this thesis. For instance, a photon has the intrinsic property of *polarisation*, which can either be *horizontal (H)*, *vertical (V)* or a superposition of both.

Non-determinism. To extract the information from a quantum system, one has to perform a *measurement*. The quantum theory is *non-deterministic* as the outcome of a measurement is not predictable. For example, if an observer measures the polarisation of a photon, which is assumed to be in a superposition of H and V. Then, the measurement causes the photon's polarisation to collapse, at random, to either the H or V with certain probabilities. This important *non-deterministic* aspect of quantum theory is in sharp contrast with the deterministic classical theory more often predicted by the Newtonian laws of classical physics.

Interference. This is a feature which is exhibited by the wave-like behaviour of the particle. According to classical theory, constructive interference occurs when the crest of one wave joins with the crest of the other to produce a much stronger wave, while destructive interference occurs when the crest of one meets with the trough of the other, resulting in nothing only when both wave amplitudes are identical. Also, another widely known fact is that the quantum system can show not only wave-like behaviour but also particle-like behaviour, referred to as the *wave-particle duality* [41].

Uncertainty. The common example for this property occurs in quantum theory for a single particle. The uncertainty principle states that it is impossible to know precisely both the position and momentum of a quantum particle. Many quantum protocols rely on this property. For example, BB84 [23] uses the uncertainty principle and statistical analysis to determine the presence of an eavesdropper on a quantum communication channel by encoding the information into two complementary variables.

Non-cloneability. The *no-cloning* theorem [166] states that an unknown quantum state cannot be cloned or copied. This property is used in many quantum cryptographic protocols and we illustrate it in Chapter 4.

Entanglement. The last and most striking quantum feature that has no classical analog is called the *entanglement*. This refers to the strong quantum correlations that two or more quantum particles can possess and is used in most communication protocols. The direct interaction between these quantum particles that are separated in space with no intermediate mechanism between them is also referred to as the *quantum non locality*. For example, entanglement is a resource which is used in the teleportation protocol [24] that teleports a quantum state from one location to another. More examples of exploiting this feature is seen in the coming chapters of this thesis.

Noise is a common feature which is visible in both classical and quantum information. In quantum information theory the noise is referred to as *decoherence* [132]. The stability of a quantum state tends to reduce when the quantum particle interacts with its environment and this phenomenon is also known as *decoherence*. This phenomenon is an obstacle to carry out computation and communication. The development of quantum error correction techniques (discussed in Chapters 2, 4 and 5) helps to overcome this problem. Quantum systems such as photons are less susceptible to decoherence than other systems. This is the main reason which makes photons one of the most prominent candidates for implementing quantum computing.

The above six phenomena capture the essence of the quantum theory. More aspects of it will be clearly seen as we progress in the later chapters of the thesis.

1.1.2 Quantum Computation

The concept of using quantum physics for computation was first introduced by Yuri Manin [117] in 1980 and then by Richard Feynman [67] in 1982. Feynman also described the difficulties of simulating quantum mechanical systems on classical computers. In 1985, Deutsch [54] proposed an abstract machine that could be used to model the effect of a quantum computer. This device, called the *universal quantum computer* or the *quantum Turing machine*, is meant to be a simple model which has all the power of quantum computation.

In analogy with a classical algorithm, a quantum algorithm is a sequence of steps that can be performed on a quantum computer. Deutsch and Josza [55] proposed one of the first examples of a quantum algorithm that is exponentially faster than any other classical algorithm. It is a deterministic algorithm that always provides the correct answer and demonstrates that there exists, a certain class of problems outside the complexity class **P** which could be solved in polynomial time on a quantum computer.

This generated a widespread interest and led to the development of the most well known algorithms of Shor [154] and Grover [83]. Shor's algorithm runs exponentially faster for prime factoring than any known classical algorithm. According to [84], the best known classical algorithm (the *quadratic sieve*) for n bit number factorisation runs in time $O(exp(c.log(n)^{\frac{1}{3}}\ loglog(n)^{\frac{2}{3}}))$ for $c = (\frac{64}{9})^{\frac{1}{3}}$ while Shor's algorithm requires $O(n^2 log(n).log(log(n)))$ steps on a quantum computer. This is very important, as the most popular public key system such as RSA [103], based on the one-way character of multiplying two large prime numbers (typical over 200 decimal digits), assumes that

there is no polynomial time factorisation algorithm. A quantum computer implementing Shor's algorithm may be the only possibility which could break the security of such systems.

In 1996, Grover [83] proposed a quantum algorithm for the efficient search of unordered lists. The classical algorithm for this problem has a complexity of $O(N)$ for a list of N elements while the Grover's algorithm takes $O(\sqrt{N})$ computations. Grover's algorithm doesn't have exponential speed up, but it clearly shows to solve certain computation problems with high efficiency.

The above quantum algorithms are designed to run on a quantum computer. In recent years, there has been an intense research in developing suitable architectures that can perform the role. Some of the leading candidates are: atoms in optical cavities [122], trapped ions [100], superconducting charge [130], nuclear magnetic resonance in molecules [162], spin and charge based quantum dots [104], trapped electrons and single photons [107].

Optical quantum computing has the advantage of exhibiting less decoherence when it comes to transmission of information over large distances which is needed for quantum key distribution (QKD) [22, 23]. Also with the fact that the present infrastructure on communication is based on fibre and integrated optics, it provides an experimental advantage as the components are readily available. Quantum cryptographic systems are now commercially available [93, 116] and there has been a significant progress in demonstrating several quantum protocols experimentally.

In this thesis, we concentrate on a certain optical implementation of system for QIP, and photons naturally allow to integrate quantum computation and quantum communication. Photons can easily be generated, manipulated and detected. The fact that they possess large coherence times makes them the excellent candidates for computation and communications. Linear optical quantum computing (LOQC) is an optical implementation of small-scale quantum computing [106]. We will be discussing in detail about LOQC in the later part of the thesis (Chapter 6 and Chapter 7) as we work to model and verify LOQC, using the mathematical tools and techniques of theoretical computer science.

1.1.3 Quantum Communication and cryptography

The idea of quantum cryptography was first introduced by Wiesner [163] and later on further developed by Bennett and Brassard [23] which resulted in the BB84 quantum key distribution (QKD) protocol. QKD exploits the properties of quantum physics to provide

a secure communication. This allows two parties to create a shared random key that is known only to them. This key is used to encrypt and decrypt the messages. In contras to the *bits* of classical communication, quantum communication involves encoding of information in quantum states or *qubits*. Mostly, photons are used for these quantum states. The BB84 protocol may be realised with photon polarisaton, ionic quantum levels or any other quantised 2 level system. Assuming the quantum information is encoded using the *polarisation* of photons. The qubits are encoded either in the $H-V$ basis or the diagonal basis. The presence of the eavesdropper can be detected if he/she measures the information in the wrong basis which generates random errors that are communicated to the parties. This was later implemented experimentally over a certain distance (32cm) [22].

EPR protocol by Ekert [62] and the B92 protocol by Bennet *et al.* [21] are other examples for QKD. The EPR protocol uses an entangled pair of qubits from a third party source and Bell's theorem [20] to detect the presence of an eavesdropper while in B92 the Bell's theorem is not needed. Examples of other cryptographic protocols include *bit commitment* [32], *coin flipping* [23, 27] and secret sharing [88, 118].

Several applications of quantum communication like *quantum teleportation* [24], *superdense coding* [24] and quantum gate teleportation [79] have been demonstrated. Quantum teleportation is a protocol that enables a quantum state to be transmitted using entanglement and the communication of two classical bits. Superdense coding is the reverse of teleportation where two classical bits are transmitted by communicating a qubit. The protocols are the building blocks to provide information on the construction of a quantum computer based on just single qubit operations, measurements and entanglement [79].

In quantum information processing (QIP), *qudits* (d-level systems) are an extension of qubits that could improve the speed of computation in comparison to the two dimensional quantum systems. Higher-dimensional QIP and cryptography is an exciting feature which is exhibited by optical quantum computing using another intrinsic property of photon called the orbital angular momentum (OAM). There has been a significant interest in this area of research as it offers the possibility of higher rate of data transmission and more powerful security of cryptographic systems. We will be looking into this property of light in more detail in Chapter 8 of the thesis.

1.2 Motivation

After a brief review of the field of quantum computation and information, we focus our attention now on the motivation of our research in the theory and applications of *formal methods* to quantum computing and quantum information.

Formal methods comprise a range of mathematical techniques and tools that are used in the field of theoretical computer science for modelling and verifying the correctness of systems. Each technique comes with a specification language for modelling systems and semantics that helps to describe the systems' behaviour.

Lowe [115] used process algebra CSP [90] and the automated Failure Divergences Refinement (FDR) model checking tool [147] to formally analyse the Needham-Schroeder public key authentication protocol [131]. He discovered a flaw in the security of the protocol and verified a corrected version. NASA uses formal methods in a number of projects. These methods also have an impact in certain areas like microprocessor designs [98] and safety-critical or high assurance systems [165].

The success of these methods in classical computer science is one of the prime motivation for applying them to quantum information processing (QIP) systems. We are mainly concerned with communication and cryptographic systems, which will produce benefits similar to those already achieved for classical systems. We use *formal methods* in the following ways:

- *Formal modelling languages* for describing systems at various levels of abstraction.
- *Property specification languages* for characterising the properties of systems.
- *Compositional analysis* for verifying systems by analysing their individual components in isolation.
- *Automated tools* for facilitating modelling and analysing for large scale applications.

Quantum process calculi have been developed as part of a programme to transfer ideas form the field of *formal methods* to quantum systems. On applying these techniques to QIP systems, one could achieve a conceptual understanding of concurrent, communicating systems. Mayers [120] has proved that the quantum key distribution protocol BB84 is *unconditionally secure*. But, the information-theoretic proof doesn't necessarily confirm that the implemented systems are unconditionally secure. Hence, another motivation to use formal techniques in QIP is to develop tools for verifying the correctness of practical quantum technologies such as cryptosystems.

Communicating Quantum Processes (CQP), a quantum process calculus developed by Gay and Nagarajan [74], is based on the classical π-calculus [126, 150] with the addition of operations for QIP. Another established quantum process calculus is qCCS by Feng *et al.* [65]. The property of *behavioural equivalence* of processes in quantum process calculus helps to verify the correctness of a system. First, we define two processes: *System* (which models the system of interest) and *Specification* (which expresses the desired behaviour of *System*), and then prove that these two processes are equivalent, i.e. the behaviour of the two processes are indistinguishable to an observer. The *congruence* property of equivalence makes it more powerful by preserving the equivalence in any environment. This has been defined for CQP [51] and qCCS [66]. Also, the property supports equational reasoning which reduces the need to explicitly construct bisimulation relations which is reported in [51] for CQP with an analysis of the quantum teleportation protocol.

1.3 Thesis Contribution

The aim of the work described in this thesis is to investigate the theory and applications of quantum process calculus, Communicating Quantum Processes (CQP). By using the theory of behavioural equivalence of CQP [51], we analyse a simple three qubit flip error correcting code [132] and verify two models of the error correction systems by proving that they are equivalent to their respective specifications. The work is presented in Chapter 4 of this thesis and is also reported at 8^{th} International Workshop on Quantum Physics and Logic (QPL 2011) [52].

Automated tools [11, 12, 77], are able to verify that a quantum protocol satisfies a specification by using the stabilizer formalism [132]. The formalism provides an efficient simulation but is restricted to Clifford group operations [132]. The use of process calculus approach that is demonstrated in this thesis provides two significant advantages in comparison with the automated tools:

- First, since there is no computer simulation, we are not restricted to stabilizer states.
- Second, since the equivalence is a congruence [51], we can use equational reasoning.

The equational theory of CQP [51] helps to deduce further equivalences, whereas in the model-checking approach we only obtain the particular fact that is checked. The equational axioms are presented in [51] and are used in the analysis of quantum teleportation.

We define an additional three new axioms which helps us to take a step further to analyse protocols comprising *superdense coding*, *quantum error correction*, *quantum secret sharing* and *remote CNOT*.

In all previous work of quantum process calculus, a qubit is considered as a *localised* unit of information. Modelling in CQP provides us an abstract view of the quantum system. We present in this thesis an extension of the language CQP to model realistic QIP systems such as LOQC and the associated experimental processes. This work has been reported at the 5^{th} International Conference on Reversible Computation (RC2013) [70].

In order to have a physical understanding on the property of equivalence, we present in Chapter 8 the extension of the theory of equivalence of CQP to verify linear optical quantum computing (LOQC). This work has ben reported at the Combined 21^{st} International Workshop on Expressiveness in Concurrency and 11^{th} Workshop on Structural Operational Semantics (EXPRESS/SOS 2014) [71].

As mentioned in the earlier section, there has been a significant interest in higher-dimensional QIP and cryptography. The earlier work of CQP has been primarily focussed on describing systems comprising quantised 2 level systems. The general framework of CQP makes it easier to adapt to certain tasks. We demonstrate this by extending the language to describe higher dimensional quantum protocols, i.e. qudit teleportation and superdense coding. This work has been reported at the 9^{th} International Workshop on Quantum Physics and Logic (QPL 2012) [78]. Recent optical experiments have demonstrated higher-dimensional quantum systems by using the orbital angular momentum (OAM) of a photon. In the later part of the thesis, we present an investigation in order to describe or model the experiments in CQP.

1.4 Publications

1. S. Franke-Arnold, S. J. Gay and **I. V. Puthoor** (2014). Verification of linear optical quantum computing using quantum process calculus. *In Proceedings of Combined* 21^{st} *International Workshop on Expressiveness in Concurrency and* 11^{th} *Workshop on Structural Operational Semantics, Electronic Proceedings in Theoretical Computer Science (EPTCS), 160, 111-129.*

2. S. Franke-Arnold, S. J. Gay and **I. V. Puthoor** (2013). Quantum Process Calculus for linear optical quantum Computing. *In Proceedings of* 5^{th} *Conference on Reversible Computation (RC 2013), Lecture Notes in Computer Science, 7948, 264-276.*

3. S. J. Gay and **I. V. Puthoor** (2012). Applications of Quantum Process Calculus to Higher Dimensional Quantum Protocols. *In Proceedings of* 9^{th} *International Workshop on Quantum Physics and Logic (QPL 2012), EPTCS, 158, 15-28.*

4. T. A. S. Davidson, S. J. Gay, R. Nagarajan and **I. V. Puthoor** (2011). Analysis of a Quantum Error Correcting Code using Quantum Process Calculus. *In Proceedings of* 8^{th} *International Workshop on Quantum Physics and Logic (QPL 2011), EPTCS, 95, 67-80.*

5. S. J. Gay and **I. V. Puthoor** (2014). Equational reasoning about quantum protocols. *Under preparation.*

1.5 Outline

A short outline of the work presented in this thesis is as follows:

- **Chapter 2** discusses the literature review.
- **Chapter 3** provides a short review of the relevant background theory and concepts needed in this thesis. We start with some fundamental concepts of quantum information and, then introduce the theory of process calculus in the classical regime.
- In **Chapter 4**, we review the syntax and semantics of CQP and summarise the theory of behavioural equivalence of CQP [51] based on the probabilistic branching bisimilarity. The equivalence is also proved to be a congruence. We use this theory to analyse models of quantum error correction and a model of quantum secret sharing protocol.
- In **Chapter 5**, we present briefly the equational theory of CQP [51] for full probabilistic branching bisimilarity. Then, we define three new axioms and illustrate how we could analyse the protocols comprising *superdense coding*, *quantum error correction*, *quantum secret sharing* and *remote CNOT*. We also prove that the new axioms are sound.
- In **Chapter 6**, we present an attempt to extend CQP to model linear optical quantum computing. We do this by allowing multiple particles as information carriers, described by Fock states. We consider the transfer of information from one particular qubit realisation (polarisation) to another (path encoding), and describe post-selection. We illustrate this approach by presenting a model of an LOQC CNOT gate.

- In **Chapter 7**, we extend the theory of probabilistic branching bisimulation in CQP to model and verify LOQC. We introduce two new measurement semantics in order to work at different levels of abstraction. To illustrate this we present two models of an LOQC CNOT gate and verify them with respect to their specifications. This demonstrates the flexibility of the process calculus approach.

- In **Chapter 8**, we investigate extensions of the syntax and semantics of CQP to model higher dimensional protocols. With the help of the extensions, we model two higher dimensional quantum protocols namely teleportation and superdense coding. We present a study on the OAM of light.

- **Chapter 9** concludes with a final review of our contributions and discussion, as well as directions of future work.

Chapter 2

Literature Review

In this chapter, we provide the six areas of literature that we review in this thesis

- Process Calculus
- Quantum Process Calculus
- Automated verification of quantum systems
- Semantic techniques for the analysis of quantum systems
- Quantum Programming Languages
- Quantum computing using linear optics

Each area is discussed as separate sections in this chapter.

2.1 Process Calculus

Process calculi are formal techniques that help us to describe and analyse the behaviour of classical concurrent systems that combine both computation and communication. Some of the most common examples are *Calculus of Communicating Systems (CCS)* [124], *Algebra of Communicating Processes (ACP)* [26], *Communicating Sequential Processes (CSP)* [90] and *Language Of Temporal Ordering Specification (LOTOS)* [60].

CCS and CSP are used for describing communicating and concurrent processes at a high level syntax. CCS uses operational semantics while CSP and ACP uses denotational and axiomatic semantics respectively. In process calculus, a system and the sub-components

of the system are defined as processes. The process communicate with each other through channels. To illustrate the semantics of CCS, we consider an example of communication between a student and a coffee machine that are defined as processes, **ST** and **CM** respectively

$$\mathbf{ST} = \overline{coin}\,.\,coffee\,.\,\mathbf{ST}$$
$$\mathbf{CM} = coin\,.\,\overline{coffee}\,.\,\mathbf{CM}$$

coin and *coffee* are the actions of the two processes. The over line indicates an output action and the communication between the two processes is given by $\mathbf{ST} \mid \mathbf{CM}$.

Mobility is an important concept which is not modelled by CCS. This can mean several things:

- processes move in the physical space of computing sites;
- processes move in the virtual space of linked processes;
- links or channels move in the virtual space of linked processes.

π-calculus [126, 150] is regarded as an extension of CCS which includes the channel mobility in processes. The ambient calculus [34] evolved from π-calculus and describes the movement of processes through administrative domains. Various notions of equivalence between processes have been defined for these process calculi.

Equivalence relations are important in order to verify the correctness of a system. This is performed by initially defining two processes, *System* and *Specification*. The former models the system of interest and the latter expresses the desired behaviour of *System*. Finally, in order to prove the correctness of *System*, it is required to prove that *System* and *Specification* are equivalent to each other, which means that their behaviour are indistinguishable by any observer. Such a proof can be automated and the *Concurrency Workbench of the New Century (CWB-NC)* [39] is an automation tool which tests the equivalence of processes for formalisms CCS, CSP and others.

2.2 Quantum Process Calculus

Quantum process calculi are the quantum versions of process calculus. In addition to the basic principles of process calculus, quantum process calculus includes certain principles and operations of quantum mechanics in order to take into account of the quantum systems. The quantum process calculi developed up to now are *Quantum Process Algebra (QPAlg)* [95], *Communicating Quantum Processes (CQP)* [74] and *qCCS* [66].

QPAlg [95] is a language for modelling quantum systems and is quite similar to the classical process calculi CCS [124] and LOTOS [60]. The extensions that are added in this quantum process calculus to describe QIP are the rules for applying unitary operators, measurements and the ability to send and receive qubits. QPAlg uses the density matrix representation and an operational semantics is given where the labelled transitions are complemented by probabilistic transitions. The probabilistic transitions arise due to the result of quantum measurements.

There has been investigations on the equivalence of processes in QPAlg [110]. This is obtained by defining a *probabilistic branching bisimilarity* based on the branching bismilarity given by van Glabbeek and Weijland [160]. The equivalence is shown to be preserved by all operators except for parallel composition. The two problems that prevent this preservation of parallel composition are: the restriction of quantum variables to individual processes and the comparison between probabilistic and non-deterministic actions. As a result, the equivalence relation is not a congruence for QPAlg. Apart from this result, there has not been much of a progress in research in QPAlg.

However, there has been a significant progress of research to date for the quantum process calculi CQP and qCCS. Gay and Nagarajan [129] analysed BB84 by modelling it in CCS in combination with the results of some initial analysis using CWB-NC for the verification of the protocol. Their investigations lead them to the development of CQP [74]. A classical model checking tool PRISM [109] has been used for the analysis of the quantum systems as a part of the same research programme [76].

CQP is a quantum process calculus which is based on the π-calculus [150] with the addition of primitive operations for quantum information inspired by Selinger's *quantum programming language (QPL)* [152]. The original operational semantics of CQP is defined using *reductions* under the assumption that the transmission of qubits is internal and no external communication is considered. In other words, it is assumed that the quantum systems are closed to any environment. The quantum measurements lead to probabilistic transitions, and this is similar to the approach of QPAlg. One of the distinctive features of CQP is its type system, which ensures that operations can only be applied to data of the appropriate type. The purpose of the type system is not only to classify quantum and classical data but also to enforce the view of qubits as physical resources, each of which has a unique owning process at any given time. A complete treatment of the type system with associated proofs is presented by Gay and Nagarajan [75].

The reduction semantics allows to define the behaviour of a complete system but it is the *labelled transitions* that are needed to define equivalence between processes. Davidson [51] in his PhD work has provided a different style of definition compared to the

previous work of CQP. The significant difference is in the treatment of quantum measurement by the semantics. As discussed earlier that using the reduction semantics of CQP, a quantum measurement leads to a probabilistic transition. But to prove equivalence of processes to have the important property of congruence, the semantics incorporates an analysis such that a quantum measurement would result in a probabilistic distribution if the measurement outcome is communicated to the environment. Otherwise, if the outcome is communicated internally, it would not result in a probabilistic distribution but would give rise to a mixed distribution. By defining the labelled semantics of CQP, Davidson defined the theory of equivalence in CQP based on *probabilistic branching bisimilarity* [10] and applied the result to protocols quantum teleportation and superdense coding. The result is similar to that obtained independently by Feng *et al.* [66] for qCCS. The theory is briefly discussed in Chapter 4 and is applied to quantum error code correction.

The language qCCS by Feng *et al.* [65] is a quantum extension of the classical value-passing CCS. The language uses probabilistic transitions to deal with measurement, however it doesnt treat these as branching transitions, instead maintaining a distribution over each outcome. There has been investigations on process equivalences, namely strong and weak probabilistic bisimilarity, which are shown to be preserved by various operators. More importantly, their equivalences are preserved by parallel composition with processes that do not change the quantum context.

A later version of qCCS [170] excludes classical information in an attempt to better understand quantum processes. In qCCS, the quantum operations are modelled using super-operators. This enables the operational semantics to be defined by a non-probabilistic transition system. Several notions of equivalence are considered by Ying *et al.* [169, 170] and introduced the notion of approximate bisimilarity as a way of quantifying differences in purely quantum behaviour. Their strong reduction-bisimilarity as a congruence is not sufficient for the analysis of most interesting quantum protocols, as the language does not include a full treatment of measurement.

A couple of years later Feng *et al.* [66] define a new version of qCCS. This latest version models general processes comprising classical and quantum and also maintains the use of super-operators. They prove that their weak bisimilarity is a congruence and apply their result to quantum teleportation and superdense coding. Also, using the same result of qCCS, Kubota *et al.* [108] verified the security proof of quantum key distribution protocol BB84 in qCCS.

2.3 Automated verification of quantum systems

Automated model checking techniques have been applied to many quantum protocols. Gay *et al.* [74] use the probabilistic model checker PRISM [109] for the analysis of protocols namely quantum teleportation, superdense coding and quantum error correction protocols. Elboukhari *et al.* [63] also used PRISM for the verification of B92 quantum key distribution protocol [21].

The QMC (Quantum Model-Checker) system [77] is a model checking tool that is developed by Papanikolaou to verify quantum protocols satisfying a specification expressed in a quantum logic. The use of logical formulae is known as *property-oriented* specification, which is different from the *process-oriented* specification adopted in this thesis. For simulation efficiency reasons, QMC is restricted to *stabilizer* formalism and checks properties in *Quantum Computation Tree Logic (QCTL)* [14] on models which lie within the formalism.

Belardinelli *et al.* [18] developed a technique for the verification of quantum protocols using a model checker MCMAS [114]. They used the framework of D'Hondt and Panangaden [57] to specify protocols on the basis of epistemic properties. A compiler is implemented to translate the description of the protocols to the language of MCMAS for analysis.

Recently, Ardeshir-Larijani *et al.* has developed a model checking tool [11] for the verification of quantum protocols using equivalence checking based on Selinter's QPL. The tool uses the stabilizer formalism, verified quantum teleportation and error correction. In a later version, the techniques were extended to model systems comprising concurrent components [12]. The input-output relations are abstracted by superoperators. This enables the analysis of various quantum protocols with arbitrary input, by simulating their operation on a finite basis set consisting of stabilizer states.

2.4 Semantic techniques for the analysis of quantum systems

Abramsky and Coecke [2, 3] developed an approach for analysing quantum protocols by using the mathematical tools of category theory. Their approach is based on recasting the standard axiomatisation of quantum mechanics by employing category theory to describe the protocols at a more abstract level. The method allows for a mathematical analysis of information flow in quantum protocols and have verified the correctness of

teleportation protocol. Related to the work, Duncan [59] constructed a new category-theoretic semantics of multiplicative linear logic within Abramsky and Coecke's framework. The research led to the development of a graphical calculus [40] for reasoning about quantum systems. The diagrammatic reasoning is supported by the underlying categorical theory and using graph rewriting techniques, the idea has been implemented in the tool *Quantomatic* [58, 101, 102].

Perdrix [139] analysed the properties of entanglement using an abstract interpretation method. The study focussed on the evolution of the entanglement, while Prost and Zerrari [144] used the logical approach for the same purpose. Blute *et al.* [29] introduced another category-theoretic framework to understand the behaviour of quantum systems. The focus is on the connections between their approach and linear logic to describe entanglement.

Adao and Mateus [4] designed a process algebra for the analysis of quantum cryptographic systems based on the *quantum random access machine (QRAM)*. The language describes the computational complexity of systems and implements a cost model. They develop the theory of observational equivalence of processes and computational indistinguishability. The language is different from CQP as it describes a system that is a parallel combination of QRAMs.

2.5 Quantum Programming Languages

Some of the programming languages defined for quantum systems till date are: QCL [135, 136], qGCL [149], QPL [151, 152], QML [7, 8] and the quantum λ-calculus of Van Tonder [161]. For more details, see the survey by Gay [73] and Sofge [155].

The general idea of the research in quantum programming languages was to provide an alternate approach to describe quantum systems compared to the one given by the quantum circuit diagrams. Probably, the earliest proposal was given by Knill [105] in 1996 where he defines a pseudocode for the description of quantum algorithms. This was connected to the QRAM. Omer developed one of the first real quantum programming language QCL [135, 136]. He defined the syntax using C and implemented a simulator for the language. QCL contains a full programming language as a sublanguage and provides several useful features such as memory management and automatic derivation of conditional versions of operators.

Sanders and Zuliani [149] defined a guarded-command language called qGCL. The semantics of the language is in the form of either predicate transformers or relations, and refinement calculus. The focus of the work is on the derivation of quantum algorithms.

Selinger [152] defines a functional language QPL, with a static type system. The denotational semantic approach uses the standard mechanism of complete partial orders and continuous functions in the framework of vector spaces and superoperators. One of the important feature is the treatment of partiality arising from non-terminating recursion and loops. There are other ideas suggested by the same author [153] for denotational semantics of a higher-order quantum programming language aiming to provide a secure theoretical foundation for different styles of quantum programming language.

Altenkirch and Grattage [7, 8] developed a functional programming language, QML, where the semantics of the language is expressed in category-theoretic terms. The type system of QML is based on linear logic and a sound and complete equational theory for the measurement free QML is given in [9].

Danos *et al.* [47, 48] studied the one-way model of quantum computation. They developed notations based on patterns for entanglement, measurement and local correction. The research focussed on defining a *measurement calculus* based on the equation of patterns. This led to the development of an algorithm where any pattern can be transformed to a form consisting of entanglement, then followed by measurement and by correction. This formalism has been used to derive several properties of measurement-based circuits. A notion of operational equivalence is considered and is applied to shown that quantum teleportation is equivalent to a direct quantum channel [46].

Recently, Green *et al.* [81] introduced an embedded functional programming language for quantum computation called *Quipper*. The language uses Haskell as the host language and thereby it can use a collection of data types, combinators, and a library of functions of Haskell, along with an *idiom* (i.e. a preferred style of writing embedded programmes). The authors illustrate the language by describing quantum teleportation, the quantum Fourier transform (QFT), and an application of QFT known as quantum adder.

2.6 Quantum computing using linear optics

Optical implementations are one of the prominent candidates of quantum computing. The important reason is that photons are easily generated, manipulated and detected. Several proposals that manipulate the state of light are carried out. This ranges from cat state logic [145] to encoding a qubit in harmonic oscillator [80] and continous-variable quantum computing [113].

In this thesis, we focus on quantum computing with linear quantum optics and single photons. The advantage of using photons is that it possess large coherence times which makes them suitable for computation and communication applications. The drawback

is that there is no natural interaction between photons which makes it hard to implement two-qubit quantum gates that are essential for quantum computing. To introduce an effective interaction between photons in one way or another is the interesting and essential part in an optical quantum computer.

One of the methods to induce the interaction is to introduce nonlinearities referred to as cross-Kerr effects [37, 92]. Although these nonlinearities induce a single-photon controlled-Not operation, they are very small in magnitude to serve the actual purpose of quantum computing. Another alternate method to produce an effective interaction between photons is to make projective measurements with photodetectors. The difficulty of this technique arises due to the probabilistic nature of the optical quantum gates. This is because the gate fails more often and destroys the information in the quantum computation. But this can be overcome by using a polynomial number of optical modes.

In [36], Cerf *et al.* proposed a scheme for quantum logic with only linear optical devices and a single photon. To simulate n qubits, a single photon is put into 2^n different paths. They show that implementation of a universal set of gates is possible by demonstrating a Hadamard, CNOT and reverse CNOT gates. To implement a Hadamard gate, they use linear optical elements like beam splitter and phase shifters. These elements are vital and more discussions are present in Chapters 6 and 7 of the thesis. Their CNOT gate is encoded with respect to the polarisation and position of photon. The problem with this scheme is that n qubits requires 2^n paths which in turn requires $2^n - 1$ beam splitters to setup, which is not scalable as it means that one qubit encoded in polarisation will need 2^{n-1} optical paths.

In 2001, Knill, Laflamme and Milburn (KLM) [106] designed a protocol showing the possibility of scalable quantum computing by using only single photon sources and detectors, and simple (linear) optical circuits consisting of beam splitters. They demonstrated that two-photon gates that are probabilistic can be teleported into a quantum circuit with high probability. The protocol initiated experiments in quantum optics that demonstrate the operation of high-fidelity probabilistic gates [134, 141].

Prior to the work of KLM, the concept of scalable quantum computing was believed to be performed using a non-linear component, such as Kerr medium. These media are characterised by having a refractive index that contains a nonlinear component. Yamamoto *et al.* [168] developed a Kerr base Fredkin gate which gave rise to several architectures based on nonlinear optical gates [49, 91]. More details are provided in the review article by Kok *et al.* [107].

Chapter 3

Background

In this chapter, we provide a review on the important concepts that are needed for the understanding of the thesis.

3.1 Qubits

The fundamental unit of quantum information processing (QIP) is a *quantum bit* or a *qubit*. A *qubit* is quantum analogue of a classical bit. It is associated with a complex *Hilbert space* $\mathbb{H}$, called its *state space*. Any quantum system is completely described by a *state vector* $|\psi\rangle$ within its state space, which is a 2-dimensional vector space over the complex numbers ($\mathbb{C}$).

The set of vectors $\{|0\rangle, |1\rangle\}$ is called the *standard basis* of the state space $\mathbb{H}$. We can write the general state of a qubit as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \tag{3.1}$$

where $\alpha_0, \alpha_1 \in \mathbb{C}$ are complex amplitudes such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$. In comparison to a classical bit, whose state is either 0 or 1, the state space of a qubit therefore consists of all *superpositions* of the basis states. The states alternatively can be represented by column vectors:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha_0|0\rangle + \alpha_1|1\rangle$$

The state space of a multiple qubit system is given by the *tensor product* ($\otimes$) of each qubit state space. An *n-qubit* is a state in the tensor product Hilbert space given by $(\mathbb{H})^{\otimes n} = \mathbb{H} \otimes \ldots \otimes \mathbb{H}$. The *standard basis* is the orthonormal basis given by the 2^n

classical n-qubits.

$$|i_1 i_2 \ldots i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_n\rangle$$

where $i_j \in \{0, 1\}$.

For example, a two-qubit system has the orthonormal basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. The general state is given by the state vector:

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} \tag{3.2}$$

where $\alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{C}$ and $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$

The two-qubit state can be represented by the tensor product of two single qubit states provided the two-qubit quantum state is separable. For example, if we have have a two-qubit state given by $|\psi\rangle = \alpha_0\alpha_2|00\rangle + \alpha_0\alpha_3|01\rangle + \alpha_1\alpha_2|10\rangle + \alpha_1\alpha_3|11\rangle$. This state can be written as $(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\alpha_2|0\rangle + \alpha_3|1\rangle)$ and is given by the vector notation:

$$|\psi\rangle = \begin{pmatrix} \alpha_0\alpha_2 \\ \alpha_0\alpha_3 \\ \alpha_1\alpha_2 \\ \alpha_1\alpha_3 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \alpha_3 \end{pmatrix} \tag{3.3}$$

3.2 Quantum entanglement

One of the concepts which has brought a lot of discussions in the foundations of quantum mechanics is quantum entanglement. This originates from the famous Gendanken experiment proposed by Einstein, Podolsky and Rosen (EPR) in 1935 [61]. Entanglement is a non local quantum correlation between two or more quantum-mechanical systems. This means that the individual outcomes of the observables cannot be determined with certainty for each of the two or more EPR systems, but the outcomes of the observables for the systems are always strictly correlated. The existence of such non local quantum correlation was established by Bell in 1960 and can be quantified by the Bell's inequality [20]. The generalisation is known as CHSH-Bell's inequality [38]. Quantum systems that exhibit entanglement may be either two-level (qubit) systems such as electron spins and photon polarisations, or continuous variable systems such as position-momentum [61, 159], or discrete systems in higher dimensions, as orbital angular momentum (OAM) [94, 121]. In our work, we use the most simple and primary entangled system, that is the entanglement shared between two qubits.

An example for an entangled state between two qubits in two distinguishable systems are given by

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.4}$$

The state given by Eq. 3.4 cannot be written as a direct product of two separate systems which means that the two systems are no longer independent but hold quantum correlation between them.

The *Bell bases* or *Bell states* are the four orthogonal states that are given by:

$$|\phi^\pm\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle),$$

$$|\psi^\pm\rangle = \tfrac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$$

These states hold complete entanglement and any 2-qubit state can be produced by a linear combination of the Bell states.

3.3 Quantum operators for qubits

The time-evolution of a closed quantum system can be described by *unitary operations* that are acted upon the quantum state. A *linear operator* M on the Hilbert space $\mathbb{H}$ is a mapping that assigns to every state $|\psi\rangle$ in $\mathbb{H}$ a state $M|\psi\rangle$ in $\mathbb{H}$, in such a way that

$$M(|\psi\rangle + |\phi\rangle) = M|\psi\rangle + M|\phi\rangle$$

A linear operator U is *unitary* if $UU^\dagger = U^\dagger U = I$, where I is the identity operator and the symbol $U^\dagger$ is the conjugate transpose of U. For example, the *Hadamard* transformation is defined by

$$\mathsf{H}|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$\mathsf{H}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

This is interesting as it can create and remove superpositions and corresponds to the matrix:

$$\mathsf{H} = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

We will also make use of the *Pauli operators* and give their matrix representations with respect to the standard basis. These are single qubit operators, denoted by $\mathsf{I}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}$ or

$\sigma_0, \sigma_1, \sigma_2, \sigma_3$.

$$\mathsf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathsf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\mathsf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \mathsf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The action of these quantum operators on the quantum state $|\psi\rangle$ given by Eqn. 3.1 are as follows:

$$\begin{aligned} \mathsf{I}|\psi\rangle &= \alpha_0|0\rangle + \alpha_1|1\rangle \\ \mathsf{X}|\psi\rangle &= \alpha_0|1\rangle + \alpha_1|0\rangle \\ \mathsf{Y}|\psi\rangle &= -i\alpha_1|0\rangle + i\alpha_0|1\rangle \\ \mathsf{Z}|\psi\rangle &= \alpha_0|0\rangle - \alpha_1|1\rangle \end{aligned}$$

The next quantum operator is important as it is a primary component in building a quantum computer. This is the *controlled-NOT* or CNOT operator. The matrix representation for a two-qubit CNOT operator is given by

$$\mathsf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The action of the CNOT operator is that it flips the second qubit (target qubit) if and only if the first qubit (control qubit) is 1. On basis states, we have $\mathsf{CNOT}|0x\rangle = |0x\rangle$ and $\mathsf{CNOT}|1x\rangle = |1y\rangle$ where $x, y \in \{0, 1\}$ and $y = x \oplus 1$ with $\oplus$ denoting addition modulo 2. The combination of CNOT and Hadamard operator, is mainly used to create and remove entanglement. For example, if we have a two qubit state $|\psi\rangle = |00\rangle$ (which is separable), we then apply the Hadamard operator on the first qubit, followed by a CNOT operation to both qubits, to get:

$$\mathsf{CNOT}.(\mathsf{H} \otimes \mathsf{I})|00\rangle = \mathsf{CNOT}(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

3.4 Pure and mixed states: density matrix

A quantum state is said to be *pure* if it is represented by a single ket vector in a Hilbert space, which is $|\psi\rangle = \sum_{i\in\{0,1\}} \alpha_i|i\rangle$. The Bell states are *pure states* since they are expressed by the linear combination of the basis vectors. A *mixed* quantum state is a collection of pure states $|\psi_i\rangle$, each associated with probability p_i satisfying the conditions $1 \geq p_i \geq 0$ and $\sum_i p_i = 1$. The main reason to consider mixed states is

because the quantum states are difficult to isolate and hence are often entangled with the environment. So, in order to express quantum systems which includes mixed states in general, we introduce the *density matrix* (denoted as ρ) representation for an *ensemble* of pure states $\{p_i, |\psi_i\rangle\}$

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

For example, for a given state $|\psi\rangle$, if $|\alpha_0|^2$ and $|\alpha_1|^2$ are respective probabilities for the states $|0\rangle$ and $|1\rangle$, then the density matrix is given by

$$\rho = |\alpha_0|^2|0\rangle\langle 0| + |\alpha_1|^2|1\rangle\langle 1| = \begin{pmatrix} |\alpha_0|^2 & 0 \\ 0 & |\alpha_1|^2 \end{pmatrix}$$

The density matrix representation also helps to describe subsystems within a composite system. Suppose, we have a composite system whose Hilbert space is given by the tensor product $\mathbb{H}_A \otimes \mathbb{H}_B$ where $\mathbb{H}_A$ and $\mathbb{H}_B$ are the Hilbert spaces for the subsystems A and B respectively. If $|\Psi\rangle = \sum_i |\psi_i\rangle_A |\phi_i\rangle_B$ and ρ_{AB} is the density matrix of the system, then the subsystems can be described by their *reduced density matrices*. The reduced density matrix of A (respectively B) is

$$\rho_A = Tr_B(\rho_{AB}) \quad \rho_B = Tr_A(\rho_{AB})$$

Here the trace is performed over $\mathbb{H}_B$ only (respectively $\mathbb{H}_A$). This is called a partial trace and can be defined as follows

$$\rho_A = Tr_B(\rho_{AB}) = Tr_B(|\psi_i\rangle\langle\psi_j| \otimes |\phi_i\rangle\langle\phi_j|) = |\psi_i\rangle\langle\psi_j|(Tr|\phi_i\rangle\langle\phi_j|) = |\psi_i\rangle\langle\psi_j|\langle\phi_j|\phi_i\rangle$$

For instance, if we consider an entangled 2-qubit system such as $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ then the reduced density matrix ρ_A of the qubit A is given by

$$\rho_A = \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.5}$$

This is the completely mixed state in the 1-qubit system. In general, any state whether it's pure or mixed is characterised by its density matrix. It's also important to note that two systems can have the same density matrix but need not have the same state and still would produce the same results on measurement. To illustrate this, we consider a mixed state $|+\rangle$ (with probability $\frac{1}{2}$) and $|-\rangle$ (with probability $\frac{1}{2}$) where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Then, we get:

$$\rho_A = \frac{|+\rangle\langle+| + |-\rangle\langle-|}{2} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.6}$$

Therefore, from Eq. 3.5 and Eq.3.6 we can say that two different mixed states can have the same density matrix.

3.5 Measurement

Measurement is a process performed to extract information from a quantum state $|\psi\rangle$. This is an *irreversible* process as once it is completed and the information is obtained, it is not possible to return to the initial state. It is also a *non - deterministic* process as quantum measurements produce a probabilistic outcome that is dependent on the state of the system. Quantum measurements are described by a set of *measurement operators* $\{M_m\}$ that act on the state space of the system. For an orthonormal basis $\{|\psi_m\rangle\}$, a measurement on the quantum state $|\psi\rangle$ in the basis representation will provide the value ψ_m. This defines the measurement operator

$$M_m = |\psi_m\rangle\langle\psi_m| \tag{3.7}$$

that acts on the state $|\psi\rangle = \sum_{i\in\{0,1\}} \alpha_i|i\rangle$. Thus, the measurement operator M_m extracts the component of a quantum state associated with ψ_m,

$$M_m|\psi\rangle = |\psi_m\rangle\langle\psi_m|\psi\rangle = \sum_{i\in\{0,1\}} \alpha_i\delta_{im}|\psi_m\rangle = \alpha_m|\psi_m\rangle \tag{3.8}$$

Measurement operators are Hermitian, that is $M^\dagger = M$, and the index m refers to the possible measurement outcomes. For a system in state $|\psi\rangle$, the probability that the outcome of the measurement is m is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle \tag{3.9}$$

and the state after the measurement is $\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}$.

An important consequence of the measurement process is that it changes the quantum state. Unlike unitary operators, this change is not reversible and hence is not possible to discover more information about the original state through multiple measurements. The

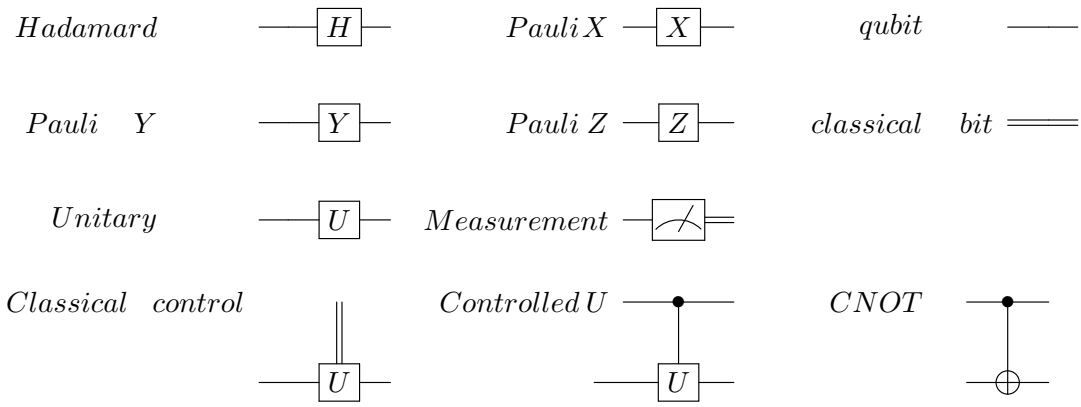


FIGURE 3.1: Basic elements that are used in a quantum circuit

measurement operators of a quantum measurement satisfy the completeness property

$$\sum_m M_m^\dagger M_m = I \tag{3.10}$$

This corresponds to the condition that the probabilities sum to 1.

3.6 Modelling for Quantum computation: Quantum circuits

The fundamentals of quantum operations which can be applied to qubits were introduced in the previous sections. In general, a quantum computation of an algorithm consists of these operations.

The Quantum circuit model was first introduced by Deutsch [54], a convenient method of describing a sequence of quantum operations. This is analogous to classical computing circuits where the logic gates are replaced by quantum gates and the classical wires with quantum wires.

A quantum circuit consists of a finite sequence of parallel wires which run in a single direction from left to right. Each wire represents the state of one qubit. The quantum gates corresponding to the unitary operations that are acting upon the qubits are represented by different boxes, which are denoted in Figure 3.1. The quantum measurements are normally performed on the standard basis $\{|0\rangle, |1\rangle\}$ and are designated by a *meter* symbol. The outcomes of the measurement are classical values represented as double wires or lines. The time scale is increasing, when the circuit is read from left to right.

All quantum gates have the same number of input qubits as output qubits. The classical control of quantum gates is represented by a classical wire entering a quantum gate.

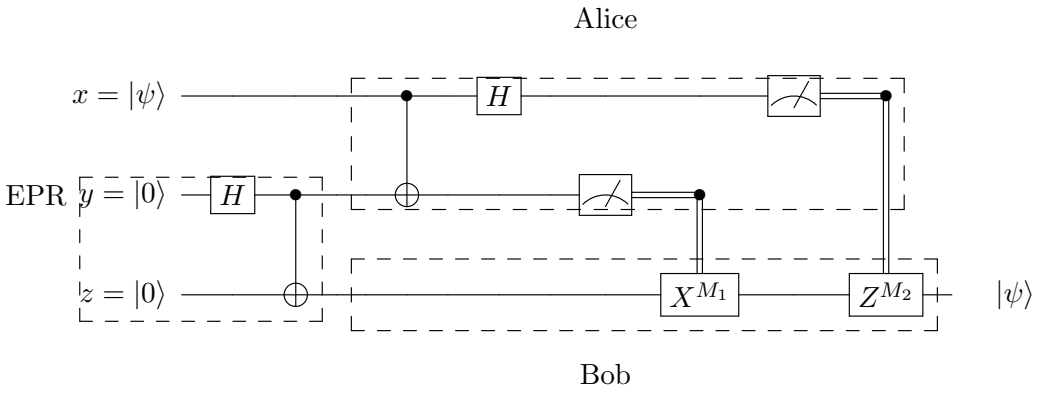


FIGURE 3.2: Teleportation

Controlled U and *CNOT* are the controlled quantum gates. The symbol • represents the control qubit for both gates and for *CNOT* gate, the target is represented with $\oplus$.

In the next section, we describe the modelling and working of certain quantum protocols with the help of quantum circuits.

3.7 Quantum Protocols

3.7.1 Quantum Teleportation

Quantum teleportation [16, 24, 72] is a process by which a quantum state can be transferred from one user to another. This is performed with the help of communicating two classical bits and using an entangled pair of qubits that is shared between the two users. The quantum circuit model of the protocol is shown in Figure 3.2. Using the familiar convention, we say the sender is Alice and the receiver is Bob.

Alice possess the qubit labelled x which is in some unknown state $|\psi\rangle$; this is the qubit to be teleported. Qubits y and z is an EPR pair, which is generated by applying a Hadamard and CNOT operation to the qubits. This is represented by the Figure 3.3 which is a part of the teleportation circuit.

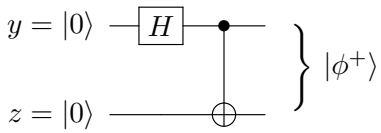


FIGURE 3.3: Entangled pair (EPR pair)

The entangled state or EPR pair, is the Bell state represented as $|\phi^+\rangle$ which is $\frac{1}{\sqrt{2}}(|0\rangle_y|0\rangle_z + |1\rangle_y|1\rangle_z)$. The qubits y and z are given to Alice and Bob respectively.

Before measurement, Alice applies the CNOT operation to her qubits x and y, followed by the Hadamard operator to qubit x. After measuring her qubits, she sends the results

(classical values M_1 and M_2) to Bob. Assuming the arbitrary state is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then there are four possible measurement results with each having the same probability of 0.25.

Quantum State	M_1	M_2	Bob's unitary operators
$\alpha\|000\rangle + \beta\|001\rangle$	0	0	I
$\alpha\|010\rangle - \beta\|011\rangle$	0	1	Z
$\alpha\|101\rangle + \beta\|100\rangle$	1	0	X
$\alpha\|111\rangle - \beta\|110\rangle$	1	1	ZX

TABLE 3.1: Teleportation: Operation of Bob

Table 3.1 shows the four possible cases where Bob can fix up his state to recover $|\psi\rangle$ by applying the appropriate unitary operations. Based on the classical bits (M_1 and M_2), Bob applies the necessary quantum operators to his qubit z. By performing this, he can recover the original state $|\psi\rangle$. For example, if we see the first case, Bob gets 0 as both values from Alice. He knows that the state of his qubit z is the same as that of $|\psi\rangle$ and it is not necessary for him to apply any unitary operations.

3.7.2 Superdense Coding

Superdense coding [25] is a protocol which is similar to quantum teleportation as it involves two parties (Alice and Bob). As in the previous protocol, Alice and Bob may be a long distance away from one another. The difference between this protocol and teleportation is that in this case the goal is to transmit some classical data from Alice to Bob. Alice is in possession of two classical bits which she communicates to Bob by exchanging a single qubit. The term *superdense* refers to this doubling of efficiency.

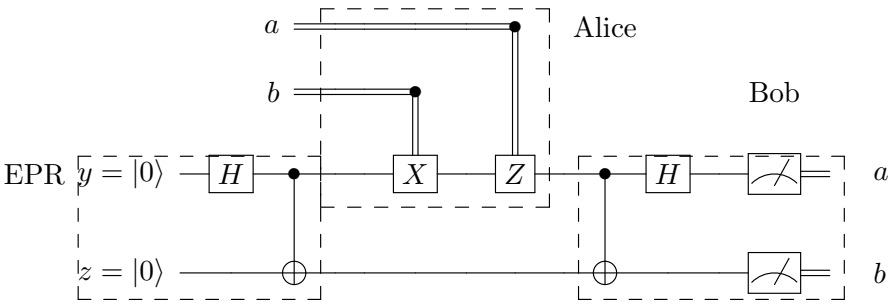


FIGURE 3.4: Superdense coding protocol

The quantum circuit for this protocol is given in Figure 3.4. As in teleportation, this protocol also involves the two users sharing a pair of entangled qubits (EPR Pair). Alice is having the first qubit y, while Bob has possession of the second qubit z. Alice also has two classical bits a and b, which she intends to communicate to Bob. She performs

the task by applying a combination of the Pauli operators X and Z to her qubit y depending on the classical values a and b. The theory of Pauli operators were discussed in the previous section 3.3. By sending the single qubit in her possession to Bob, it turns out that Alice can communicate two classical bits to Bob. Table 3.2 provides the operation of the protocol. First, Alice performs her encoding.

a	b	Alice Operation	Resulting quantum state	Bob's quantum state
0	0	I	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\|00\rangle$
0	1	X	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$	$\|01\rangle$
1	0	Z	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\|10\rangle$
1	1	XZ	$\frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$	$\|11\rangle$

TABLE 3.2: Operation of superdense coding protocol

After the encoding, she sends her qubit to Bob. Bob now has two qubits. He does a CNOT and a Hadamard operations before he performs the measurement on the two qubits. The quantum state before measurement is $|ab\rangle$ and the results he obtains from the measurement are the classical bits that Alice wishes to communicate. The outcomes are certain because the resulting quantum state is not a superposition.

3.8 Quantum Error Correction

As in any information processing systems, noise or errors are a great problem in quantum computing. Errors can arise when qubits are sent along quantum channels which causes *decoherence*. It can also arise from entanglement with the environment. This can result in the input state being changed. Quantum error-correcting codes [56, 132, 157] are introduced to protect the quantum information against the noise. The idea, called *redundancy*, is to introduce additional information apart from the original message. Redundancy is performed by *encoding* the qubits in a way to protect them against the effects of noise. Later after the computation, the qubits are then recovered by the process of *decoding*. We assume that encoding and decoding are done perfectly and do not cause any error. There are quite a few quantum error correcting codes and in this thesis, we concentrate on the three qubit bit flip code.

3.8.1 Quantum Error Correction Codes (QECC)

Like the classical error correction code, a QECC system comprises of four stages as represented in Figure 3.5.

FIGURE 3.5: Four main stages of quantum error correction.

The first stage involves encoding the m qubits into n qubits where $n > m$ and the extra $n - m$ qubits are the redundancy which protects actual data from noise. It is assumed that the errors can only occur during the transmission. The next two stages are error detection and recovery, which happens before the message is received. These two stages combine to form the *error-correction*. The final phase is decoding the qubits to retrieve the original data. The two simplest codes are the three qubit bit flip code and the three qubit phase flip code. In this work, we concentrate on the three qubit bit flip code and the quantum circuit for the three qubit error correction is given in Figure 3.6. The boxed lines in the circuit represents the stages of the quantum error correction code.

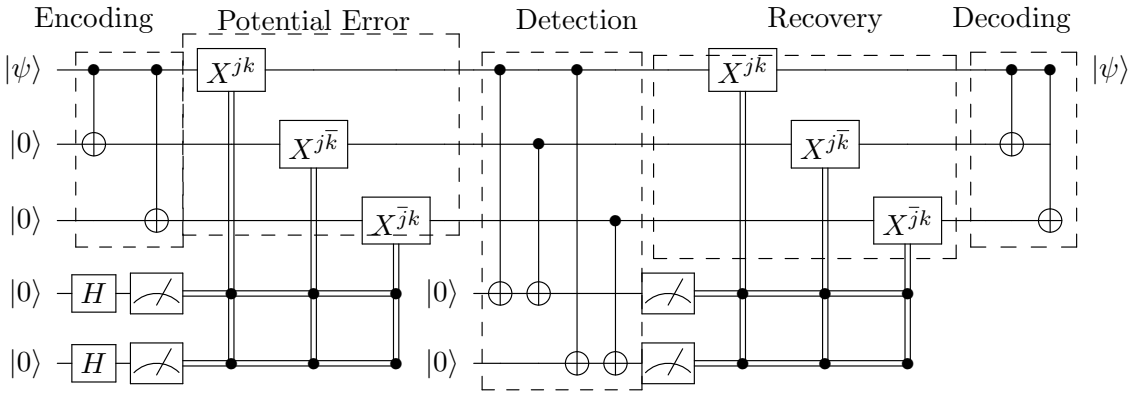


FIGURE 3.6: Three qubit error correction

The encoding for the two pure states of the three qubit bit code are defined as

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$

$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

This is similar to the classical three bit repetition code but in this case only the basis states are cloned. This is because of the no-cloning theorem. Hence, an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is encoded as $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$. Each of the three qubits is then transmitted along a channel which can cause a bit flip. The code can correct up to 1 bit flip error. Bit flip error is equivalent to a Pauli X operator on a qubit.

Error-correction involves first detecting the error and then correcting it. The detection involves in carrying out a projection measurement performed by the projection operator P. This helps to determine about the occurrence of the error and also provides the information on which qubit is flipped if the error has occurred. The measurement result is called the *error syndrome* and there are four different error syndromes corresponding

to a projection operator being used. Table 3.3 shows the projection operators used and the type of error that has occurred.

Projection Operator	Error indication	Syndrome Results	Action
$P_0 \equiv \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	No error	0	No action
$P_1 \equiv \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	Bit flip on qubit 1	1	X_1
$P_2 \equiv \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	Bit flip on qubit 2	2	X_2
$P_3 \equiv \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	Bit flip on qubit 3	3	X_3

TABLE 3.3: Three qubit bit flip code

For example, if the bit flip occurs on the first qubit then the quantum state with error is $|\psi_L\rangle = \alpha|100\rangle + \beta|011\rangle$. Hence, the outcome of the operation $\langle\psi_L|P_1|\psi_L\rangle$ is always 1. It is important to note that the error syndrome measurement does not alter the state before and after the measurement but only contains the information about the kind of error which has occurred. The result of the error syndrome results are provided in table 3.3. Depending on the result the necessary unitary operation is performed on the qubit to flip it again in order to recover the original state

3.9 Process Calculus

Process calculi (or process algebras) are algebraic methods which are used for formally modelling systems that involve concurrent and communicating components. They provide laws that allow formal reasoning about equivalences between systems. The most common process calculi are CCS [123, 124], ACP [26] and CSP [89]. The key features for all process calculi are the following:

- *Syntax.* This is the basic component of process algebra and is determined by the combination of operators and some primitives. This is a set of rules that define the combinations of objects that are considered to be perfectly structured programs in that language.

- *Semantics.* The syntax is accompanied with the semantics to describe the behaviour of the system. There are many approaches in describing the semantics of sequential systems and the main ones are namely operational, denotational and algebraic semantics. The work in this thesis follows structural operational semantics that defines step by step execution of a system.

- *Behavioural equivalence.* Mechanisms that allow to analyse the systems and identify whether they exhibit the same behaviour. In some of the process algebras, there are certain algebraic rules which are called equational axioms that helps

in capturing certain identical properties of the systems through the behavioural operational semantics.

The process calculus approach to verification is to define a process *Model* which models the system of interest, another process *Specification* which expresses the specification that *Model* should satisfy, and then prove that *Model* and *Specification* are equivalent. Usually *Specification* is defined in a sufficiently simple way that it can be taken as self-evident as it describes the intended behaviour at a high-level. The correctness of the *Model* with respect to the *Specification* can be proved by using the theory of behavioural equivalence, which is discussed in detail in the later part of the thesis.

3.9.1 Labelled Transition Systems

An *operational semantics* of process calculus models a system by either a *reduction system* or a *labelled transition system (LTS)*. The first scenario describes the evolution of a system without interacting with the environment by using sequentialisation or inter-process communications. The second case describes the evolution of a system which also includes the interactions between the system and the environment. This is essential as we study the behaviour of communications systems not only with respect to the interactions between its components but also the influence of the surrounding the system is placed in. Hence, we concentrate the rest of the thesis on the use of LTS rather than the reductions.

The LTS consists of a set of *states*, a set of *transition labels* and a *transition relation*. The states $\{Q_i\}$ are generally the process terms while the transition labels are the *actions* $\{\alpha_i\}$. The actions represent the interactions that are possible between the states and normally are classified as *visible* or *observable* and *invisible*. The visible actions include the *input* and the *output*. We will use the notations $a?[x]$ and $a![x]$ for input and output respectively, where a is the channel through which the data x is communicated. The invisible action is denoted as τ which represents the internal action. Together, we use the notation $P \xrightarrow{\alpha} Q$ as a transition relation, which means that the process P can perform an action α and after completing the action it would reach to the state where its remaining behaviour is Q.

The transition relation $P \xrightarrow{\alpha_i} Q_i$ represents *branching* where $\{Q_i\}$ is a set of states and $\{\alpha_i\}$ is a set of actions which P can perform, which we can infer to the *capability* of P. In order to associate an LTS to a process term, the inference systems are used [140].

Inference System. This is a set of inference rules of the form:

$$\frac{p_1, \ldots, p_n}{c}$$

where $p_1, \ldots, p_n$ are the *premises* and c is the *conclusion*. Each rule implies that if all the premises are true then the conclusion is true. If there are no premises then the rule is called an *axiom* and is of the form:

$$\frac{}{c}$$

Transition Rules. The set of rules for operational semantics using the above inference system are defined as actions ($\alpha \,.\, P$ performs α and then behaves as P), parallel composition ($P|Q$ allows computation in P and Q to proceed simultaneously and independently) and choice ($P + Q$ behaves as P or as Q):

$$\alpha \,.\, P \xrightarrow{\alpha} P$$

$$\frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \qquad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

The rule for synchronisation which allows interaction between the processes, that is the communication between P and Q, and the value-passing is:

$$\frac{P \xrightarrow{a![y]} P' \qquad Q \xrightarrow{a?[x]} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{y/x\}}$$

In the above case, communication results from an output by one process (P) and a corresponding input by another process (Q). A common feature in value-passing calculus is that the output values are substituted in the receiving side. This is denoted as $Q'\{y/x\}$ where the value y is substituted in place of x. The communication is considered as an internal action , and hence is classified as a τ action.

Null process. Process algebra also includes a null process (denoted as $\mathbf{0}$) which has no transition. The semantics of this process is designed by the fact that there is no rule to define its transition.

In a formal way, we say that an LTS represents a directed graph where the nodes corresponds to the process terms or states and the transition relation $\xrightarrow{\alpha}$ between

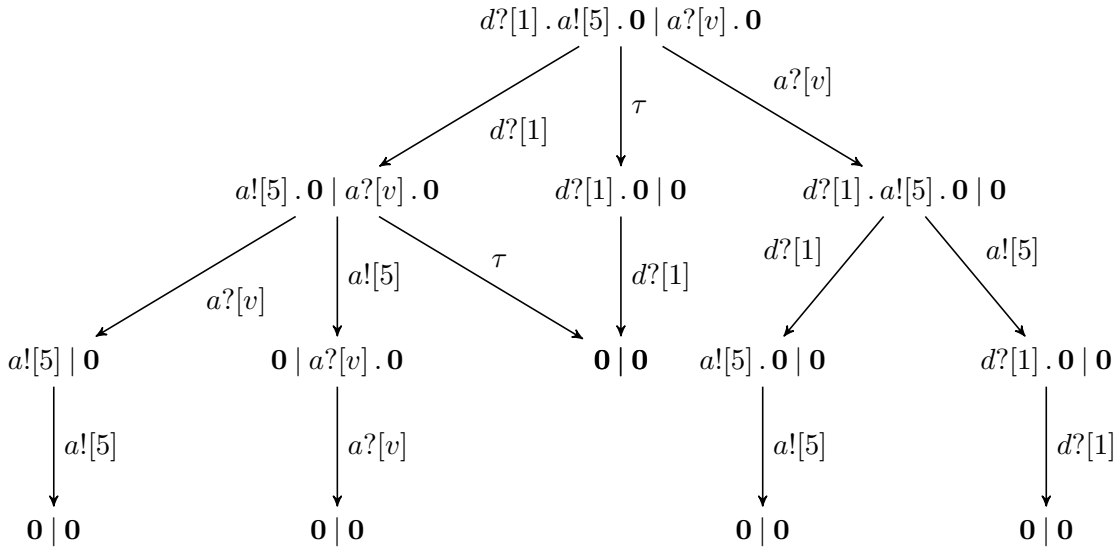


FIGURE 3.7: A labelled transition system

nodes corresponds to process transitions. Figure 3.7 is an example of a LTS where the transitions through the graph represent the possible computations.

The process $d?[1] \,.\, a![5] \,.\, \mathbf{0} \mid a?[v] \,.\, \mathbf{0}$ has two components which are in parallel. The branches show the possible computations of these two processes. The leftmost branch, labelled as $d?[1]$, represents an external communication (that is input from environment) and then is followed by a synchronisation or internal communication, which is again represented by a τ. The middle branch shows that the synchronisation step happens first and is then followed by unitary operation which in case of parallel composition that the computations can occur independently. The right most branch shows an external input action and every node represents a process term that describes the behaviour at that point.

3.9.2 Behavioural Equivalence - Bisimulation

The concept of *behavioural equivalence* helps to analyse the behaviour of the system. It is useful to have theories which can establish whether two systems are equivalent or how "approximately" equivalent they are with each other. The idea is that two processes are equivalent if their behaviour is indistinguishable by an observer. That is, if they do the same thing in the same circumstances. If the same techniques are used to model what is required of a system (its *Specification*) and how it can actually be implemented (its *System*) then it is possible to use the concept of equivalence to prove that a particular description of a system is correct with respect to a given abstract version.

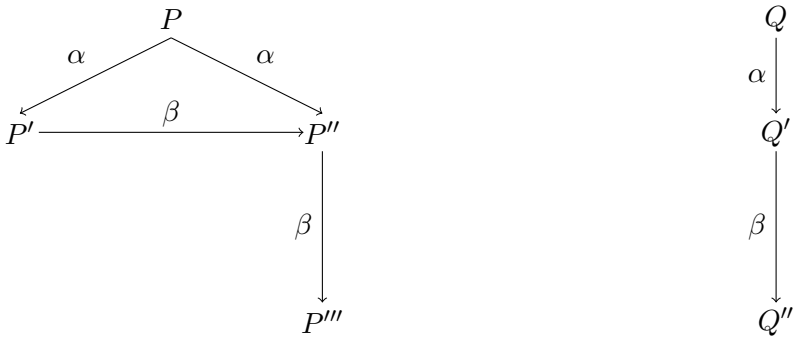


FIGURE 3.8: Strong Bisimulation

Bisimulation equivalence also known as observational equivalence was developed by Park [138]. This considers two systems to be equivalent if they can simulate each other step after step which is based on the concept of *simulation* due to Milner [123]. Bisimulation requires that the simulation relation to be symmetric. Hence this is stronger than the simulation relation as it not only requires the two processes to simulate each other but their simulation relations has to be symmetric to each other.

Strong Bisimulation

Strong bisimulation requires every action, whether its visible or internal of the processes to match each other.

Definition 3.1 (Strong Bisimulation)**.** A relation $\mathcal{R}$ is a *strong bisimulation* if whenever $(P, Q) \in \mathcal{R}$ then for all labels α, both

1. if $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$, and
2. if $Q \xrightarrow{\alpha} Q'$ then $P \xrightarrow{\alpha} P'$ and $(P', Q') \in \mathcal{R}$.

For a given labelled transition system there are many relations that have the property of strong bisimulation, including (trivially) the empty relation. The key idea is to define the largest strong bisimulation which is *strong bismilarity*. In other words, P and Q are *strong bisimilar* (denoted $P \sim Q$) if and only if there exists a bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$. Figure 3.8 gives an example of strong bisimulation ($P \sim Q$). We find that there exists a relation where $\mathcal{R} = \{(P, Q), (P', Q'), (P'', Q'), (P''', Q'')\}$.

Weak Bisimulation

Generally in any two system, the internal behaviours are often different. Strong bisimulation requires each and every computation to be matched which makes it not possible to verify the systems if their internal actions are not the same. In order to solve this problem, there is another useful concept in process algebra called the *weak bisimulation*

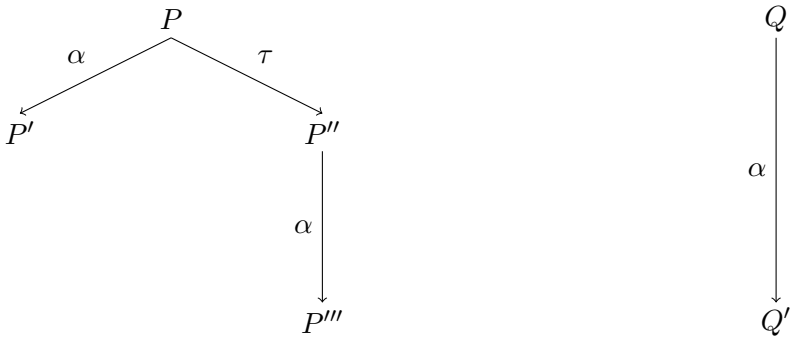


FIGURE 3.9: Weak Bisimulation

which verifies the system that possess external actions in spite of having different internal actions. In general, weak bisimulation allows internal actions to be matched by *zero* or *more* τ actions.

We use the notation $\Longrightarrow$ to denote zero or more τ transitions; and $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\overset{\alpha}{\longrightarrow}\Longrightarrow$.

Definition 3.2 (Weak Bisimulation)**.** A relation $\mathcal{R}$ is a *weak bisimulation* if whenever $(P, Q) \in \mathcal{R}$ then, both

1. if $P \overset{\alpha}{\longrightarrow} P'$ then $Q \overset{\alpha}{\Longrightarrow} Q'$ and $(P', Q') \in \mathcal{R}$, and
2. if $Q \overset{\alpha}{\longrightarrow} Q'$ then $P \overset{\alpha}{\Longrightarrow} P'$ and $(P', Q') \in \mathcal{R}$.

Processes P and Q are *weak bisimilar* (denoted $P \approx Q$) if and only if there exists a bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$. An example for weak bisimilar processes ($P \approx Q$) is provided in Figure 3.9. We find that there exists a relation where $\mathcal{R} = \{(P, Q), (P', Q'), (P''', Q')\}$.

Branching Bisimulation

Branching bisimulation [160] is another equivalence property similar to weak bisimulation which also does not give importance to internal actions. The difference between the equivalences is that in branching bisimulation the branching structure is also matched.

Definition 3.3 (Branching Bisimulation)**.** A relation $\mathcal{R}$ is a *branching bisimulation* if whenever $(P, Q) \in \mathcal{R}$ then, both

1. if $P \overset{\alpha}{\longrightarrow} P'$ then $Q \overset{\tau}{\Longrightarrow} Q' \overset{\alpha}{\longrightarrow} Q''$, $(P, Q') \in \mathcal{R}$ and $(P', Q'') \in \mathcal{R}$, and
2. if $Q \overset{\alpha}{\longrightarrow} Q'$ then $P \overset{\tau}{\Longrightarrow} P' \overset{\alpha}{\longrightarrow} P''$, $(Q, P') \in \mathcal{R}$ and $(Q', P'') \in \mathcal{R}$.

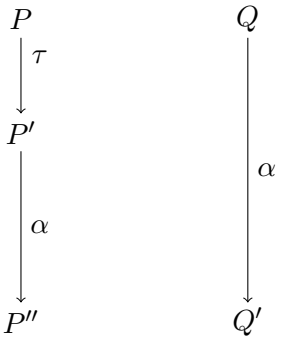


FIGURE 3.10: Branching Bisimulation

Processes P and Q are *branching bisimilar* (denoted $P \simeq Q$) if and only if there exists a bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

The relationship between branching bisimilar processes are shown in Figure 3.10. The addition of (P', Q) in $\mathcal{R}$ is the difference between branching bisimilarity and weak bisimilarity. Hence, there exists a relation where $\mathcal{R} = \{(P, Q), (P', Q), (P'', Q')\}$.

Congruence

Equivalence relations in this style are generically called *behavioural* equivalences. Suppose that $\cong$ is an equivalence relation on processes. The ideal situation is for $\cong$ to have a further property called *congruence*, which means that it is preserved by all of the constructs of the process calculus. A convenient way to express this property involves the notion of a *process context* $C[]$. This is a process term containing a *hole*, represented by $[]$, into which a process term may be placed. For example, $c?[x] \,.\, []$ is a context, and putting the process $d![x] \,.\, \mathbf{0}$ into the hole results in the process $c?[x] \,.\, d![x] \,.\, \mathbf{0}$.

Definition 3.4. An equivalence relation $\cong$ on processes is a *congruence* if $\forall P, Q.\ P \cong Q \Rightarrow \forall C[].\ C[P] \cong C[Q]$.

This definition of congruence corresponds to the idea that observers are themselves expressed as processes. Congruence, in addition to the property of being an equivalence relation, is what is required in order to allow equational reasoning about equivalence of processes. It means that if a system satisfies its specification, then it continues to satisfy its specification no matter what environment it is placed in.

The main advantage of the congruence property of equivalence is *composability*. This is a key property of process calculus based models that can help to manage complexity and provides scalable solutions in modelling. In particular, *composability* of the algebraic operators is widely used in process calculi based modelling of computer systems and is instrumental in ecological modeling [97]. For example, if P is a complicated process which is indistinguishable from the simple ideal behaviour process Q, and process R is

similarly indistinguishable from S, then P composed with R is indistinguishable from Q composed with S. In general, many processes do not compose but the importance of comparability should not be underestimated.

In the following chapters, we will be discussing the complete syntax and semantics of CQP.

Chapter 4

Theory and Applications of Communicating Quantum Processes (CQP)

This chapter provides an introduction to CQP and presents the use of CQP in the verification of quantum error correction. This study involves in applying the theory of behavioural equivalence in CQP defined in Davidson's Thesis [51]. Davidson in his Ph.D work has developed the theory of behavioural equivalence in CQP and used it in the verification of quantum protocols namely quantum teleportation and super dense coding. We use the operational semantics of CQP using labelled transition system as defined in [51] in order to describe two models of a three qubit error correcting code and quantum secret sharing protocol. We focus on quantum error correction, which helps us to begin a simple study on noise and also the error correction, and quantum secret sharing models has a simple high-level specification that is identical as that of the teleportation protocol described in [51]. With the help of the process equivalence (*probabilistic branching bismilarity*), we prove the correctness of the models by verifying with respect to their specifications.

4.1 Syntax and semantics of CQP

Simon Gay and Rajagopal Nagarajan designed the language CQP based on pi-calculus [125, 150], with the addition of primitive operations for quantum information processing. The general picture is that a system consists of a number of independent components, or *processes*, which can communicate by sending data along *channels*. In particular, qubits can be transmitted on channels. The complete description of the language is provided

$$
\begin{array}{rcl}
T & ::= & \mathsf{Int} \mid \mathsf{Qbit} \mid \hat{}[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v & ::= & 0 \mid 1 \mid \cdots \mid \mathsf{H} \mid \cdots \\
e & ::= & v \mid x \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} *= e^e \mid e + e \\
P & ::= & \mathbf{0} \mid (P|P) \mid P + P \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid [e].P \mid (\mathsf{qbit}\ x)P \mid \\
 & & (\mathsf{new}\ x : \hat{}[T])P
\end{array}
$$

FIGURE 4.1: Syntax of CQP.

in [73, 74]. One of the distinctive features of CQP is its type system, which ensures that operations can only be applied to data of the appropriate type. The type system is also used to enforce the view of qubits as physical resources, each of which has a unique owning process at any given time. If a qubit is send from A to B, then ownership is transferred and A can no longer access it.

Syntax

The syntax of CQP is defined by the grammar as shown in Figure 4.1. We use the notation $\tilde{e} = e_1, \ldots, e_n$, and write $|\tilde{e}|$ for the length of a tuple. The syntax of CQP which consists of types T, values v, expressions e (including quantum measurements and the conditional application of unitary operators $\tilde{e} *= e^e$), and processes P. The data types include integers of type Int, qubit of type Qbit, channel types $\hat{}[\tilde{T}]$, and n-qubit unitary operators types $\mathsf{Op}(n)$. Other data types can also be easily included which is evident in the later part of the thesis. Values v consist of variables (x,y,z etc), literal values of data types (0,1,..), unitary operators such as the Hadamard operator H.

Expressions e consist of values, measurements $\mathsf{measure}\ e_1, \ldots, e_n$, applications $e_1, \ldots, e_n *= e$ of unitary operators, and expressions involving data operators such as $e + e'$. Processes include the nil process $\mathbf{0}$, parallel composition $P|P$, inputs $e?[\widetilde{x}:\widetilde{T}].P$, outputs $e![\widetilde{e}].P$, actions $\{e\}.P$ (typically a unitary operation or measurement), typed channel restriction $(\mathsf{new}\ x : \hat{}[\widetilde{T}])P$, and qubit declaration $(\mathsf{qbit}\ x)P$. We use the notation $\tilde{x} : \tilde{T} = x_1 : T_1, \ldots, x_n : T_n$ in declaring the types of all input-bound variables.

In order to define the operational semantics we provide the *internal syntax* in Figure 4.2. Values are supplemented with qubit names q and channel names c. The qubit names are generated at run-time and substituted for the variables used in qbit declarations respectively. Evaluation contexts for expressions ($E[]$) and processes ($F[]$) are used to define the operational semantics [167]. The structure of $E[]$ is used to define call by value evaluation of expressions and the hole $[]$ provides the first part of the expression

$$
\begin{array}{lcl}
v & ::= & \cdots \mid q \mid c \\
E & ::= & [] \mid \mathsf{measure}\ E, \tilde{e} \mid \mathsf{measure}\ v, E, \tilde{e} \mid \cdots \mid \mathsf{measure}\ \tilde{v}, E \mid E + e \mid v + E \\
F & ::= & []?[\tilde{x}].P \mid []![\tilde{e}].P \mid v![[].\tilde{e}].P \mid v![v, [], \tilde{e}].P \mid \cdots \mid v![\tilde{v}, []].P \mid \{[]\}.P
\end{array}
$$

FIGURE 4.2: Internal syntax of CQP.

to be evaluated. The structure of $F[]$ is used to define the reduction of the processes by specifying the expressions within a process which needs to be evaluated.

Given a process P we define its free variables $fv(P)$, free qubit names $fq(P)$ and free channel names $fc(P)$ as usual; the binders (of x or $\tilde{x}$) are $y?[\tilde{x} : \tilde{T}]$, ($\mathsf{qbit}\ x$) and ($\mathsf{new}\ x : T$).

4.1.1 Operational Semantics

The first presentation of CQP [73, 74] defined the operational semantics based on *reductions* instead of labelled transitions. These correspond to τ transitions and were defined directly. The reduction semantics considers the use of closed quantum systems, i.e. there are no qubits outside the system. This is primarily motivated due to the inability to completely describe the state of a quantum subsystem. However, the reduction semantics allows the behaviour of a complete system to be defined but is not sufficient in describing the *potential* interactions of a process. The interpretations of these interactions are necessary in order to define the behavioural equivalence between the processes.

The next version of CQP was introduced in Davidson's Ph.D thesis [51] in order to consider behavioural equivalence, which defined the operational semantics of CQP using a labelled transition system. The main difference in reduction semantics and the labelled transition system is the consideration of external interactions of a process, that is the inclusion of the input and output transitions. Here, the quantum systems are considered to be open which allows the system to interact with its environment. We use the remaining definitions from [51] in our analysis for quantum error correction.

Configurations

In a quantum process calculus such as CQP, the execution of a system is not completely described by the process term (which is the case for classical process calculus) but also depends on the quantum state. Hence the operational semantics are defined using *configurations*, which represent both the quantum state and the process term.

Definition 4.1 (Configuration)**.** A configuration is defined as a tuple of the form $(\sigma; \omega; P)$ where σ is a mapping from qubit names to the quantum state and ω is a list of names associated with the process P.

We operate with configurations such as

$$([q_0, q_1 \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_1; c![q_1] \,.\, P). \tag{4.1}$$

For example, in this case, q_0 and q_1 represent the two qubits. This configuration means that the global quantum state consists of the qubits, q_0 and q_1, in the specified state; that the process term under consideration has access to qubit q_1 but not to q_0 and that the process itself is $c![q_1] \,.\, P$.

Now consider a configuration with the same quantum state but a different process term:

$$([q_0, q_1 \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_0; d![q_0] \,.\, Q).$$

The parallel composition of these configurations is the following:

$$([q_0, q_1 \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_0, q_1; c![q_1] \,.\, P \mid d![q_0] \,.\, Q)$$

where the quantum state is still the same.

The semantics of CQP consists of labelled transitions between configurations, which are defined in a similar way to classical process calculus. For example, configuration (4.1) has the transition

$$([q_0, q_1 \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q_1; c![q_1] \,.\, P) \xrightarrow{c![q_1]} ([q_0, q_1 \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; \emptyset; P).$$

The quantum state is not changed by this transition, but because qubit q_1 is output, the continuation process P no longer has access to it; the final configuration has an empty list of owned qubits. These configurations are also called *pure* configurations.

Another major difference between the *reduction* semantics [73, 74] and *LTS* semantics [51] of CQP is in the treatment of quantum measurement. In the reduction semantics of CQP, a measurement leads to a probability distribution over configurations, which at the next step reduces probabilistically to one particular configuration. But, the *LTS* semantics treats the analysis of measurement in a different manner. In *LTS* semantics, the measurement leads to a distribution of pure configurations (defined as *mixed configuration*) if the measurement result is communicated within the system. But, if the measurement result is communicated to the environment (i.e. outside the system) then

this leads to a probability distribution of configurations. The role of *mixed configurations* is important when considering the *congruence* property of the equivalence of processes.

Definition 4.2 (Mixed Configuration)**.** [51] A *mixed configuration* is a weighted distribution (denoted as $\oplus$) of pure configurations, written as $\oplus_{i\in I}\ g_i\ ([\widetilde{x} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{y} \bullet P; \widetilde{v_i})$ with weights g_i where $\sum_{i\in I} g_i = 1$ and for each $i \in I, 0 < g_i \leq 1$ and $|\psi_i\rangle \in \mathbb{H}$ (which is a two dimensional Hilbert space) and $|\widetilde{v_i}| = |\widetilde{y}|$.

The operator $\oplus$ (not to be confused with $\boxplus$ which represents the probabilistic distribution) represents a distribution over the set I with weights g_i. The process term is replaced by the expression $\lambda\widetilde{y} \bullet P$ that shows the components of the mixed configuration have the same process structure. The components differ with each other in respect to the values $\widetilde{v_i}$ that are substituted in the expression $\lambda\widetilde{y} \bullet P$, which is the reason for the λ notation. The variables $\widetilde{y}$ are placeholders in the expression. If the result of a quantum measurement is not made available to an observer then the system is considered to be in a mixed state, but it is not sufficient to simply write a mixed quantum state in a configuration. In general the mixture includes the process term, because the measurement result occurs within the term.

Example 4.1. $([q \mapsto \alpha_0|0\rangle + \alpha_1|1\rangle]; q; c![\mathsf{measure}\ q].P) \xrightarrow{\tau} \oplus_{i\in\{0,1\}}\ |\alpha_i|^2\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i)$.

This transition represents the effect of a measurement, within a process which is going to output the result of the measurement. But, the output, however, is not part of the transition. Hence, it is a τ transition and the process term on the right still contains $c![]$. The configuration on the left is a *pure configuration*, as described before. On the right we have a *mixed configuration* in which the $\oplus$ ranges over the possible outcomes of the measurement and the $|\alpha_i|^2$ are the weights of the components in the mixture. The quantum state $[q \mapsto |i\rangle]$ corresponds to the measurement outcome. The expression $\lambda x \bullet c![x].P$ represents the fact that the components of the mixed configuration have the same process structure and differ only in the values corresponding to measurement outcomes. The final term in the configuration, i, shows how the abstracted variable x should be instantiated in each component. Thus the λx represents a term into which expressions may be substituted. So the mixed configuration is essentially an abbreviation of

$$|\alpha_0|^2([q \mapsto |0\rangle]; q; c![0].P\{0/x\}) \oplus |\alpha_1|^2([q \mapsto |1\rangle]; q; c![1].P\{1/x\}).$$

If a measurement result is output to the environment, then the observer would know the possible state of the system. This is represented by probabilistic branching in which case the system is no longer a mixture of the two.

Definition 4.3 (Probabilistic Configuration)**.** A *probabilistic configuration* is a probability distribution of configurations, written as $\boxplus_{i\in I} p_i([\widetilde{x} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{y} \bullet P; \widetilde{v_i})$ with weights p_i where $\sum_{i\in I} p_i = 1$ and for each $i \in I, p_i > 0$ and $|\psi_i\rangle \in \mathbb{H}$ (i.e. a two dimensional Hilbert space) and $|\widetilde{v_i}| = |\widetilde{y}|$.

Example 4.2.

$$\oplus_{i\in\{0,1\}}\ |\alpha|_i^2\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i) \xrightarrow{c![\{0,1\}]} \boxplus_{i\in\{0,1\}} |\alpha_i|^2([q \mapsto |i\rangle]; q; \lambda x \bullet P; i)$$

$$\overset{|\alpha_0|^2}{\rightsquigarrow}\ ([q \mapsto |0\rangle]; q; \lambda x \bullet P; 0)$$

Example 4.2 shows the effect of the output from the final configuration of Example 4.1. The output transition produces the intermediate configuration called the probabilistic configuration (in contrast to a mixed configuration; note the change from $\oplus$ to $\boxplus$). Because it comes from a mixed configuration, the output transition contains a *set* of possible values. From the intermediate configuration there are two possible probabilistic transitions, of which one is shown ($\overset{|\alpha_0|^2}{\rightsquigarrow}$).

Example 4.3.

$$\oplus_{i\in\{0,1\}}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x \bullet (c![x].P \mid c?[y].Q); i) \xrightarrow{\tau}$$

$$\oplus_{i\in\{0,1\}}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x \bullet (P \mid Q\{x/y\}); i)$$

The measurement results could be communicated internally. This would not create a probability distribution and the system would still be in a mixed configuration. In Example 4.3 there is a mixed configuration on the left, with arbitrary weights g_i, which we imagine to have been produced by a measurement. However, there is now a receiver for the output. Although there is no difference in process Q between the two components of the mixed configuration, we include it in the λ because the communication will propagate the different possible values for x to Q.

We now present the labelled transition rules of CQP and the type system which are discussed in detail in [51].

Expression Transition Rules

The semantics of expressions [51] is defined by the reduction relations $\longrightarrow_v$ (on values) and $\longrightarrow_e$ (on expressions), given in Figure 4.3. Rule R-Plus deal with the evaluation of terms that result in values. The rule introduces a variable x which is a placeholder for the value w. The placeholder plays an important part in mixed expression configuration. This is evident in R-Context where each component of the configuration

$$([\widetilde{q} \mapsto |\psi\rangle]; \omega; u + v) \longrightarrow_v ([\widetilde{q} \mapsto |\psi\rangle]; \omega; \lambda x \bullet x; w) \quad \text{where } w = u + v \qquad \text{(R-Plus)}$$

$$([q_0, \ldots, q_{n-1} \mapsto \alpha_0|\phi_0\rangle + \cdots + \alpha_{2^n-1}|\phi_{2^n-1}\rangle]; \omega; \mathsf{measure}\ q_0, \ldots, q_{r-1}) \longrightarrow_v$$
$$\oplus_{0 \le m < 2^r}\ g_m\ ([q_0, \ldots, q_{n-1} \mapsto \tfrac{\alpha_{l_m}}{\sqrt{g_m}}|\phi_{l_m}\rangle + \cdots + \tfrac{\alpha_{u_m}}{\sqrt{g_m}}|\phi_{u_m}\rangle]; \omega; \lambda x \bullet x; m)$$
$$\text{(R-Measure)}$$

$$\text{where } l_m = 2^{n-r}m, u_m = 2^{n-r}(m+1) - 1, g_m = |\alpha_{l_m}|^2 + \cdots + |\alpha_{u_m}|^2$$

$$([q_0, \ldots, q_{n-1} \mapsto |\phi\rangle]; \omega; q_0, \ldots, q_{r-1} *= U^m) \longrightarrow_v \qquad \text{(R-Trans)}$$
$$([q_0, \ldots, q_{n-1} \mapsto (U^m \otimes I_{n-r})|\phi\rangle]; \omega; \mathsf{unit}; \cdot)$$

$$\frac{\forall i \in I.([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; e\{\widetilde{u}_i/\widetilde{y}\}) \longrightarrow_v \oplus_{j \in J_i}\ g_{ij}\ ([\widetilde{q} \mapsto |\psi_{ij}\rangle]; \omega; \lambda\widetilde{x} \bullet e'\{\widetilde{u}_i/\widetilde{y}\}; \widetilde{v}_{ij})}{\oplus_{i \in I}\ h_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{y} \bullet E[e]; \widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i \in I \\ j \in J_i}}\ h_i g_{ij}\ ([\widetilde{q} \mapsto |\psi_{ij}\rangle]; \omega; \lambda\widetilde{y}\widetilde{x} \bullet E[e']; \widetilde{u}_i, \widetilde{v}_{ij})} \qquad \text{(R-Context)}$$

Figure 4.3: Transition rules for values and expressions. [51]

gives rise to a particular value. R-Measure is measurement rule which produces a mixed configuration over the possible measurement outcomes m. R-Trans deals with unitary transformations which result in literal unit. The important aspect of R-Trans and R-Measure is the effect they have on the quantum state. R-Context is used for the evaluation of expressions in an expression context E and also for the evaluation of the expressions in mixed configurations. The mixed expression configuration $\oplus_{i \in I}\ h_i\ (\sigma_i; \omega; \lambda\tilde{y}.E[e]; \tilde{u}_i)$ is evaluated by determining each individual component of the mixed configuration.

Pure Configuration Transition Rules

The transition rules for pure process configurations [51] are given in Figure 4.4. This defines the input and output transitions for pure configurations. The rules P-Par and P-Res are needed to define input and output actions for arbitrary processes.

Now we define some notation. There are two types of transition: probabilistic transitions which take the form $\boxplus_i p_i s_i \overset{p_i}{\rightsquigarrow} s_i$ where $\forall i.(p_i < 1)$, and non-deterministic transitions which have the general form $s \xrightarrow{\alpha} \boxplus_i p_i s_i$ where $\forall i.(p_i \le 1)$ and α is an *action*. The notation $\boxplus_i p_i s_i \equiv p_1 \bullet s_1 \boxplus \cdots \boxplus\ p_n \bullet s_n$ denotes a probability distribution over configurations in which $\sum_i p_i = 1$. If there is only a single configuration (with probability 1) we omit the probability, for example $s \xrightarrow{\alpha} s'$.

Mixed Configuration Transition Rules

The transition rules for mixed configurations are defined in Figure 4.5. The rule L-Prob is a probabilistic transition in which p_i is the probability of the transition. The rules L-In and L-Out represent the input and output actions respectively, which are the visible interactions with the environment. When the two processes of input and output

$$([\widetilde{pqr} \mapsto |\psi\rangle]; \widetilde{p}, \widetilde{q}; c![\widetilde{v}, \widetilde{q}].P) \xrightarrow{c![\widetilde{v},\widetilde{q}]}_p ([\widetilde{pqr} \mapsto |\psi\rangle]; \widetilde{p}; P) \quad \text{(P-Out)}$$

$$([\widetilde{q} \mapsto |\psi\rangle]; \omega; c?[\widetilde{y}].P) \xrightarrow{c?[\widetilde{v},\widetilde{r}]}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega, \widetilde{r}; P\{\widetilde{v}, \widetilde{r}/\widetilde{y}\}) \quad \text{(P-In)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P \mid Q) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P' \mid Q)} \quad \text{(P-Par)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P + Q) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega'; P')} \quad \text{(P-Sum)}$$

$$\frac{([\widetilde{q} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega; P')}{([\widetilde{q} \mapsto |\psi\rangle]; \omega; (\mathsf{new}\ c)P) \xrightarrow{\alpha}_p ([\widetilde{q} \mapsto |\psi\rangle]; \omega; (\mathsf{new}\ c)P')} \quad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \quad \text{(P-Res)}$$

Figure 4.4: Transition rules for pure process configurations. [51]

actions are put in parallel then each has a partner for its potential interaction, and the input and output can synchronise, resulting in a τ transition which is given by the rule L-Com. The rule L-Act converts the action expression to a value, which can be removed. This is a reduction which involves effects like measurement or transformation of the quantum state. Rule L-Qbit introduces additional Qbit variable. The rule L-Out is the output rule which combines mixed configurations along with probabilistic branching. The branching happens only when there is an information to differentiate the components. Normally, the information are classical values that are given as outputs and these can vary between the components.

4.1.2 Type System

In this section we introduce the type system for the LTS semantics of CQP that is presented in detail in [51], which is similar to the type system has been originally defined for CQP using reduction semantics [74, 75]. The important contribution of the type system is that it gives an assurance that each qubit is owned by a unique process and cannot be duplicated. Hence, the well-typed processes respect the no-cloning principle and the treatment of qubits as physical resources. For the analysis of executing processes, it is necessary that the types be preserved, which is one of the main results in [51].

The typing rules for the syntax defined in Figure 4.1 are shown in Figure 4.6. Environments Γ are mappings from variables to types in the usual way. There are two kinds of typing judgements: $\Gamma \vdash e : T$ means that an expression e has type T in the environment Γ, and $\Gamma \vdash P$ means that a process P is well-typed in the environment Γ.

$$\boxplus_j p_j(\oplus_i\ g_i\ (\sigma_i;\omega;P_i)) \overset{p_i}{\rightsquigarrow} \oplus_i\ g_i\ (\sigma_i;\omega;P_i) \qquad \text{(L-Prob)}$$

$$\oplus_i\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet c?[\widetilde{y}].P;\widetilde{v}_i) \xrightarrow{c?[\widetilde{u},\widetilde{r}]} \oplus_i\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}\bullet P\{\widetilde{u},r/\widetilde{y}\};\widetilde{v}_i) \text{ where } |\widetilde{u}|+|\widetilde{r}|=|\widetilde{y}| \qquad \text{(L-In)}$$

$$\frac{\forall i\in I.([\widetilde{p}\widetilde{q}\mapsto|\psi_i\rangle];\widetilde{p};P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i,\widetilde{r}]}_p ([\widetilde{p}\widetilde{q}\mapsto|\psi_i\rangle];\widetilde{p}';P'\{\widetilde{v}_i/\widetilde{x}\})}{\oplus_{i\in I}\ g_i\ ([\widetilde{p}\widetilde{q}\mapsto|\psi_i\rangle];\widetilde{p};\lambda\widetilde{x}\bullet P;\widetilde{v}_i) \xrightarrow{c![U,\widetilde{r}]} \boxplus_{j\in J}p_j(\oplus_{i\in I_j}\ \frac{g_i}{p_j}\ ([\widetilde{p}'\widetilde{r}\widetilde{q}\mapsto\Pi|\psi_i\rangle];\widetilde{p}';\lambda\widetilde{x}\bullet P';\widetilde{v}_i))} \qquad \text{(L-Out)}$$

where $U=\{\widetilde{u}_i\mid i\in I\}=\{\widetilde{u}_{k_j}\mid j\in J\}$ and $\forall j\in J, I_j=\{i|\widetilde{u}_i=\widetilde{u}_{k_j}\}, p_j=\sum_{i\in I_j} g_i$

and $\widetilde{r}\subseteq\widetilde{p},\widetilde{p}'=\widetilde{p}\setminus\widetilde{r},\Pi$ corresponds to the permutation $\pi:\widetilde{p}\widetilde{q}\mapsto\widetilde{p}'\widetilde{r}\widetilde{q}$.

$$\frac{\forall i\in I.(\sigma_i;\omega,\widetilde{r};P\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c![\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i;\omega;P'\{\widetilde{v}_i/\widetilde{x}\}) \quad \forall i\in I.(\sigma_i;\omega;Q\{\widetilde{v}_i/\widetilde{x}\}) \xrightarrow{c?[\widetilde{u}_i,\widetilde{r}]}_p (\sigma_i;\omega,\widetilde{r};Q'\{\widetilde{v}_i/\widetilde{x}\})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}\bullet P\mid Q;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\sigma_i;\omega,\widetilde{r};\lambda\widetilde{x}\bullet P'\mid Q';\widetilde{v}_i)} \qquad \text{(L-Com)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P\mid Q;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet P'\mid Q;\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Par)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P+Q;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet P';\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Sum)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet P';\widetilde{v}_i\widetilde{w}_{ij})}{\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet (\mathsf{new}\ c)P;\widetilde{v}_i) \xrightarrow{\alpha} \oplus_{\substack{i\in I\\ j\in J_i}}\ g_ih_{ij}\ (\sigma_{ij};\omega';\lambda\widetilde{x}\widetilde{y}\bullet (\mathsf{new}\ c)P';\widetilde{v}_i\widetilde{w}_{ij})} \qquad \text{(L-Res)}$$

$$\text{if } \alpha\notin\{c?[\cdot],c![\cdot]\}$$

$$\oplus_{i\in I}\ g_i\ ([\widetilde{q}\mapsto|\psi_i\rangle];\omega;\lambda\widetilde{x}\bullet(\mathsf{qbit}\ y)P;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ ([\widetilde{q},q\mapsto|\psi_i\rangle|0\rangle];\omega,q;\lambda\widetilde{x}\bullet P\{q/y\};\widetilde{v}_i) \text{ where } q \text{ is fresh} \qquad \text{(L-Qbit)}$$

$$\oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet\{u\}.P_i;\widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\sigma_i;\omega;\lambda\widetilde{x}\bullet P;\widetilde{v}_i) \qquad \text{(L-Act)}$$

$$\frac{\oplus_{i\in I}\ h_i\ (\sigma_i;\omega;\lambda\widetilde{y}\bullet e;\widetilde{u}_i) \longrightarrow_e \oplus_{\substack{i\in I\\ j\in J_i}}\ h_ig_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}\widetilde{x}\bullet e';\widetilde{u}_i\widetilde{v}_{ij})}{\oplus_{i\in I}\ h_i\ (\sigma_i;\omega;\lambda\widetilde{y}\bullet F[e];\widetilde{u}_i) \xrightarrow{\tau} \oplus_{\substack{i\in I\\ j\in J_i}}\ h_ig_{ij}\ (\sigma_{ij};\omega;\lambda\widetilde{y}\widetilde{x}\bullet F[e'];\widetilde{u}_i\widetilde{v}_{ij})} \qquad \text{(L-Expr)}$$

Figure 4.5: Transition rules for mixed process configurations. [51]

The treatment of qubits in they type system is the key to ensuring that the no-cloning principle is obeyed. It is ensured that in rules T-Msure, T-Out and T-Trans the qubit variables are distinct which prevents qubits being cloned at a local level. With the use of the + operation on environments (Definition 4.4) in rule T-Par, we ensure the unique ownership of qubits amongst parallel components.

Definition 4.4 (Addition of Environments)**.** [51, 75] The partial operation of adding a typed variable to an environment, $\Gamma + x{:}T$, is defined by

$$\Gamma + x{:}T = \begin{cases} \Gamma, x{:}T & \text{if } x \notin dom(\Gamma) \\ \Gamma & \text{if } T \neq \mathsf{Qbit} \text{ and } x{:}T \in \Gamma \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This operation is extended inductively to a partial operation $\Gamma + \Delta$ on environments. The soundness of the type system are proved in [51].

4.2 Equivalence in quantum process calculus

In the previous section, we have discussed the labelled transition semantics of CQP, which helps us to describe the interactions within and outside a quantum system. Based on these semantics, we will now consider behavioural equivalence of quantum processes, which is important in proving the correctness of a system. We have introduced bisimulation in our previous chapter and we shall now extend this concept to quantum systems.

Bisimulation is a binary relation, which associates two systems to match each other's actions or simulate one another. This means that an observer cannot distinguish each of the systems from each other. Actions can either be internal or external. The internal actions are generally labelled as τ and is straightforward to match for a quantum system. But, for external actions which is either input or output, it is not quite straightforward as the system depends not only depends on the process but also on the quantum state of the system. Hence, in order to capture the behaviour of the system, one must consider matching the qubits associated with the system or the quantum state or both.

One of the characteristics of strong bisimilarity is that it is a stronger relation than trace equivalence; it is possible for two processes to generate the same sequences of labels, but not be strong bisimilar. Strong bisimilarity depends on the branching structure of the processes as well as on their sequences of labels. Another characteristic is that *every* transition must be matched exactly, including τ transitions. However, because they arise from internal communications, it is often undesirable to insist that equivalent

$$\Gamma \vdash v : \mathsf{Int} \quad \text{if } v \text{ is an integer literal} \qquad \text{(T-IntLit)}$$

$$\Gamma \vdash \mathsf{unit} : \mathsf{Unit} \qquad \text{(T-Unit)}$$

$$\Gamma \vdash \mathsf{H} : \mathsf{Op}(2) \quad \text{etc.} \qquad \text{(T-Op)}$$

$$\Gamma, x : T \vdash x : T \qquad \text{(T-Var)}$$

$$\frac{\forall i (\Gamma \vdash x_i : \mathsf{Qbit}) \quad x_1, \ldots, x_n \text{ distinct}}{\Gamma \vdash \mathsf{measure}\ x_1, \ldots, x_n : \mathsf{Int}} \qquad \text{(T-Msure)}$$

$$\frac{\Gamma \vdash e : \mathsf{Int} \quad \Gamma \vdash e' : \mathsf{Int}}{\Gamma \vdash e + e' : \mathsf{Int}} \qquad \text{(T-Plus)}$$

$$\Gamma \vdash \mathbf{0} \qquad \text{(T-Nil)}$$

$$\frac{\Gamma, x : \mathsf{Qbit} \vdash P}{\Gamma \vdash (\mathsf{qbit}\ x)P} \qquad \text{(T-Qbit)}$$

$$\frac{\Gamma_1 \vdash P \quad \Gamma_2 \vdash Q \quad \Gamma_1 + \Gamma_2 \text{ defined}}{\Gamma_1 + \Gamma_2 \vdash P \mid Q} \qquad \text{(T-Par)}$$

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P + Q} \qquad \text{(T-Sum)}$$

$$\frac{\Gamma \vdash x : \hat{}[T_1, \ldots, T_n] \quad \Gamma, y_1 : T_1, \ldots, y_n : T_n \vdash P}{\Gamma \vdash x?[y_1 : T_1, \ldots, y_n : T_n].P} \qquad \text{(T-In)}$$

$$\frac{\begin{array}{c}\Gamma \vdash x : \hat{}[T_1, \ldots, T_m, \mathsf{Qbit}, \ldots, \mathsf{Qbit}] \quad \forall i.(T_i \neq \mathsf{Qbit}) \\ \forall i.(\Gamma \vdash e_i : T_i) \quad y_i \text{ distinct} \quad \Gamma \vdash P\end{array}}{\Gamma, y_1 : \mathsf{Qbit} \ldots, y_n : \mathsf{Qbit} \vdash x![e_1, \ldots, e_m, y_1, \ldots, y_n].P} \qquad \text{(T-Out)}$$

$$\frac{\Gamma, x : \hat{}[T_1, \ldots, T_n] \vdash P}{\Gamma \vdash (\mathsf{new}\ x : \hat{}[T_1, \ldots, T_n])P} \qquad \text{(T-New)}$$

$$\frac{\Gamma \vdash e : T \quad \Gamma \vdash P}{\Gamma \vdash \{e\}.P} \qquad \text{(T-Act)}$$

$$\frac{\forall i (\Gamma \vdash x_i : \mathsf{Qbit}) \quad x_1 \ldots x_n \text{ distinct} \quad \Gamma \vdash U : \mathsf{Op}(n) \quad \Gamma \vdash e : \mathsf{Int} \quad \Gamma \vdash P}{\Gamma \vdash x_1, \ldots, x_n *= U^e : \mathsf{Unit}} \qquad \text{(T-Trans)}$$

FIGURE 4.6: Typing rules. [51]

processes must match each other's τ transitions. Hence weaker variations of bisimilarity have been defined, including *weak bisimilarity* [124], which ignores τ transitions, and *branching bisimilarity* [160], which reduces the significance of τ transitions but retains information about their branching structure.

Lalire [110] defined a probabilistic branching bisimilarity for the process calculus QPAlg (Quantum Process Algebra). This is based on the branching bisimilarity of van Glabbeek and Weijland [160], which identifies quantum processes associated with graphs having the same branching structure. However, the bisimulation was not preserved by parallel composition and hence not congruent. Feng *et al.* [65] developed qCCS and defined strong and weak probabilistic bisimilarity. Their equivalences are preserved by parallel

composition with processes that do not change the quantum context. A later version of qCCS [170] excluded classical information and introduced the notion of approximate bisimilarity as a way of quantifying differences in purely quantum process behaviour. In the latest version of qCCS, Feng *et al.* [66] prove that weak bisimilarity is a congruence. They apply their result to quantum teleportation and superdense coding. Kubota *et al.* [108] has described BB84 using qCCS and proved that it is equivalent to an EDP (entanglement distillation protocol)-based protocol using the property of bisimulation in qCCS.

4.2.1 Probabilistic branching bisimulation in CQP

Davidson defined an equivalence of CQP [51] based on probabilistic branching bisimilarity [10] which combines the notion of branching bisimulation along with probabilistic transitions. This is similar to the equivalence defined for QPAlg [110] with a difference in the treatment of non deterministic actions. In QPAlg [110] non deterministic branching happens with equal probability which is a drawback as this is not preserved by parallel composition. Davidson in his definition differentiates non deterministic and probabilistic branching by using a function that is preserver in parallel composition. This is based on the bisimulation [158] which assigns a probability 1 to all non-deterministic transitions. The separation of probabilistic and non-deterministic transitions avoids the need to consider non-deterministic and probabilistic transitions from the same configuration. Another important point is that when considering matching of input or output transitions involving qubits, it is the reduced density matrices of the transmitted qubits that are required to be equal. The definitions in the remainder of this section are from [51].

The relations $\xrightarrow{\alpha}$ and $\overset{\pi}{\rightsquigarrow}$ induce a partition of $\mathcal{S}$ (a set of all configurations) into non-deterministic configurations $\mathcal{S}_n$ and probabilistic configurations $\mathcal{S}_p$: let $\mathcal{S}_p = \{s \in \mathcal{S} \mid \exists \pi \in (0,1), \exists t \in \mathcal{S}, s \overset{\pi}{\rightsquigarrow} t\}$; and let $\mathcal{S}_n = \mathcal{S} \setminus \mathcal{S}_p$. By this definition a configuration with no transitions belongs to $\mathcal{S}_n$.

Definition 4.5 (Density Matrix of Configurations [51])**.** Let $\sigma_i = [\widetilde{p} \mapsto |\psi_i\rangle]$ and $\widetilde{q} \subseteq \widetilde{p}$ and $s_i = (\sigma_i; \omega; \lambda \widetilde{x} \bullet P; \widetilde{v}_i)$ and $s = \oplus_i \; g_i \; s_i$. Then

1. $\rho(\sigma_i) = |\psi_i\rangle\langle\psi_i|$
2. $\rho^{\widetilde{q}}(\sigma_i) = \mathrm{tr}_{\widetilde{p}\setminus\widetilde{q}}(|\psi_i\rangle\langle\psi_i|)$
3. $\rho(s_i) = \rho(\sigma_i)$
4. $\rho^{\widetilde{q}}(s_i) = \rho^{\widetilde{q}}(\sigma_i)$
5. $\rho(s) = \sum_i g_i \rho(s_i)$
6. $\rho^{\widetilde{q}}(s) = \sum_i g_i \rho^{\widetilde{q}}(s_i)$

Here, the notation ρ_E denotes the reduced density matrix of the *environment* qubits. Formally, if $s = ([\widetilde{q} \mapsto |\psi\rangle]; \widetilde{p}; P)$ then $\rho_E(s) = \rho^{\widetilde{r}}(s)$ where $\widetilde{r} = \widetilde{q} \setminus \widetilde{p}$. The definition of ρ_E is extended to mixed configurations in the same manner as ρ.

The probabilistic function $\mu : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is defined in the style of [158]. This allows the possibility of treating non-deterministic transitions as transitions with probability 1, which is necessary when calculating the total probability of reaching a terminal state. $\mu(s, t) = \pi$ if $s \overset{\pi}{\rightsquigarrow} t$; $\mu(s, t) = 1$ if $s = t$ and $s \in \mathcal{S}_n$; $\mu(s, t) = 0$ otherwise.

Let $\xrightarrow{\tau}{}^{+}$ denote zero or one τ transitions; let $\Longrightarrow$ denote zero or more τ transitions; and let $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. We write $\widetilde{q}$ for a list of qubit names, and similarly for other lists.

Definition 4.6 (Probabilistic Branching Bisimulation [51])**.** An equivalence relation $\mathcal{R}$ on configurations is a *probabilistic branching bisimulation* on configurations if whenever $(s, t) \in \mathcal{R}$ the following conditions are satisfied.

I. If $s \in \mathcal{S}_n$ and $s \xrightarrow{\tau} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ with $(s, t') \in \mathcal{R}$ and $(s', t'') \in \mathcal{R}$.

II. If $s \xrightarrow{c![V,\widetilde{q}_1]} s'$ where $s' = \boxplus_{j\in\{1\ldots m\}} p_j s'_j$ and $V = \{\widetilde{v}_1, \ldots, \widetilde{v}_m\}$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c![V,\widetilde{q}_2]} t''$ with

   a) $(s, t') \in \mathcal{R}$,

   b) $t'' = \boxplus_{j\in\{1\ldots m\}} p_j t''_j$,

   c) for each $j \in \{1, \ldots, m\}$, $\rho_E(s'_j) = \rho_E(t''_j)$.

   d) for each $j \in \{1, \ldots, m\}$, $(s'_j, t''_j) \in \mathcal{R}$.

III. If $s \xrightarrow{c?[\widetilde{v}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c?[\widetilde{v}]} t''$ with $(s, t') \in \mathcal{R}$ and $(s', t'') \in \mathcal{R}$.

IV. If $s \in \mathcal{S}_p$ then $\mu(s, D) = \mu(t, D)$ for all classes $D \in \mathcal{S}/\mathcal{R}$.

This relation follows the standard definition of branching bisimulation [160] with additional conditions for probabilistic configurations and matching quantum information. In condition II we require that the distinct set of values V must match and although the qubit names ($\widetilde{q}_1$ and $\widetilde{q}_2$) need not be identical, their respective reduced density matrices ($\rho^{\widetilde{q}_1}(s)$ and $\rho^{\widetilde{q}_2}(t')$) must.

Following the approach of [158], we have Condition IV that provides the matching on probabilistic configurations . In this relation, a probabilistic configuration which necessarily evolves from an output will satisfy IV if the prior configuration satisfies II d). It is important to have Condition IV as it ensures that the probabilities are paired with their respective configurations, which thereby leads to the following definition of bisimilarity on configurations.

Definition 4.7 (Probabilistic Branching Bisimilarity [51])**.** Configurations s and t are *probabilistic branching bisimilar*, denoted $s \leftrightarroweq t$, if there exists a probabilistic branching bisimulation $\mathcal{R}$ such that $(s,t) \in \mathcal{R}$.

Since, we require equivalence of processes, independently of configurations (i.e. independently of particular quantum states), we get:

Definition 4.8 (Probabilistic Branching Bisimilarity of Processes [51])**.** Processes P and Q are *probabilistic branching bisimilar*, denoted $P \leftrightarroweq Q$, if and only if for all σ, $(\sigma; \emptyset; P) \leftrightarroweq (\sigma; \emptyset; Q)$.

For convenience, in the remainder of this thesis we refer *bisimilarity* as probabilistic branching bisimilarity and it will be clear from the context whether this is the relation on processes or configurations. The same symbol, $\leftrightarroweq$, is used for both relations.

We now consider the preservation properties of bisimilarity on processes. The first main result of [51] is that bisimilarity is a *non-input, non-qubit congruence* (Theorem 4.18). The key to this result is that the bisimilarity is preserved by parallel composition (Theorem 4.13) and is also shown for qCCS independently by [66]. The important *congruence* property of equivalence helps to realise that the equivalent processes remain equivalent in any context.

Before continuing, we use the formal definitions of *contexts* and *congruence*, and their *non-input, non-qubit* variants presented in [51]. The reason for considering variants without input and qubit declaration prefixes, is that substitution must also be considered when these are included.

Definition 4.9 (Context [51])**.** A *context* C is a process where occurrence of $\mathbf{0}$ replaced by a hole, $[\cdot]$. Formally,

$$C ::= [] \mid (C \mid P) \mid \alpha.C + P \mid \alpha.C \mid (\mathsf{new}\ x\widehat{\ }[T])C$$

for $\alpha \in \{e?[\tilde{x} : \tilde{T}], e![\tilde{e}], \{e\}, (\mathsf{qbit}\ x)\}$.

Definition 4.10 (Congruence [51])**.** An equivalence relation $\mathcal{R}$ on processes is a *congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P,Q) \in \mathcal{R}$ and C is a context.

Definition 4.11 (Non-input, non-qubit context [51])**.** A *non-input, non-qubit context* is a context in which the hole does not appear under an input or qubit declaration.

Definition 4.12 (Non-input, non-qubit congruence [51])**.** An equivalence relation $\mathcal{R}$ on processes is a *non-input, non-qubit congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P,Q) \in \mathcal{R}$ and C is a non-input, non-qubit context.

Theorem 4.13 (Parallel preservation for configurations [51]). *Assume that* $\Gamma \vdash P$, $\Gamma \vdash Q$, $\Gamma \vdash P \mid R$, *and* $\Gamma \vdash Q \mid R$. *If* $(\sigma; \emptyset; P) \leftrightarrows (\sigma; \emptyset; Q)$ *then* $(\sigma; \emptyset; P \mid R) \leftrightarrows (\sigma; \emptyset; Q \mid R)$.

Theorem 4.14 (Parallel Preservation [51]). *If* $P \leftrightarrows Q$ *then for any process* R *such that* $\Gamma \vdash P \mid R$ *and* $\Gamma \vdash Q \mid R$ *then* $P \mid R \leftrightarrows Q \mid R$.

Theorem 4.15 (Probabilistic branching bisimilarity is a non-input, non-qubit congruence [51]). *If* $P \leftrightarrows Q$ *and for any non-input, non-qubit context* C *if* $\Gamma \vdash C[P]$ *and* $\Gamma \vdash C[Q]$ *then* $C[P] \leftrightarrows C[Q]$.

It turns out that probabilistic branching bisimilarity is not a congruence because it is not preserved by substitution of values for variables, which is significant because of the use of substitution to define the semantics of input. We therefore define a stronger relation, *full probabilistic branching bisimilarity*, which is the closure of probabilistic branching bisimilarity under substitutions.

Definition 4.16 (Full probabilistic branching bisimilarity [51]). Processes P and Q are *full probabilistic branching bisimilar*, denoted $P \leftrightarrows^c Q$, if for all substitutions κ and all quantum states σ, $(\sigma; \widetilde{q}; P\kappa) \leftrightarrows (\sigma; \widetilde{q}; Q\kappa)$.

We are now able to state the main result of [51].

Theorem 4.17 (Full probabilistic branching bisimilarity is a congruence [51]). *If* $P \leftrightarrows^c Q$ *then for any context* $C[]$, *if* $C[P]$ *and* $C[Q]$ *are typable then* $C[P] \leftrightarrows^c C[Q]$.

The condition that $C[P]$ and $C[Q]$ are typable is used to ensure that the context does not manipulate qubits that are owned by P or Q.

4.3 Applications

In this section, we demonstrate the verification of quantum error correction and quantum secret sharing by applying the theory of behavioural equivalence of CQP. Verification of quantum protocols like quantum teleportation and superdense coding are demonstrated in [51]. We will present two models of three qubit flip error correction and a model of quantum secret sharing in CQP and formally define a specification process for each of the model, that is a high-level abstraction of the model. Proving that they are equivalent or bisimilar to their respective specification processes achieves verification of these models. By showing that each process is bismilar to its specification, we find that these processes are equivalent to one another.

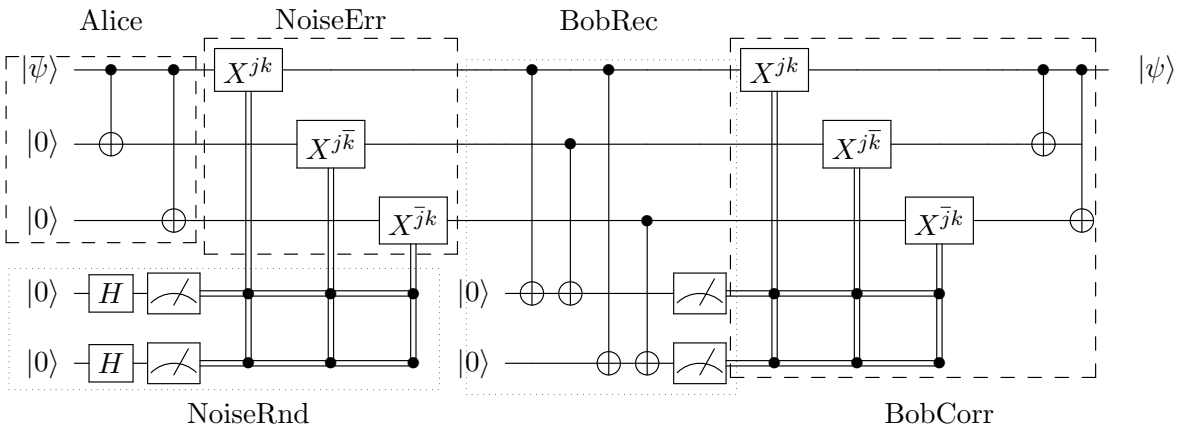


FIGURE 4.7: QECC

4.3.1 Error Correction - First Model

Our model of a quantum error correction system as shown in Figure 4.7 consists of three processes: *Alice*, *Bob* and *Noise*. *Alice* wants to send a qubit to *Bob* over a noisy channel, represented by *Noise*. She uses a simple error correcting code based on threefold repetition [132, Chapter 10]. This code is able to correct a single bit-flip error in each block of three transmitted qubits, so for the purpose of this example, in each block of three qubits, *Noise* either applies X to one of them or does nothing. *Bob* uses the appropriate decoding procedure to recover *Alice*'s original qubit. The CQP definition of *Alice* is as follows.

$$\begin{aligned} \mathit{Alice}(a{:}\widehat{\ }[\mathsf{Qbit}], b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) \quad = \quad & (\mathsf{qbit}\ y, z)a?[x{:}\mathsf{Qbit}]\,.\,\{x, z *{=}\ \mathsf{CNot}\}\,. \\ & \{x, y *{=}\ \mathsf{CNot}\}\,.\,b![x, y, z]\,.\,\mathbf{0} \end{aligned}$$

Alice is parameterized by two channels, a and b. In order to give *Alice* a general definition independent of the qubit to be sent to *Bob*, she will receive the qubit on channel a. The type of a is $\widehat{\ }[\mathsf{Qbit}]$, which is the type of a channel on which each message is a qubit. Channel b is where *Alice* sends the encoded qubits. Each message on b consists of three qubits, as indicated by the type $\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]$.

The right hand side of the definition specifies *Alice*'s behaviour. The first term, $(\mathsf{qbit}\ y, z)$, allocates two fresh qubits, each in state $|0\rangle$, and gives them the local names y and z. Then follows a sequence of terms separated by dots. This indicates temporal sequencing, from left to right. $a?[x{:}\mathsf{Qbit}]$ specifies that a qubit is received from channel a and given the local name x. The term $\{x, z *{=}\ \mathsf{CNot}\}$ specifies that the CNot operation is applied to qubits x and z; the next term is similar. These operations implement the threefold repetition code: if the intial state of x is $|0\rangle$ (respectively, $|1\rangle$) then the state of x, y, z becomes $|000\rangle$ (respectively, $|111\rangle$). In general, of course, the initial state of x may be a superposition, and then so will be the final state of x, y, z. Finally, the term $b![x, y, z]$

means that the qubits x, y, z are sent as a message on channel b. The term $\mathbf{0}$ simply indicates termination.

We model a noisy quantum channel by the process *Noise*, which receives three qubits from channel b (connected to *Alice*) and sends three (possibly corrupted) qubits on channel c (connected to *Bob*). *Noise* has four possible actions: do nothing, or apply X to one of the three qubits. These actions are chosen with equal probability. We produce probabilistic behaviour by introducing fresh qubits in state $|0\rangle$, applying H to put them into state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and then measuring in the standard basis. The definition of *Noise* is split into two sub-processes, of which the first, *NoiseRnd*, produces two random classical bits and sends them to the second, *NoiseErr*, on channel p. This programming style, using internal messages instead of assignment to variables, is typical of pi-calculus.

$$\mathit{NoiseRnd}(p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit}]) \quad = \quad (\mathsf{qbit}\ u, v)\{u *{=}\ \mathsf{H}\}\,.\,\{v *{=}\ \mathsf{H}\}\,.\,p![\mathsf{measure}\ u, \mathsf{measure}\ v]\,.\,\mathbf{0}$$

The process *NoiseErr* receives three qubits from channel b, and two classical bits from channel p. It interprets the classical bits, locally named j and k, as instructions for corrupting the qubits. This uses appropriate Boolean combinations of j and k to construct conditional quantum operations such as $\mathsf{X}^{j\bar{k}}$.

$$\begin{aligned}\mathit{NoiseErr}(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit}], c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = b?[x{:}\mathsf{Qbit}, y{:}\mathsf{Qbit}, z{:}\mathsf{Qbit}]\,.\\ p?[j{:}\mathsf{bit}, k{:}\mathsf{bit}]\,.\,\{x *{=}\ \mathsf{X}^{jk}\}\,.\,\{y *{=}\ \mathsf{X}^{j\bar{k}}\}\,.\,\{z *{=}\ \mathsf{X}^{\bar{j}k}\}\,.\,c![x,y,z]\,.\,\mathbf{0}\end{aligned}$$

The complete *Noise* process consists of *NoiseRnd* and *NoiseErr* in parallel, indicated by the vertical bar. Channel p is designated as a private local channel; this is specified by $(\mathsf{new}\ p)$. This construct comes from pi-calculus, where it can be used to dynamically create fresh channels, but here we are using it in the style of older process calculi such as CCS, to indicate a channel with restricted scope. Putting *NoiseRnd* and *NoiseErr* in parallel means that the output on p in *NoiseRnd* synchronizes with the input on p in *NoiseErr*, so that data is transferred.

$$\mathit{Noise}(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = (\mathsf{new}\ p)(\mathit{NoiseRnd}(p) \mid \mathit{NoiseErr}(b,p,c))$$

Bob consists of *BobRec* and *BobCorr*, where *BobRec* receives the qubits and measures the error syndrome, and *BobCorr* applies the appropriate correction. An internal channel p is used to transmit the result of the measurement, as well as the original qubits, again in pi-calculus style. After correcting the error in the group of three qubits, *BobCorr* reconstructs a quantum state in which qubit x has the original state received by *Alice*

and is separable from the auxiliary qubits. Finally, *BobCorr* outputs x on channel d.

$$\begin{gathered}BobRec(c{:}\widehat{}[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], p{:}\widehat{}[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit},\mathsf{bit},\mathsf{bit}]) = (\mathsf{qbit}\ s,t)\,.\\ c?[x{:}\mathsf{Qbit}, y{:}\mathsf{Qbit}, z{:}\mathsf{Qbit}]\,.\,\{x, s *{=}\,\mathsf{CNot}\}\,.\,\{y, s *{=}\,\mathsf{CNot}\}\,.\,\{x, t *{=}\,\mathsf{CNot}\}\,.\\ \{z, t *{=}\,\mathsf{CNot}\}\,.\,p![x, y, z, \mathsf{measure}\ s, \mathsf{measure}\ t]\,.\,\mathbf{0}\end{gathered}$$

$$\begin{gathered}BobCorr(p{:}\widehat{}[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit},\mathsf{bit},\mathsf{bit}], d{:}\widehat{}[\mathsf{Qbit}]) = p?[x{:}\mathsf{Qbit}, y{:}\mathsf{Qbit}, z{:}\mathsf{Qbit}, j{:}\mathsf{bit}, k{:}\mathsf{bit}]\,.\\ \{x *{=}\,\mathsf{X}^{jk}\}\,.\,\{y *{=}\,\mathsf{X}^{j\bar{k}}\}\,.\,\{z *{=}\,\mathsf{X}^{\bar{j}k}\}\,.\,\{x, y *{=}\,\mathsf{CNot}\}\,.\,\{x, z *{=}\,\mathsf{CNot}\}\,.\,d![x]\,.\,\mathbf{0}\end{gathered}$$

$$Bob(c{:}\widehat{}[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], d{:}\widehat{}[\mathsf{Qbit}]) = (\mathsf{new}\ p)(BobRec(c,p) \mid BobCorr(p,d))$$

The overall effect of the error correcting system is to input a qubit from channel a and output a qubit, in the same state, on channel d, in the presence of noise. The complete system is defined as follows.

$$QECC(a{:}\widehat{}[\mathsf{Qbit}], d{:}\widehat{}[\mathsf{Qbit}]) = (\mathsf{new}\ b,c)(Alice(a,b) \mid Noise(b,c) \mid Bob(c,d))$$

When we consider correctness of the error correction system, we will prove that *QECC* is equivalent to the following *identity process*, which by definition transmits a single qubit faithfully.

$$Identity(a{:}\widehat{}[\mathsf{Qbit}], d{:}\widehat{}[\mathsf{Qbit}]) = a?[x{:}\mathsf{Qbit}]\,.\,d![x]\,.\,\mathbf{0}$$

Correctness of QECC

We now sketch the proof that $QECC \leftrightarrows^c Identity$, which by Theorem 4.17 implies that the error correction system works in any context. An interesting consequence is that the qubit being transmitted may be part of any quantum state, meaning that it is correctly transmitted with error correction even if it is entangled with other qubits; the entanglement is also preserved by the error correction system. This property of error correction, although easily verified by hand, is not usually stated explicitly in the literature.

Lemma 4.18 ($Identity \leftrightarrows^c QECC$)**.**

Proof. First we prove that $QECC \leftrightarrows Identity$, by defining an equivalence relation $\mathcal{R}$ that contains the pair $((\sigma;\emptyset;QECC),(\sigma;\emptyset;Identity))$ for all σ and is closed under their transitions. $\mathcal{R}$ is defined by taking its equivalence classes to be the $S_i(\sigma)$ defined below, for all states σ. The idea is to group configurations according to the sequences of observable transitions leading to them. S_2 is also parameterized by the input qubit, as

this affects the output qubit and hence the equivalence class.

$$\begin{aligned} S_1(\sigma) &= \{s \mid (\sigma;\emptyset;P) \Longrightarrow s \text{ and } P \in \{QECC, Identity\}\} \\ S_2(\sigma,x) &= \{s \mid (\sigma;\emptyset;P) \stackrel{a?[x]}{\Longrightarrow} s \text{ and } P \in \{QECC, Identity\}\} \\ S_3(\sigma) &= \{s \mid (\sigma;\emptyset;P) \stackrel{a?[x]}{\Longrightarrow}\stackrel{d![x]}{\Longrightarrow} s \text{ and } P \in \{QECC, Identity\}\} \end{aligned}$$

We demonstrate the interesting steps in one possible execution of QECC, omitting the new declarations from the process terms to reduce clutter. The semantics of CQP is nondeterministic, so transitions can proceed in a different order; the order shown here is chosen for presentational convenience. The initial configuration is $(\sigma;\emptyset;\mathit{Alice} \mid \mathit{Noise} \mid \mathit{Bob})$, where σ is $[x \mapsto \alpha|0\rangle + \beta|1\rangle]$. In the first few steps which is a sequence of τ transitions, the processes execute qbit terms (denoted as $\Longrightarrow$), constructing a quantum state:

$$([x,y,z,u,v,s,t \mapsto \alpha|0\rangle + \beta|1\rangle \otimes |000000\rangle]; y,z,u,v,s,t; \mathit{Alice}' \mid \mathit{Noise}' \mid \mathit{Bob}')$$

Alice receives qubit x, in state $\alpha|0\rangle + \beta|1\rangle$, from the environment, via transition $\stackrel{a?[x]}{\longrightarrow}$. We now abbreviate the list of qubits to $\widetilde{q} = x,y,z,u,v,s,t$. After some τ transitions corresponding to *Alice*'s CNot operations, we have:

$$([\widetilde{q} \mapsto \alpha|0000000\rangle + \beta|1110000\rangle]; \widetilde{q}; b![x,y,z] \,.\, \mathbf{0} \mid \mathit{Noise}' \mid \mathit{Bob}')$$

$\mathit{Noise}' = \mathit{NoiseErr} \mid \mathit{NoiseRnd}'$ ($\mathit{NoiseRnd}'$ has already done its qbit). The output on b interacts with the input on b in *NoiseErr*. Meanwhile, the measurements in *NoiseRnd* produce a mixed configuration because the results are communicated internally, to *NoiseErr*:

$$\oplus_{j,k\in\{0,1\}} \frac{1}{4}(|\psi\rangle; \widetilde{q}; \lambda jk \bullet \{x *= \mathsf{X}^{jk}\} \,.\, \{y *= \mathsf{X}^{j\bar{k}}\} \,.\, \{z *= \mathsf{X}^{\bar{j}k}\} \,.\, c![x,y,z] \,.\, \mathbf{0} \mid \mathit{Bob}'; j,k)$$

Where $|\psi\rangle$ is $[\widetilde{q} \mapsto \alpha|000jk00\rangle + \beta|111jk00\rangle]$. After τ transitions from the controlled X operations, we can write the mixed configuration explicitly:

$$\begin{aligned} &\tfrac{1}{4}([\widetilde{q} \mapsto \alpha|0000000\rangle + \beta|1110000\rangle]; \widetilde{q}; c![x,y,z] \,.\, \mathbf{0} \mid \mathit{Bob}') \\ &\oplus \tfrac{1}{4}([\widetilde{q} \mapsto \alpha|0010100\rangle + \beta|1100100\rangle]; \widetilde{q}; c![x,y,z] \,.\, \mathbf{0} \mid \mathit{Bob}') \\ &\oplus \tfrac{1}{4}([\widetilde{q} \mapsto \alpha|0101000\rangle + \beta|1011000\rangle]; \widetilde{q}; c![x,y,z] \,.\, \mathbf{0} \mid \mathit{Bob}') \\ &\oplus \tfrac{1}{4}([\widetilde{q} \mapsto \alpha|1001100\rangle + \beta|0111100\rangle]; \widetilde{q}; c![x,y,z] \,.\, \mathbf{0} \mid \mathit{Bob}') \end{aligned}$$

The remaining transitions operate within the mixed configuration. In each component of the mixture, the measurement of s,t by *BobRec* has a deterministic outcome, so no further mixedness is introduced. Eventually we have a mixed configuration in which the process term is the same, $d![x] \,.\, \mathbf{0}$, in every component, so we can just consider the

mixed *state*, which is

$$\oplus_{j,k\in\{0,1\}}\frac{1}{4}[x,y,z,u,v,s,t \mapsto \alpha|000jkjk\rangle + \beta|100jkjk\rangle].$$

The mixture over j,k is the residue of the random choice made by *NoiseRnd*, and the dependence of s and t on j,k is because *BobRec*'s measurement recovers the values of j and k (which is what allows the error to be corrected). In this final mixed state, the reduced density matrix of x, which is what we are interested in when x is output, is the same as the original density matrix of x.

Now, we define $\mathcal{R}$ to be the relation where $S_1(\sigma)$,$S_2(\sigma)$ and $S_3(\sigma)$ are the equivalence classes:

$$\mathcal{R} \;=\; \bigcup_{i\in\{1,2,3\}}\{(s,t) \mid s,t \in S_i(\sigma)\}$$

We now prove that $\mathcal{R}$ is a probabilistic branching bisimulation. It suffices to consider transitions between S_i classes, as transitions within classes must be τ and are matched by τ.

If $s,t \in S_1(\sigma)$ and if $s \stackrel{\tau}{\Longrightarrow} s'$ then we have $s' \in S_1(\sigma)$. Therefore $(s',t) \in \mathcal{R}$. Otherwise if $s \stackrel{a?[x]}{\longrightarrow} s'$ then $s' \in S_2(\sigma)$ and we find t', t'' such that $t \Longrightarrow t' \stackrel{a?[x]}{\longrightarrow} t''$ with $t' \in S_1(\sigma)$ and $t'' \in S_2(\sigma)$, so $(s,t') \in \mathcal{R}$ and $(s',t'') \in \mathcal{R}$ as required.

Transitions from $S_2(\sigma)$ are matched similarly. There are no transitions from $S_3(\sigma)$.

There is no need for a probability calculation (case IV of Definition 7.3) because no probabilistic configurations arise; measurement results are always communicated internally, and never to the external environment.

Finally, because *QECC* and *Identity* have no free variables, their equivalence is trivially preserved by substitutions. □

4.3.2 Error Correction - Second Model

We now consider a different noise model shown in Figure 4.8, in which random X errors are applied independently to each of the three qubits being transmitted. In our previous model, the error causes only one of the qubit to be flipped, in which case the error could be corrected. Here, the error can cause any number of qubits to be flipped. The new definition of *Noise* is shown below; we use the original definitions of *Alice* and *Bob*; the overall system is now *QECC*2.

$$\begin{aligned}&\mathit{NoiseRnd}(p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit},\mathsf{bit}]) = \\ &\quad (\mathsf{qbit}\ u,v,w)\,.\,\{u \mathrel{*}= \mathsf{H}\}\,.\,\{v \mathrel{*}= \mathsf{H}\}\,.\,\{w \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u, \mathsf{measure}\ v, \mathsf{measure}\ w]\,.\,\mathbf{0}\end{aligned}$$

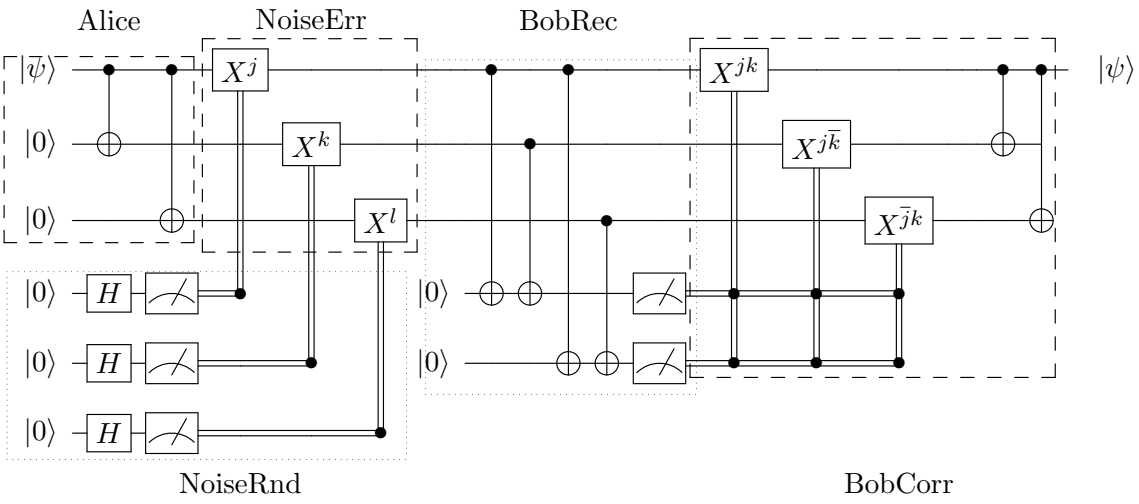


FIGURE 4.8: QECC2

Error qubits $\|uvw\rangle$	Error indication	Action	Final three qubit
$\|000\rangle$	No flip/error	No action	$\alpha_1\|000\rangle + \beta_1\|100\rangle$
$\|001\rangle$	Bit flip on qubit 3	Flip qubit 3 (X_3)	$\alpha_1\|000\rangle + \beta_1\|100\rangle$
$\|010\rangle$	Bit flip on qubit 2	Flip qubit 2 (X_2)	$\alpha_1\|000\rangle + \beta_1\|100\rangle$
$\|011\rangle$	Bit flip on qubits 2,3	Flip qubit 1 (X_1)	$\alpha_1\|100\rangle + \beta_1\|000\rangle$
$\|100\rangle$	Bit flip on qubit 1	Flip qubit 1 (X_1)	$\alpha_1\|000\rangle + \beta_1\|100\rangle$
$\|101\rangle$	Bit flip on qubits 1,3	Flip qubit 2 (X_2)	$\alpha_1\|100\rangle + \beta_1\|000\rangle$
$\|110\rangle$	Bit flip on qubits 1,2	Flip qubit 3 (X_3)	$\alpha_1\|100\rangle + \beta_1\|000\rangle$
$\|111\rangle$	Bit flip on all qubits	No action	$\alpha_1\|100\rangle + \beta_1\|000\rangle$

TABLE 4.1: Analysis for QECC2

$$NoiseErr(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit},\mathsf{bit}], c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) =$$
$$b?[x:\mathsf{Qbit}, y:\mathsf{Qbit}, z:\mathsf{Qbit}]\,.\,p?[j:\mathsf{bit}, k:\mathsf{bit}, l:\mathsf{bit}]\,.\,\{x *{=}\, \mathsf{X}^j\}\,.\,\{y *{=}\, \mathsf{X}^k\}\,.\,\{z *{=}\, \mathsf{X}^l\}\,.$$
$$c![x,y,z]\,.\,\mathbf{0}$$
$$Noise(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = (\mathsf{new}\ p)(NoiseRnd(p) \mid NoiseErr(b,p,c))$$

$$QECC2(a{:}\widehat{\ }[\mathsf{Qbit}], d{:}\widehat{\ }[\mathsf{Qbit}]) \quad = \quad (\mathsf{new}\ b,c)(Alice(a,b) \mid Noise(b,c) \mid Bob(c,d))$$

The overall analysis of $QECC2$ is provided in the table 4.1. We have eight possible ways of error actions which are shown in the table. Only four of them could be corrected which is an indication that the model corrects only if there is a bit flip on one of the qubits and not for the rest of the possibilities. The threefold repetition code is not able to correct multiple errors, so we do not have $QECC2 \leftrightarrows^c Identity$.

For a successful outcome, we get the final quantum state as $\alpha_1|000\rangle + \beta_1|100\rangle$ and otherwise the state is $\alpha_1|100\rangle + \beta_1|000\rangle$. The error correction system has a probability of $\frac{1}{2}$ of transmitting a qubit with an X error. We can express this in CQP by using

BitFlip as a specification process:

$$\begin{aligned} Rnd(p{:}\widehat{\ }[\mathsf{bit}]) &= (\mathsf{qbit}\ u)\{u \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u]\,.\,\mathbf{0} \\ Flip(a{:}\widehat{\ }[\mathsf{Qbit}], p{:}\widehat{\ }[\mathsf{bit}], d{:}\widehat{\ }[\mathsf{Qbit}]) &= a?[x:\mathsf{Qbit}]\,.\,p?[j:\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^i\}\,.\,d![x]\,.\,\mathbf{0} \\ BitFlip(a{:}\widehat{\ }[\mathsf{Qbit}], d{:}\widehat{\ }[\mathsf{Qbit}]) &= (\mathsf{new}\ p)(Rnd(p) \mid Flip(a,p,d)) \end{aligned}$$

As in the previous scenario, we perform a similar analysis for the model $QECC2$ and prove that it is equivalent to the its high-level specification, *BitFlip*.

Lemma 4.19 ($BitFlip \leftrightarrows^c QECC2$)**.**

Proof. We prove that $QECC2 \leftrightarrows BitFlip$, by defining an equivalence relation $\mathcal{R}$ that contains the pair $((\sigma;\emptyset;QECC2),(\sigma;\emptyset;BitFlip))$ for all σ and is closed under their transitions. First, we shall describe the execution of $QECC2$ and then we shall formally define an equivalence relation. Based on the execution, we analyse that this relation is a probabilistic branching bismulation

Consider an arbitrary quantum state $x = \alpha|0\rangle + \beta|1\rangle$. Let $s = (\alpha|0\rangle + \beta|1\rangle;\emptyset;QECC2)$, then the execution is as follows.

$$\begin{aligned} & s \xrightarrow{\tau} ([x,y,z,u,v,s,t \mapsto \alpha|0\rangle + \beta|1\rangle \otimes |000000\rangle]; y,z,u,v,s,t; Alice' \mid Noise' \mid Bob') \\ & \xrightarrow{a?[x]} ([\widetilde{q} \mapsto \alpha|0000000\rangle + \beta|1110000\rangle]; \widetilde{q}; b![x,y,z]\,.\,\mathbf{0} \mid Noise' \mid Bob') \\ & \xrightarrow{\tau} \oplus_{j,k,l\in\{0,1\}} \tfrac{1}{8}(|\psi\rangle; \widetilde{q}; \lambda jkl \bullet \{x \mathrel{*}= \mathsf{X}^j\}\,.\,\{y \mathrel{*}= \mathsf{X}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^l\}\,.\,c![x,y,z]\,.\,\mathbf{0} \mid Bob'; j,k,l) \\ & \xrightarrow{d![x]} \oplus_{j,k,l,m,n\in\{0,1\}} \tfrac{1}{8}(|\phi_i\rangle; \widetilde{q}; \lambda jklmn \bullet \mathbf{0}; j,k,l,m,n) \end{aligned}$$

where $i \in \{0,1\}$, $\phi_1 = \alpha|000jklmn\rangle + \beta|100jklmn\rangle$ such that $j,k,l \in \{000,001,010,100\}$ and $\phi_2 = \alpha|100jklmn\rangle + \beta|000jklmn\rangle$ such that $j,k,l \in \{101,110,011,111\}$. Thus, we find that out of the eight possible outcomes of the $QECC2$, four are correct and the other four are not. Hence, this model works with a probability of $\frac{1}{2}$.

As before, $\mathcal{R}$ is defined by taking its equivalence classes to be the $S_i(\sigma)$ defined below, for all states σ.

$$\begin{aligned} S_1(\sigma) &= \{s \mid (\sigma;\emptyset;P) \Longrightarrow s \text{ and } P \in \{QECC2, BitFlip\}\} \\ S_2(\sigma,x) &= \{s \mid (\sigma;\emptyset;P) \xrightarrow{a?[x]} s \text{ and } P \in \{QECC2, BitFlip\}\} \\ S_3(\sigma) &= \{s \mid (\sigma;\emptyset;P) \xrightarrow{a?[x]} \xrightarrow{d![x]} s \text{ and } P \in \{QECC2, BitFlip\}\} \end{aligned}$$

There is still no probability calculation because the results of the measurements in *NoiseRnd* and *Rnd* are not output. The equal probability of correct and incorrect transmission manifests itself in the fact that the reduced density matrix of the final output qubit, from both $QECC2$ and *BitFlip*, is an equal mixture of the input qubit and its inverse. For error qubits that are having values in set J where $J = \{011,101,110,111\}$,

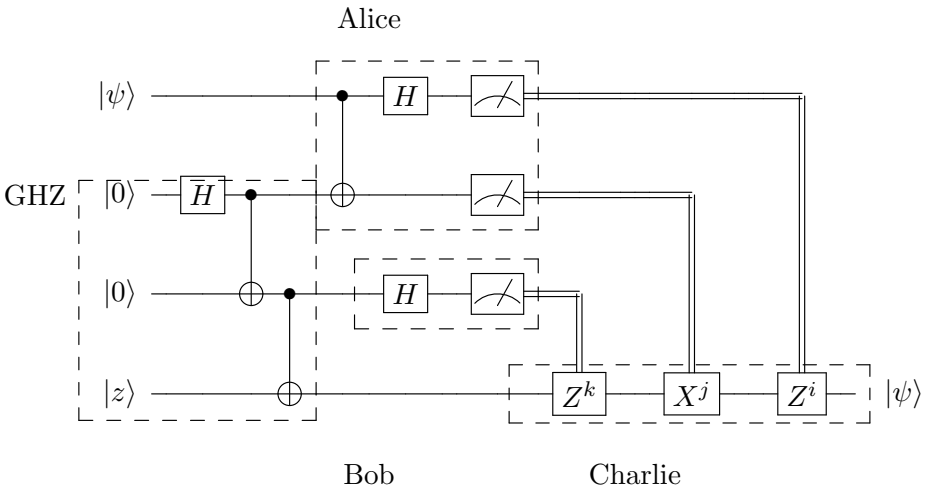


FIGURE 4.9: Quantum circuit for quantum secret sharing

we get the ouput qubit having a quantum state that is the inverse of the original input quantum state. We only get the correct output if the error qubits are having values that are in the set comprising $\{000, 001, 010, 100\}$. The transitions that happen between configurations from S_2 and S_3 which produces this inversion works with a probability of $\frac{1}{2}$ which clearly proves that $QECC2$ is not equivalent to the *Identity*. The only way to introduce probability into this example is for *Flip* to observably output j and *NoiseErr* to observably output the majority value of j, k, l, before the final qubit output. □

We know from the standard analysis of this error correction system that if the independent probability of flipping each qubit is $p < \frac{1}{2}$, $QECC2$ reduces the overall probability of a bit-flip error to $p^2(3 - 2p) < p$. This could be achieved if we could define a process that generates a probability p and the specification process would need to explicitly include the error probability. With a slightly more complicated analysis we could also express this property in CQP. In the later part of the thesis (Chapter 7), we demonstrate this property in our definition of the specification process to work with a probability of $\frac{1}{9}$ for the application of linear optical quantum computing.

4.3.3 Quantum Secret Sharing

We describe a quantum secret sharing protocol [88] that consists of three users represented by the processes *Alice*, *Bob* and *Charlie*. The quantum circuit of the protocol is represented in Figure 4.9. *Alice* would like to send a message to *Bob* and *Charlie*. She encodes her message in a way such that *Bob* and *Charlie* must cooperate with each other to retrieve it. The protocol begins by applying a Hadamard (H) and CNot operations to qubits x, y and z in order to generate the GHZ state $(\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ [82]. The qubits are shared between the three users. *Alice* also possesses the qubit labelled q which is in

some unknown state $|\psi\rangle$; this is the qubit she wishes to send. We analyse a scenario in which *Charlie* ends up with the original qubit.

Alice receives the qubit q ($[q \mapsto |\psi\rangle]$) from the environment through her channel c and performs unitary operations (CNot and H) before measuring her qubits. She sends the outcomes which are classical bits i and j through channel e to *Charlie*. *Charlie* cannot retrieve the information without the help of *Bob*. *Bob* performs a Hadamard operation on his qubit y before measuring it. Then, he sends the outcome to *Charlie*. Using the classical bits from *Alice* and *Bob*, *Charlie* performs the necessary unitary operations on his qubit z in order to recover the original state $|\psi\rangle$. The CQP definitions of the processes are:

$$\begin{aligned} \mathit{Alice}(c,e,x) &= c?[q:\mathsf{Qbit}]\,.\,\{q,x *\!\!= \mathsf{CNot}\}\,.\,\{q *\!\!= \mathsf{H}\}\,.\,e![\mathsf{measure}\ q, \mathsf{measure}\ x]\,.\,\mathbf{0} \\ \mathit{Bob}(f,y) &= \{y *\!\!= \mathsf{H}\}\,.\,f![\mathsf{measure}\ y]\,.\,\mathbf{0} \\ \mathit{Charlie}(e,f,d,z) &= e?[i:\mathsf{Bit}, j:\mathsf{Bit}]\,.\,f?[k:\mathsf{Bit}]\,.\,\{z *\!\!= \mathsf{Z}^k\}\,.\,\{z *\!\!= \mathsf{X}^j\}\,.\,\{z *\!\!= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0} \end{aligned}$$

The whole system is a parallel composition of the processes given by:

$$\begin{aligned} \mathit{QSS}(c,d) = (\mathsf{qbit}\ x,y,z)(\{x *\!\!= \mathsf{H}\}\,.\,\{x,y *\!\!= \mathsf{CNot}\}\,.\,\{y,z *\!\!= \mathsf{CNot}\}\,.\\ (\mathsf{new}\ e,f)(\mathit{Alice}(c,e,x) \mid \mathit{Bob}(f,y) \mid \mathit{Charlie}(e,f,d,z)) \end{aligned}$$

QSS process consists of *Alice*, *Bob* and *Charlie* in parallel. That is the outputs on e and f in *Alice* and *Bob* respectively synchronise with the inputs on e and f in *Charlie*. Channel e and f are designated as private local channels. The next three terms create the GHZ state with qubits x, y and z. The aim is to prove that *QSS* is equivalent to its specification process *Identity*. The execution of the protocol is shown in Figure 4.10.

Lemma 4.20 ($\mathit{Identity} \leftrightarrows^c \mathit{QSS}$)**.**

Proof. As similar to our previous examples, we prove that $\mathit{QSS} \leftrightarrows \mathit{Identity}$, by defining an equivalence relation $\mathcal{R}$ that contains the pair $((\sigma;\emptyset;\mathit{QSS}),(\sigma;\emptyset;\mathit{Identity}))$ for all σ and is closed under their transitions. $\mathcal{R}$ is defined by taking its equivalence classes to be the $T_i(\sigma)$ defined below, for all states σ. We group configurations according to the sequences of observable transitions leading to them. T_2 is also parameterized by the input qubit.

$$\begin{aligned} T_1(\sigma) &= \{t \mid (\sigma;\emptyset;P) \Longrightarrow t \text{ and } P \in \{\mathit{QSS}, \mathit{Identity}\}\} \\ T_2(\sigma,q) &= \{t \mid (\sigma;\emptyset;P) \overset{c?[q]}{\Longrightarrow} t \text{ and } P \in \{\mathit{QSS}, \mathit{Identity}\}\} \\ T_3(\sigma) &= \{t \mid (\sigma;\emptyset;P) \overset{c?[q]}{\Longrightarrow}\overset{d![q]}{\Longrightarrow} t \text{ and } P \in \{\mathit{QSS}, \mathit{Identity}\}\} \end{aligned}$$

$$([q \mapsto \alpha|0\rangle + \beta|1\rangle]; \emptyset; QSS)$$
$$\stackrel{\tau}{\Longrightarrow} (\text{L-QBIT}, \text{R-TRANS}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT})$$
$$([q, x, y, z \mapsto \alpha|0\rangle + \beta|1\rangle \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)]; x, y, z; (\mathsf{new}\ e, f)(Alice(c, e, x)|$$
$$Bob(f, y) \mid Charlie(e, f, d, z)))$$

$$\stackrel{c?[q]}{\longrightarrow} (\text{L-IN}, \text{L-ACT})$$
$$([q, x, y, z \mapsto \alpha|0\rangle + \beta|1\rangle \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)]; q, x, y, z; (\mathsf{new}\ e, f)(\{q, x *= \mathsf{CNot}\} . \{q *= \mathsf{H}\} .$$
$$e![\mathsf{measure}\ q, \mathsf{measure}\ x] . \mathbf{0} \mid Bob(f, y) \mid Charlie(e, f, d, z)))$$

$$\stackrel{\tau}{\Longrightarrow} (\text{R-TRANS}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT}, \text{R-MEASURE}, \text{L-ACT})$$
$$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}}} \frac{1}{4} ([\widetilde{r} \mapsto |\psi_{ij}\rangle]; \widetilde{r}; \lambda_{ij}.(\mathsf{new}\ e, f)(e![i, j] . \mathbf{0} \mid Bob(f, y) \mid Charlie(e, f, d, z)); i, j)$$

$$\stackrel{\tau}{\Longrightarrow} (\text{R-TRANS}, \text{L-ACT}, \text{R-MEASURE}, \text{L-ACT})$$
$$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}\\ k\in\{0,1\}}} \frac{1}{8} ([\widetilde{r} \mapsto |\phi_{ijk}\rangle]; \widetilde{r}; \lambda_{ijk}.(\mathsf{new}\ e, f)(e![i, j] . \mathbf{0} \mid f![k] . \mathbf{0} \mid Charlie(e, f, d, z)); i, j, k)$$

$$\stackrel{\tau}{\Longrightarrow} (\text{L-COM}, \text{L-ACT}, \text{L-COM}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT}, \text{R-TRANS}, \text{L-ACT})$$
$$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}\\ k\in\{0,1\}}} \frac{1}{8} ([\widetilde{r} \mapsto |\phi'_{ijk}\rangle]; \widetilde{r}; \lambda_{ijk}.(\mathsf{new}\ e, f)(d![z] . \mathbf{0}); i, j, k)$$

$$\stackrel{d?[z]}{\longrightarrow} (\text{L-OUT})$$
$$\oplus_{\substack{i\in\{0,1\}\\ j\in\{0,1\}\\ k\in\{0,1\}}} \frac{1}{8} ([\widetilde{r} \mapsto |\phi'_{ijk}\rangle]; \widetilde{r}; \lambda_{ijk} . \mathbf{0}; i, j, k)$$

where $\widetilde{r} = q, x, y, z$; $|\psi_{00}\rangle = \alpha|0000\rangle + \beta|0011\rangle$, $|\psi_{01}\rangle = \alpha|0111\rangle + \beta|0100\rangle$,
$|\psi_{10}\rangle = \alpha|1000\rangle - \beta|1011\rangle$, $|\psi_{11}\rangle = \alpha|1111\rangle - \beta|1100\rangle$, $|\phi_{000}\rangle = \alpha|0000\rangle + \beta|0001\rangle$,
$|\phi_{001}\rangle = \alpha|0010\rangle - \beta|0011\rangle$, $|\phi_{010}\rangle = \alpha|0101\rangle + \beta|0100\rangle$, $|\phi_{011}\rangle = -\alpha|0111\rangle + \beta|0110\rangle$,
$|\phi_{100}\rangle = \alpha|1000\rangle - \beta|1001\rangle$, $|\phi_{101}\rangle = \alpha|1010\rangle + \beta|1011\rangle$, $|\phi_{110}\rangle = \alpha|1101\rangle - \beta|1100\rangle$,
$|\phi_{111}\rangle = -\alpha|1111\rangle - \beta|1110\rangle$, $|\phi'_{000}\rangle = \alpha|0000\rangle + \beta|0001\rangle$, $|\phi'_{001}\rangle = \alpha|0010\rangle + \beta|0011\rangle$,
$|\phi'_{010}\rangle = \alpha|0100\rangle + \beta|0101\rangle$, $|\phi'_{011}\rangle = \alpha|0110\rangle + \beta|0111\rangle$, $|\phi'_{100}\rangle = \alpha|1000\rangle + \beta|1001\rangle$,
$|\phi'_{101}\rangle = \alpha|1010\rangle + \beta|1011\rangle$, $|\phi'_{110}\rangle = \alpha|1100\rangle + \beta|1101\rangle$, $|\phi'_{111}\rangle = \alpha|1110\rangle + \beta|1111\rangle$.

FIGURE 4.10: Execution of quantum secret sharing.

As before, we define $\mathcal{R}$ to be the relation where $T_1(\sigma)$,$T_2(\sigma)$ and $T_3(\sigma)$ are the equivalence classes:

$$\mathcal{R} \quad = \quad \bigcup_{i\in\{1,2,3\}}\{(t,u) \mid t,u \in T_i(\sigma)\}$$

We now prove that $\mathcal{R}$ is a probabilistic branching bisimulation. It suffices to consider transitions between T_i classes, as transitions within classes must be τ and are matched by τ.

If $t,u \in T_1(\sigma)$ and if $t \stackrel{\tau}{\Longrightarrow} t'$ then we have $t' \in T_1(\sigma)$. Therefore $(t',u) \in \mathcal{R}$. Otherwise if $t \stackrel{c?[q]}{\longrightarrow} t'$ then $t' \in T_2(\sigma)$ and we find u', u'' such that $u \Longrightarrow u' \stackrel{c?[q]}{\longrightarrow} u''$ with $u' \in T_1(\sigma)$ and $u'' \in T_2(\sigma)$, so $(t,u') \in \mathcal{R}$ and $(t',u'') \in \mathcal{R}$ as required.

If $t,u \in T_2(\sigma)$ and if $t \stackrel{\tau}{\Longrightarrow} t'$ then we have $t' \in T_2(\sigma)$. Therefore $(t',u) \in \mathcal{R}$. Otherwise if $t \stackrel{d![q]}{\longrightarrow} t'$ then $t' \in T_3(\sigma)$ and we find u', u'' such that $u \Longrightarrow u' \stackrel{d![q]}{\longrightarrow} u''$ with $u' \in T_2(\sigma)$ and $u'' \in T_3(\sigma)$, so $(t,u') \in \mathcal{R}$ and $(t',u'') \in \mathcal{R}$ as required. If t happens to be a mixed configuration arising from *QSS* then for an arbitrary state $\sigma = [q \mapsto \alpha|0\rangle + \beta|1\rangle]$, with reference to Figure 4.10, we have $\rho^z(t) = \rho^z(u')$. There are no transitions from $T_3(\sigma)$. □

Lemma 4.21 (*QECC* $\leftrightarrows^c$ *QSS* $\leftrightarrows^c$ *Teleport*)**.**

Proof. We have in [51] that *Teleport* $\leftrightarrows^c$ *Identity*. From Lemma 4.18 and Lemma 4.20, we have *QECC* $\leftrightarrows^c$ *Identity* and $\leftrightarrows^c$ *Identity*. Therefore, it is obvious that *QECC* $\leftrightarrows^c$ *QSS* $\leftrightarrows^c$ *Teleport*. □

Corollary 4.22 (*QECC* = *QSS* = *Teleport*)**.**

All the three protocols (*QECC*, *QSS*, *Teleport*) have a common function which is to input a qubit and provide the same identical qubit as an output through a definite channel. We consider a version of quantum secret sharing where the processes *Bob* and *Charlie* share the secret but we decide *Charlie* ends up with the original qubit. *QECC* and *Teleport* also has a definite input and an output. Hence, all these three processes perform similar function. This makes them equivalent to the specification process *Identity* and all three of them to be equal to each other.

4.3.4 Universal Composability

Universal composability [33] involves a system to be tested, a specification process which demonstrates an ideal functionality, two adversaries, and an environment. The system or protocol is said to have the ideal functionality if, for every attack on the protocol, there exists an attack on the specification, such that the observable behaviour of the protocol

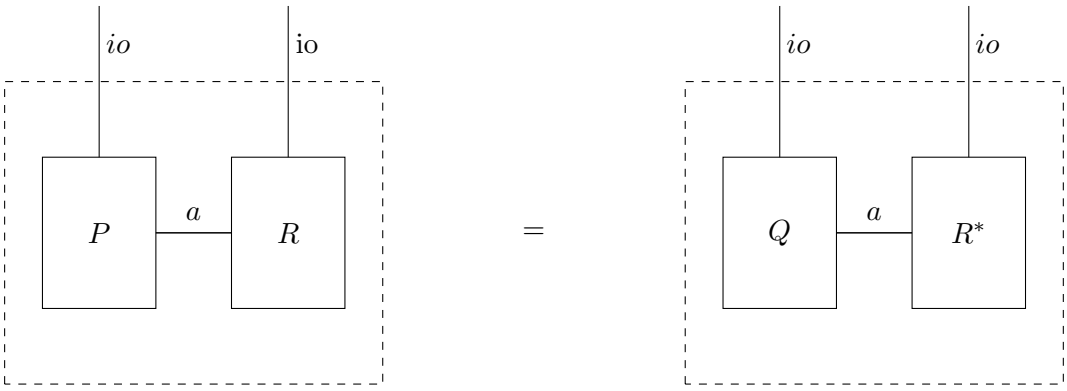


FIGURE 4.11: Universal Composability [50]

under attack is the same as the observable behaviour of the idealised functionality under attack. Now, we define the universal composability reaction on processes.

Definition 4.23 (Universal Composability)**.** A Process or protocol P is said to be equivalent to process Q (assuming Q is an idealised functionality) if for any adversary R attacking the protocol, there exists an adversary R^* attacking the ideal functionality, such that no context can distinguish whether it is interacting with P and R or with Q and R^*. Formally, $\forall R.\exists R^*.(\mathsf{new}\ a)(P \mid R) \leftrightarrows^c (\mathsf{new}\ a)(Q \mid R^*)$

Figure. 4.11 illustrates universal composability. We can think of the protocol (P) as *QECC* and the ideal functionality (Q) of the protocol as the *Identity* process. These processes communicate with the respective adversary processes over the channel (denoted a in the figure). These channels are not visible to the context or environment. However, the context gets to communicate with these processes over the input-output channels (denoted *io* in the figure).

We have seen that the two process expressions (*QECC* and *Identity*) in the definition of Universal Composability are observationally equivalent. This suggests that if there is an attack on the real protocol, then there exists an equivalent attack on the specification. Also, the congruence property of the equivalence confirms that the equivalence relationship holds good in any context. Therefore, $QECC \mid R \leftrightarrows^c Identity \mid R$. This gives a much stronger property to the definition. 4.23 where we have $R = R^*$. Hence, if the specification is unaffected to attack by any construction, then *QECC* that satisfies the above definition with respect to the ideal functionality also cannot be attacked. While [33] discuss an adversary and environment, the environment here is provided by the context used in the definition of $\leftrightarrows^c$, which is similar to the application in [50] for asynchronous classical communication.

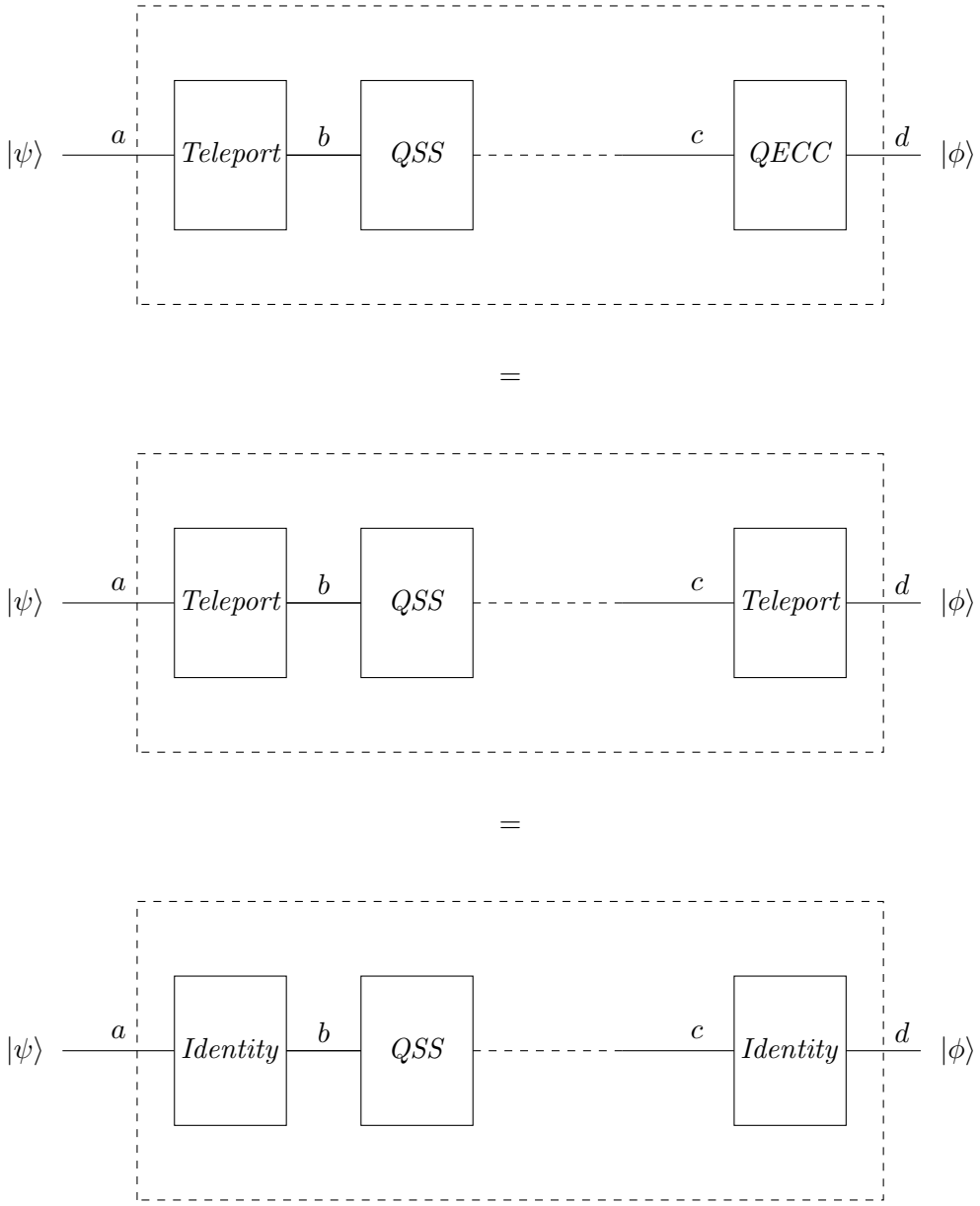


FIGURE 4.12: Compositional analysis

4.3.5 Compositional Analysis

Compositional analysis is the use of formal methods to support modular reasoning about systems that are constructed as combinations of sub-systems. These techniques have been developed for many concurrent languages [112]. Figure. 4.12 demonstrates compositional analysis where there is a system which is assumed to be made up of a combination of sub-systems like *Teleport*, *QSS*, *QECC* and so on. We have seen that *QECC* and *Teleport* are equivalent to each other as they have the same specification process *Identity*. The figure demonstrates the method of analysing each modules of a process that could be be a complicated combination of many subsystems.

4.4 Discussion

The labelled transition semantics of CQP are introduced in [51] to record the observational properties of both quantum and classical states. The important part of this semantic approach is the introduction of the mixed configuration that arises when the measurement outcomes are not communicated to the environment but rather internally between the sub-components of a system. Another crucial part of the mixed configurations is that it provides the equivalence of processes to have an important property of *congruence* and the theory has been used in the verification of quantum protocols namely teleportation (*Teleport*) and superdense coding (*SDC*).

We essentially use the theory as described in [51] and apply to quantum error code correction system. Quantum error correction can easily be analyzed by pen and paper, but the point of process calculus is that it forms part of a systematic methodology for verification of quantum systems. Two versions of a qubit error correction system based on the three qubit flip error correcting code are analysed and verified with respect to their specifications. We also prove that a version of the quantum secret sharing protocol is equivalent to the same specification process as that of the first model of quantum error code correction system.

Other error correcting codes. An interesting line of future work regarding this study would be to analyse other error correcting codes such as the three qubit phase flip code and the Shor code. In a phase flip error correction model, the Z operator is applied to the qubits to cause the phase flip, similar to the X operator in the bit flip error correction system. But a phase flip channel could be converted into a bit flip channel [132] if we work in the qubit basis, $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. We have seen that in bit flip error correction model the X operator acts as a bit flip taking $|0\rangle$ to $|1\rangle$ and vice versa. In a similar approach, the Z operator takes $|+\rangle$ to $|-\rangle$ and vice versa. The states in the phase flip error correction model are encoded as $|0_L\rangle = |++\!+\rangle$ and $|1_L\rangle = |--\!-\rangle$ as logical zero and one states. The operations that are involved in the error correction system such as encoding, flip error, detection and recovery, are performed the same as the bit flip. This is done with respect to the $|+\rangle$,—-$\rangle$ basis instead of $|0\rangle, |1\rangle$ basis. To obtain the basis change we need to apply the Hadamard (H) gate and its inverse (also the Hadamard gate) at appropriate points and this is can be achieved in CQP without much difficulty.

The Shor code protects the effect of an *arbitrary* error on a single qubit. The code is a combination of three qubit phase flip and three qubit bit flip codes. In a similar approach as shown in this chapter, we could model and analyse the Shor code system

in CQP, which is a detailed study that could be done as a future work in this field of research.

Congruence. This property of behavioural equivalence explicitly guarantees that equivalent processes remain equivalent in any context, and supports equational reasoning. For example: we have shown that $QECC \leftrightarrows^c Identity$ and $QSS \leftrightarrows^c Identity$; there is a proof in [51] that $Teleport \leftrightarrows^c Identity$; so we have, for free, that $QECC \leftrightarrows^c Teleport \leftrightarrows^c QSS$, in any context. This is also discussed in the next chapter.

Difference with qCCS. We shall briefly discuss the important differences in the definitions of processes in CQP and qCCS and a more detailed study is a part of the future work. The language presented in [66] and [170] models quantum information processing system that combines both quantum and classical information. The theory of equivalence is based on strong and weak bisimulation that is proposed in [66] and is proved to be a congruence. The theory is illustrated by verifying quantum teleportation and superdense coding protocols.

qCCS has a simpler syntax in comparison to CQP. The framework of qCCS does not include the evaluation of arbitrary expressions, which is included in CQP. The quantum operations are described usually by the application of a superoperator ($\mathcal{E}[\tilde{q}].P$). We show that the specification process for *QECC* is *Identity* which is the same for *Teleport* in [51]. But for qCCS, the specification process for teleportation is defined as a three qubit unitary operator, $SWAP_{1,3}$, which interchanges the first and third qubits. The specification process expressed in CQP is defined as

$$Tel_{spec} = a?[x] \,.\, \{x, y, z *= SWAP_{1,3}\} \,.\, d![z] \,.\, \mathbf{0}$$

The swap operator is introduced in qCCS, as the number of qubits and their names must be matched in the output action. In a similar approach, the specification process for the models (quantum error correction system and quantum secret sharing protocol) that we have analysed in this chapter, would be defined as processes $QECC_{spec}$ and QSS_{spec} in qCCS. These processes are expressed in CQP as:

$$QECC_{spec} = a?[x] \,.\, \{x, y, z, u, v, s, t *= \mathsf{I}\} \,.\, d![x] \,.\, \mathbf{0}$$

$$QSS_{spec} = c?[q] \,.\, \{q, x, y, z, *=SWAP_{1,4}\} \,.\, d![z] \,.\, \mathbf{0}$$

Here I is defined as the identity operator which does not change the qubits. We find that the specification processes for quantum protocols are to be defined differently in qCCS even though the processes provide the same output. But in CQP we have the

same specification process as our abstraction requires only the matching of the quantum state and not the quantum variables involved.

One of the essential conditions for a pair of processes to be bisimilar in qCCS is that the processes should have the same free quantum variables. From [66], we find that having identical free quantum variables is an essential requirement for strong bisimilarity $\sim$, weak bisimilarity $\approx$ and equality $\simeq$. This requirement excludes the pair of processes P and Q from being weakly bisimilar (or strongly bisimilar or equal), as highlighted in [66]. The definitions of P and Q, where I is the identity operator, written in qCCS syntax are:

$$P = I[q] \,.\, \mathbf{nil} \text{ and } Q = \tau \,.\, \mathbf{nil}$$

The corresponding processes defined in CQP,

$$P = \{q \mathrel{*}= \mathsf{I}\} \,.\, \mathbf{0} \text{ and } Q = \mathbf{0},$$

where I is the identity operator on a single qubit, are, in contrast bisimilar in an appropriate way. In fact, processes P and Q can be proved easily to be full probabilistic branching bisimilar.

Approximate bisimulation. Another interesting theory to be developed is the theory of approximate bisimulation. We have seen the model of quantum error correction ($QECC2$) that demonstrates correction with a probability of $\frac{1}{2}$. The theory of approximate bisimulation will help us to give more knowledge on the approximate equivalence of these systems with regard to certain specifications and would help to study more about the influence of decoherence in quantum information processing. The theory of approximate strong bisimulation has been defined for qCCS [66].

Chapter 5

Equational reasoning about quantum protocols

The congruence property of behavioural equivalence guarantees that equivalent processes remain equivalent in any context, which is the foundation for equational reasoning. In this Chapter, we define three new equational axioms to the existing work of CQP [51]. We show that we could analyse various quantum protocols like *quantum secret sharing*, *superdense coding*, *quantum error correction* and *remote-CNOT* by using the previous work [51] provided by Davidson along with the new axioms that are introduced in this thesis. This is achieved by using the theory to equate bisimilar process terms.

With the help of axiomatisation, one can avoid the use of computation of process terms and bisimulation relations. This gives rise to the possibility of automated reasoning, so that we can have a mechanised derivation that two process terms are bisimilar. In the previous work [51], Davidson proposed some axioms for full probabilistic branching bisimilarity and proved that the axioms are sound. The axioms were applied in the reasoning of quantum teleportation. Here, we look at a wider range of examples with applications involving quantum communication and quantum cryptography, which has led to the definition of some additional necessary axioms for the reasoning of these systems. The completeness of the axioms is not yet proved and is still a subject for future work.

5.1 Equational axioms of CQP

The axioms for full probabilistic branching bismilarity are shown in Figure 5.1 and have been proved to be sound in [51]. The axioms which are introduced in this thesis are

$$M \mid N = \Sigma_{i=1}^{m} \alpha_i.(P_i \mid N) + \Sigma_{j=1}^{n} \beta_j.(M \mid Q_j) + \Sigma_{\alpha_i C \beta_j} \tau.(P_i \mid Q_j) \quad \text{(E1)}$$

where $M = \Sigma_{i=1}^{m} \alpha_i.P_i$, $N = \Sigma_{j=1}^{n} \alpha_j.Q_j$ and $\alpha_i C \beta_j$ if α_i is $c![\widetilde{x}]$ and β_j is $c?[\widetilde{x}]$

$$\{\widetilde{x} *\!= V\}.\{\widetilde{x} *\!= W\}.P = \{\widetilde{x} *\!= U\}.P \quad \text{if } U = WV \quad \text{(QI1)}$$

$$\{\widetilde{y} *\!= \mathsf{U}^{\mathsf{measure}\ x}\}.P = \{x, \widetilde{y} *\!= \mathsf{CU}\}.\{\mathsf{measure}\ x\}.P \quad \text{(QI2)}$$

$$\{\widetilde{y} *\!= \mathsf{U}^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P = \{(x,z), \widetilde{y} *\!= \mathsf{CU}\}.\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ z\}.P \quad \text{(QI3)}$$

$$\{\widetilde{x} *\!= U\}.\{\widetilde{y} *\!= V\}.P = \{\widetilde{y} *\!= V\}.\{\widetilde{x} *\!= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC1)}$$

$$\{\widetilde{x} *\!= U\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\widetilde{x} *\!= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC2)}$$

$$\{\widetilde{x} *\!= U\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\widetilde{x} *\!= U\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC3)}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.\{\mathsf{measure}\ \widetilde{y}\}.P = \{\mathsf{measure}\ \widetilde{y}\}.\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC4)}$$

$$\{\mathsf{measure}\ \widetilde{x}\}.(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).\{\mathsf{measure}\ \widetilde{x}\}.P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC5)}$$

$$(\mathsf{qbit}\ \widetilde{x}).(\mathsf{qbit}\ \widetilde{y}).P = (\mathsf{qbit}\ \widetilde{y}).(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC6)}$$

$$\alpha.\{\widetilde{y} *\!= U\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\widetilde{y} *\!= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC7)}$$

$$\alpha.\{\widetilde{y} *\!= U\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\widetilde{y} *\!= U\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC8)}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c?[\widetilde{x}].P = \alpha.c?[\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC9)}$$

$$\alpha.\{\mathsf{measure}\ \widetilde{y}\}.c![\widetilde{x}].P = \alpha.c![\widetilde{x}].\{\mathsf{measure}\ \widetilde{y}\}.P \quad \text{if } \widetilde{y} \subseteq \mathbf{n}(\alpha), \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC10)}$$

$$(\mathsf{qbit}\ \widetilde{x}).c?[\widetilde{y}].P = c?[\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC11)}$$

$$(\mathsf{qbit}\ \widetilde{x}).c![\widetilde{y}].P = c![\widetilde{y}].(\mathsf{qbit}\ \widetilde{x}).P \quad \text{if } \widetilde{x} \cap \widetilde{y} = \emptyset \quad \text{(QC12)}$$

$$\{\mathsf{measure}\ x\}.\mathbf{0} = \mathbf{0} \quad \text{(QS1)}$$

$$\{\widetilde{x} *\!= U\}.\mathbf{0} = \mathbf{0} \quad \text{(QS2)}$$

$$(\mathsf{qbit}\ x).\mathbf{0} = \mathbf{0} \quad \text{(QS3)}$$

$$\alpha\,.\,\tau\,.\,P\,.\,\mathbf{0} = \alpha\,.\,P\,.\,\mathbf{0} \quad \text{(TAU1)}$$

$$\alpha.\{\widetilde{x} *\!= \Pi\}.P\{\pi(\widetilde{q})/\widetilde{x}\} = \alpha.P \quad \text{if } \widetilde{x} \subseteq \mathbf{n}(\alpha) \quad \text{(QP1)}$$

$$(\mathsf{qbit}\ x).\{\widetilde{y}, x *\!= U\}.P = (\mathsf{qbit}\ x).\{\widetilde{y}, x *\!= V\}.P \quad \text{if } U(I_{\widetilde{y}} \otimes |0\rangle) = V(I_{\widetilde{y}} \otimes |0\rangle) \quad \text{(QD1)}$$

$$c?[x : \mathsf{Bit}]\,.\,P(x) = c?[x : \mathsf{Bit}]\,.\,Q(x) \text{ if } P(x) = Q(x) \text{ for all } x \in \{0,1\} \quad \text{(CV1)}$$

$$(\mathsf{new}\ c)(P + Q) = (\mathsf{new}\ c)P + (\mathsf{new}\ c)Q \quad \text{(R1)}$$

$$(\mathsf{new}\ c)\alpha.P = \mathbf{0} \quad \text{if } \alpha \in \{c?[\cdot], c![\cdot]\} \quad \text{(R2)}$$

$$(\mathsf{new}\ c)\alpha.P = \alpha.(\mathsf{new}\ c)P \quad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \quad \text{(R3)}$$

FIGURE 5.1: Axioms for full probabilistic branching bisimilarity.

CV1, QI3 and TAU1.

$$c?[x : \mathsf{Bit}]\,.\,P(x) = c?[x : \mathsf{Bit}]\,.\,Q(x) \text{ if } P(x) = Q(x) \text{ for all } x \in \{0,1\} \quad \text{(CV1)}$$

The *classical value* rule CV1 enables us to compare processes that are controlled by the classical bit, say x. The emphasis of this rule is clearly visible when we analyse the superdense coding protocol.

The rules QI1 and QI2 that are introduced in [51], are called the quantum identity rules. The rule QI2 expresses the *principle of deferred measurement* [132] for an arbitrary unitary operator U. The rule is useful in the analysis of quantum protocols where the operator U is controlled by the measurement of a single qubit only. Rule QI3, introduced in this thesis, is an extension of the rule QI2. This rule expresses the principle of deferred

measurement, where the operator U is controlled by the measurement of more than one qubit.

$$\{\widetilde{y} *= \mathsf{U}^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P = \{(x,z), \widetilde{y} *= \mathsf{CU}\}.\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ z\}.P \quad \text{(QI3)}$$

In the previous chapter, we have analysed the quantum error code correction system, *QECC*. The protocol uses operators that are controlled by the measurement of two qubits. We show the need of the rule QI3 in section 5.5 to analyse *QECC*, by not creating bisimulation relations as seen in Chapter 4.

We define the rule TAU1 by

$$\alpha\,.\,\tau\,.\,P = \alpha\,.\,P \quad \text{(TAU1)}$$

Although this rule does not play a major role but it is needed to remove the un necessary τ which arise when we eliminate the parallel composition of processes.

5.2 Quantum Secret Sharing

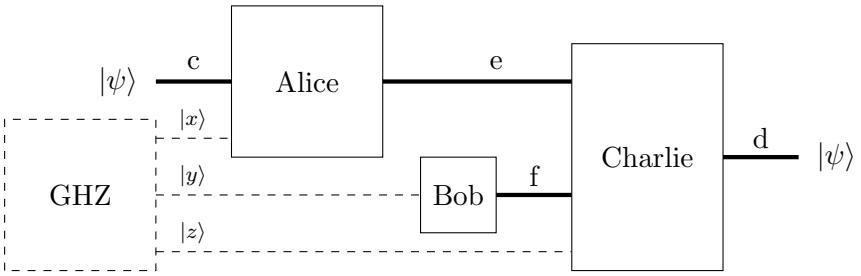


FIGURE 5.2: Quantum secret sharing protocol

In general, the schematic representation of the protocol is given in Figure 5.2 and the quantum circuit of the protocol is given in Figure 4.9. The boxes indicate the processes and the thick lines indicate the channels through which the processes communicate. The dashed lines represent the qubits that are associated with the respective processes. As seen earlier, we recall the CQP definitions of the processes that are involved in the protocol:

$$\begin{aligned} \mathit{Alice}(c,e,x) &= c?[q\!:\!\mathsf{Qbit}]\,.\,\{q,x *= \mathsf{CNot}\}\,.\,\{q *= \mathsf{H}\}\,.\,e![\mathsf{measure}\ q, \mathsf{measure}\ x]\,.\,\mathbf{0} \\ \mathit{Bob}(f,y) &= \{y *= \mathsf{H}\}\,.\,f![\mathsf{measure}\ y]\,.\,\mathbf{0} \\ \mathit{Charlie}(e,f,d,z) &= e?[i\!:\!\mathsf{Bit}, j\!:\!\mathsf{Bit}]\,.\,f?[k\!:\!\mathsf{Bit}]\,.\,\{z *= \mathsf{Z}^k\}\,.\,\{z *= \mathsf{X}^j\}\,.\,\{z *= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0} \end{aligned}$$

$$\begin{aligned} \mathit{QSS}(c,d) = (\mathsf{qbit}\ x,y,z)(&\{x *= \mathsf{H}\}\,.\,\{x,y *= \mathsf{CNot}\}\,.\,\{y,z *= \mathsf{CNot}\}\,. \\ &(\mathsf{new}\ e,f)(\mathit{Alice}(c,e,x) \mid \mathit{Bob}(f,y) \mid \mathit{Charlie}(e,f,d,z))) \end{aligned}$$

As before, our aim is to prove that *QSS* is equivalent to its specification process given by the following definition

$$Identity(a{:}\widehat{\ }[\mathsf{Qbit}], d{:}\widehat{\ }[\mathsf{Qbit}]) = a?[x{:}\mathsf{Qbit}]\,.\,d![x]\,.\,\mathbf{0}.$$

5.2.1 Expanding quantum secret sharing

The *expansion law* of CQP is used in expanding the definitions of the quantum processes [51]. The law is defined as:

If $M = \Sigma_{i=1}^{m}\alpha_i.P_i$ and $N = \Sigma_{j=1}^{n}\beta_j.Q_j$, then

$$M \mid N = \Sigma_{i=1}^{m}\alpha_i.(P_i \mid N) + \Sigma_{j=1}^{n}\beta_j.(M \mid Q_j) + \Sigma_{\alpha_i C \beta_j}\tau.(P_i \mid Q_j) \tag{E1}$$

where $\alpha_i C \beta_j$ identifies complementary actions, that is when α_i is an output $(c![\widetilde{x}])$ and β_j is a matching input $(c?[\widetilde{x}])$. The law is adapted from the *expansion lemma* of the π-calculus [150]. The prefixes of the terms in the first two parts of the summation correspond to the actions that the processes M and N can execute respectively. The third part is a summation corresponding to the communications between M and N.

The expansion law makes every action explicit by eliminating the parallel composition into a summation, in which each summation eliminates the parallel composition at the top level. Using this law many times results in a summation of sequential processes, where each term corresponds to a single interleaving of parallel operations. From [51], it is understood that a straightforward adaptation of the expansion law from the π-calculus is not possible due to the semantics of expressions in CQP.

We begin by applying the expansion law E1 to the definition of *QSS*, to get:

$$\begin{aligned}(\mathsf{qbit}\ x, y, z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x, y \mathrel{*}= \mathsf{CNot}\}\,.\,\{y, z \mathrel{*}= \mathsf{CNot}\}\,.\,(\mathsf{new}\ e, f)(c?[q]\,.\\ (Alice' \mid Bob \mid Charlie) + \{y \mathrel{*}= \mathsf{H}\}\,.\,(Alice \mid Bob' \mid Charlie) +\\ e?[i, j]\,.\,(Alice \mid Bob \mid Charlie'))\end{aligned} \tag{5.1}$$

where $Alice = c?[q{:}\mathsf{Qbit}]\,.\,Alice'$, $Bob = \{y \mathrel{*}= \mathsf{H}\}\,.\,Bob'$ and $Charlie = e?[i : \mathsf{Bit}, j : \mathsf{Bit}]\,.\,Charlie'$. The rules for manipulating restrictions represented in Figure 5.1 are R1, R2 and R3. These are common laws for classical process calculi.

Using the rules R1 and R2 on Eq. 5.1, the third term of the sum vanishes to give:

$$\begin{aligned}(\mathsf{qbit}\ x, y, z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x, y \mathrel{*}= \mathsf{CNot}\}\,.\,\{y, z \mathrel{*}= \mathsf{CNot}\}\,.\,(\mathsf{new}\ e, f)(c?[q{:}\mathsf{Qbit}]\,.\\ (Alice' \mid Bob \mid Charlie) + \{y \mathrel{*}= \mathsf{H}\}\,.\,(Alice \mid Bob' \mid Charlie))\end{aligned} \tag{5.2}$$

Expanding Eq. 5.2 as before, we get:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNot}\}\,.\,(\mathsf{new}\ e,f)(c?[q]\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\,(Alice' \mid Bob' \mid Charlie) + \{y \mathrel{*}= \mathsf{H}\}\,.\,c?[q]\,.\,(Alice' \mid Bob' \mid Charlie) + \{y \mathrel{*}= \mathsf{H}\}\,.\,f![\mathsf{measure}\ y]\,.\,(Alice \mid Bob' \mid Charlie) + c?[q]\,.\,\{q,x \mathrel{*}= \mathsf{CNot}\}\,.\,(Alice' \mid Bob \mid Charlie)) \tag{5.3}$$

The rules QC1 - QC12 are called the commuting operators. These rules help to swap the operators, actions and declarations around by using the commutativity principle. For example, the rule QC1 shows that we can swap the order of operators provided the qubits x and y are independent.

Now, using rules $\mathrm{R1}-\mathrm{R3}$ and QC7, QC8, we can commute between the process terms, which leads to the first two terms in Eq. 5.3 remaining the same and the third term is eliminated to give:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNot}\}\,.\,c?[q]\,.\,(\{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathsf{new}\ e,f)(Alice' \mid Bob' \mid Charlie) + \{q,x \mathrel{*}= \mathsf{CNot}\}\,.\,(\mathsf{new}\ e,f)(Alice' \mid Bob \mid Charlie)) \tag{5.4}$$

Expanding Eq. 5.4 and repeating the same procedure, we arrive at:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,\{y,z \mathrel{*}= \mathsf{CNot}\}\,.\,c?[q]\,.\,\{q,x \mathrel{*}= \mathsf{CNot}\}\,.\,\{q \mathrel{*}= \mathsf{H}\}\,.\,\{y \mathrel{*}= \mathsf{H}\}\,.\,(\mathsf{new}\ e,f)(e![\mathsf{measure}\ q,\mathsf{measure}\ x]\,.\,\mathbf{0} \mid f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid e?[i:\mathsf{Bit},j:\mathsf{Bit}]\,.\,f?[k:\mathsf{Bit}]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0}) \tag{5.5}$$

The next application of the expansion law results in the communication between *Alice* and *Charlie*, giving

$$(e![\mathsf{measure}\ q,\mathsf{measure}\ x]\,.\,\mathbf{0} \mid f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid e?[i,j]\,.\,f?[k]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0}) = e![\mathsf{measure}\ q,\mathsf{measure}\ x](\mathbf{0} \mid f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid e?[i,j]\,.\,f?[k]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0}) + f![\mathsf{measure}\ y](e![\mathsf{measure}\ q,\mathsf{measure}\ x]\,.\,\mathbf{0} \mid \mathbf{0} \mid e?[i:\mathsf{Bit},j:\mathsf{Bit}]\,.\,f?[k:\mathsf{Bit}]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0}) + e?[i,j]$$
$$(e![\mathsf{measure}\ q,\mathsf{measure}\ x]\,.\,\mathbf{0} \mid f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid f?[k]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0}) + \tau(\mathbf{0} \mid f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid f?[k]\,.\,\{z \mathrel{*}= \mathsf{Z}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^j\}\,.\,\{z \mathrel{*}= \mathsf{Z}^i\}\,.\,d![z]\,.\,\mathbf{0})$$

In this case, we have the complementary actions $e?[i,j]$ and $e![\mathsf{measure}\ q,\mathsf{measure}\ x]$, which result in the fourth term in the summation as representing the internal communication τ. Using the above in Eq. 5.5 and by including the restriction rules R1 and R2, we are able to identify the first three terms in the summation as semantically null

processes, thereby we get:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x,y \ast\!= \mathsf{CNot}\}\,.\,\{y,z \ast\!= \mathsf{CNot}\}\,.\,c?[q]\,.\,\{q,x \ast\!= \mathsf{CNot}\}\,.\\ \{q \ast\!= \mathsf{H}\}\,.\,\{y \ast\!= \mathsf{H}\}\,.\,(\mathbf{0}+\mathbf{0}+\mathbf{0}+(\mathsf{new}\ e,f)\,.\,\tau\,.\,f![\mathsf{measure}\ y]\,.\,\mathbf{0} \mid f?[k\!:\!\mathsf{Bit}]\,. \\ \{z \ast\!= \mathsf{Z}^k\}\,.\,\{z \ast\!= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}) \tag{5.6}$$

Similarly, the internal communication between *Bob* and *Charlie* gives rise to another τ and we get by using the expansion law and restriction rules:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x,y \ast\!= \mathsf{CNot}\}\,.\,\{y,z \ast\!= \mathsf{CNot}\}\,.\,c?[q]\,.\,\{q,x \ast\!= \mathsf{CNot}\}\,.\,\{q \ast\!= \mathsf{H}\}\,.\\ \{y \ast\!= \mathsf{H}\}\,.\,(\mathsf{new}\ e,f)\,.\,\tau\,.\,\tau\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \ast\!= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0} \tag{5.7}$$

Finallly, after several iterations using R3 and followed by $(\mathsf{new}\ e,f)\,.\,\mathbf{0}=\mathbf{0}$, we get:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x,y \ast\!= \mathsf{CNot}\}\,.\,\{y,z \ast\!= \mathsf{CNot}\}\,.\,c?[q]\,.\,\{q,x \ast\!= \mathsf{CNot}\}\,.\,\{q \ast\!= \mathsf{H}\}\,.\\ \{y \ast\!= \mathsf{H}\}\,.\,\tau\,.\,\tau\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \ast\!= \mathsf{X}^{\mathsf{measure}\ r}\}\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0} \tag{5.8}$$

The additional two τ transitions serves no purpose in this case. This is where the new rule Tau1 plays an important part. By using this rule, we remove the unwanted τ and arrive at the sequentialised definition of *QSS* represented by Eq. 5.9.

5.2.2 Analysis of *QSS*

In this section, we prove that quantum secret sharing (*QSS*) is equivalent to it's specification process (*Identity*), by using the axiomatic approach with respect to full probabilistic branching bisimilarity.

Proposition 5.1. *QSS* $\leftrightarrows^c$ *Identity*

Proof. Using the *expansion law* (E1) in process calculus, we can eliminate the parallel composition in the definition of the *QSS* process to a summation of sequential processes. Rules R1-R3 are common laws of classical process calculus and can be used for manipulating restrictions which gives rise to the following definition in sequentialised form for *QSS*, which was seen in the previous section:

$$(\mathsf{qbit}\ x,y,z)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x,y \ast\!= \mathsf{CNot}\}\,.\,\{y,z \ast\!= \mathsf{CNot}\}\,.\,c?[q\!:\!\mathsf{Qbit}]\,.\,\{q,x \ast\!= \mathsf{CNot}\}\,.\\ \{q \ast\!= \mathsf{H}\}\,.\,\{y \ast\!= \mathsf{H}\}\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \ast\!= \mathsf{X}^{\mathsf{measure}\ x}\}\,.\,\{z \ast\!= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0} \tag{5.9}$$

We will now simplify the above process and transform it into the *Identity* process by using the axioms in Figure 5.1. First, we use rule Qi1 that allows us to manipulate

quantum operators by combining the unitary actions into a single operation:

$$(\mathsf{qbit}\ x, y, z)\,.\,\{x, y, z \mathrel{*}\!= \mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,c?[q\!:\!\mathsf{Qbit}]\,.\,\{q, x, y \mathrel{*}\!= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}\}\,.\\ \{z \mathrel{*}\!= \mathsf{Z}^{\mathsf{measure}\ y}\}\,.\,\{z \mathrel{*}\!= \mathsf{X}^{\mathsf{measure}\ x}\}\,.\,\{z \mathrel{*}\!= \mathsf{Z}^{\mathsf{measure}\ q}\}\,.\,d![z]\,.\,\mathbf{0}$$

The subscripts on the unitary operators indicates to which qubits they are applied. Rule QI2 expresses the *principle of deferred measurement* [132]. Applying the rule QI2 to the measurement operations in the above process and noting that $\mathsf{CX} = \mathsf{CNOT}$, we get:

$$(\mathsf{qbit}\ x, y, z)\,.\,\{x, y, z \mathrel{*}\!= \mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,c?[q\!:\!\mathsf{Qbit}]\,.\,\{q, x, y \mathrel{*}\!= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}\}\,.\\ \{y, z \mathrel{*}\!= \mathsf{CZ}\}\,.\,\{\mathsf{measure}\ y\}\,.\,\{x, z \mathrel{*}\!= \mathsf{CNot}\}\,.\,\{\mathsf{measure}\ x\}\,.\\ \{q, z \mathrel{*}\!= \mathsf{CZ}\}\,.\,\{\mathsf{measure}\ q\}\,.\,d![z]\,.\,\mathbf{0}$$

Then, we swap the operators around due to commutativity, provided that the operators are not acting on the same qubits. For example, we swap the order of the measurement on z and the controlled-Z operator on x and y because the qubits are independent; mathematically, this is due to the use of the tensor product. The commutativity of internal operators are expressed by the rules QC1-QC6. Using QC2 on the above process, we can move the measurements, and then using QI1, the unitary operators are combined to give the resulting process:

$$(\mathsf{qbit}\ x, y, z)\,.\,\{x, y, z \mathrel{*}\!= \mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,c?[q\!:\!\mathsf{Qbit}]\,.\,\{q, x, y \mathrel{*}\!= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}\}\,.\\ \{q, x, y, z \mathrel{*}\!= \mathsf{CZ}_{qz}.\mathsf{CNot}_{xz}.\mathsf{CZ}_{yz}\}\,.\,\{\mathsf{measure}\ y\}\,.\,\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ q\}\,.\,d![z]\,.\,\mathbf{0}$$

The rules, QC7-QC10, consider the commutativity of unitary operations with input and output actions by applying certain conditions if $\widetilde{y} \subseteq \mathbf{n}(\alpha)$ and $\widetilde{x} \cap \widetilde{y} = \emptyset$. The first condition is important as it ensures that there is no blocking behaviour. This enables us to commute qubit declarations with input and output actions since a qubit declaration is never blocking. This is expressed by the rules QC11 and QC12. We use these rules to bring the input action to the top and move the measurement operations after the output to give:

$$c?[q]\,.\,(\mathsf{qbit}\ x, y, z)\,.\,\{x, y, z \mathrel{*}\!= \mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\{q, x, y \mathrel{*}\!= \mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}\}\,.\\ \{q, x, y, z \mathrel{*}\!= \mathsf{CZ}_{qz}.\mathsf{CNot}_{xz}.\mathsf{CZ}_{yz}\}\,.\,d![z]\,.\,\{\mathsf{measure}\ y\}\,.\,\{\mathsf{measure}\ x\}\,.\,\{\mathsf{measure}\ q\}\,.\,\mathbf{0}$$

With the help of the principle of deferred measurement, we were able to swap classical control for quantum control. Now we consider the *principle of implicit measurement* [132] which states that, any qubits at the end of a circuit may be assumed to be measured. This is provided by the rule QS1. Applying this rule to eliminate the measurements and

combining the remaining quantum operators with QI1, we obtain:

$$c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\,\{q,x,y,z \mathrel{*}= \mathsf{CZ}_{qz}.\mathsf{CNot}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}.\mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\\ d![z]\,.\,\mathbf{0}$$

Unitary operators and qubit declarations provide no observable effect at the end of a process. Hence, we have the rules QS2 and QS3. We see that the qubits y, q and x will each finish in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. So, we apply the Hadamard operator to each using the rule QS2 which allows these operations to be added. Combining these operators to a single unitary action by using QC8 and QI1; we get:

$$c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\,\{q,x,y,z \mathrel{*}= \mathsf{H}_y.\mathsf{H}_q.\mathsf{H}_x.\mathsf{CZ}_{qz}.\\ \mathsf{CNot}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}.\mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,d![z]\,.\,\mathbf{0}$$

Next, we insert a permutation in order to swap the output qubit z with q. Rule QP1 defines this action where π is the permutation of qubits and the corresponding permutation on the quantum state is given by Π. Applying this rule and followed by QI1, we get:

$$c?[q]\,.\,(\mathsf{qbit}\ x,y,z)\,.\,\{q,x,y,z \mathrel{*}= U\}\,.\,d![q]\,.\,\mathbf{0} \tag{5.10}$$

where $\pi(q) = z, \pi(z) = q, \pi(x) = x, \pi(y) = y$ and $U = \Pi.\mathsf{H}_y.\mathsf{H}_q.\mathsf{H}_x.\mathsf{CZ}_{qz}.\mathsf{CNot}_{xz}.\mathsf{CZ}_{yz}.\mathsf{H}_y.\mathsf{H}_q.\mathsf{CNot}_{qx}.\mathsf{CNot}_{yz}.\mathsf{CNot}_{xy}.\mathsf{H}_x$. Now, we have the qubit declaration $(\mathsf{qbit}\ x, y, z)$ which introduces three qubits in the combined state $|000\rangle$. We can define a linear map Q for which the action of teleportation on the single qubit q is given by UQ. We use the rule QD1 to deal with quantum operators that appear under qubit declarations.

We have $UQ = \mathsf{I}_{qxyz}Q$ where I_{qxyz} is the identity operator on qubits q, x, y, z. Then by applying QD1 to Eq. 5.10, we get:

$$c?[q]\,.\,\{q,x,y,z \mathrel{*}= \mathsf{I}\}\,.\,d![q]\,.\,\mathbf{0}$$

We now apply the following rules, QI1, QC8 and QS3 to give:

$$c?[q]\,.\,\{q \mathrel{*}= \mathsf{I}\}\,.\,d![q]\,.\,\mathbf{0}$$

This is a special case of QP1, where we consider identity permutation that results in the process, which we are aiming for:

$$c?[q]\,.\,d![q]\,.\,\mathbf{0}$$

□

5.3 Superdense Coding (SDC)

We recall the discussion on the protocol from Section 3.7.2. SDC [25] involves two users (*Alice* and *Bob*) sharing a pair of entangled qubits. In this protocol, two classical bits are communicated by exchanging a single qubit. Alice is in possession of the first qubit, while Bob has possession of the second qubit. By sending the single qubit in her possession to Bob, it turns out Alice can communicate two classical bits to Bob. The specification process for this protocol is *CIdent*. The CQP definitions of the processes involved in the protocol are:

$$
\begin{aligned}
Alice(c,e,x,y) &= c?[a:\mathsf{Bit}, b:\mathsf{Bit}] \,.\, \{x *= \mathsf{X}^b\} \,.\, \{y *= \mathsf{Z}^a\} \,.\, e![x] \,.\, \mathbf{0} \\
Bob(e,d,y) &= e?[x:\mathsf{Qbit}] \,.\, \{x,y *= \mathsf{CNot}\} \,.\, \{x *= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0} \\
SDC(c,d) &= (\mathsf{qbit}\ x,y)(\{x *= \mathsf{H}\} \,.\, \{x,y *= \mathsf{CNot}\} \,.\, (\mathsf{new}\ e)(Alice(c,e,x,y) \mid Bob(e,d,y)) \\
CIdent(c,d) &= c?[a:\mathsf{Bit}, b:\mathsf{Bit}] \,.\, d![a,b] \,.\, \mathbf{0}
\end{aligned}
$$

5.3.1 Analysis of SDC

As in the previous case, we apply the expansion law E1 to the definition of SDC, we get:

$$
\begin{aligned}
(\mathsf{qbit}\ x,y) \,.\, \{x *= \mathsf{H}\} \,.\, \{x,y *= \mathsf{CNot}\} \,.\, (\mathsf{new}\ e)(c?[a,b] \,.\, (Alice' \mid Bob) + \\
e?[x] \,.\, (Alice \mid Bob'))
\end{aligned}
\tag{5.11}
$$

where $Alice' = c?[a,b] \,.\, Alice$ and $Bob' = e?[x] \,.\, Bob$.

Using the rules R1 – R3 on Eq. 5.11, the second term of the sum vanishes. Rearranging the terms, we get:

$$
\begin{aligned}
(\mathsf{qbit}\ x,y) \,.\, \{x *= \mathsf{H}\} \,.\, \{x,y *= \mathsf{CNot}\} \,.\, c?[a,b] \,.\, (\mathsf{new}\ e)(\{x *= \mathsf{X}^b\} \,.\, \{y *= \mathsf{Z}^a\} \\
\,.\, e![x] \,.\, \mathbf{0} \mid e?[x] \,.\, \{x,y *= \mathsf{CNot}\} \,.\, \{x *= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0})
\end{aligned}
\tag{5.12}
$$

Expanding Eq. 5.12 as before and doing similar manipulations, we arrive at:

$$
\begin{aligned}
(\mathsf{qbit}\ x,y) \,.\, \{x *= \mathsf{H}\} \,.\, \{x,y *= \mathsf{CNot}\} \,.\, c?[a,b] \,.\, \{x *= \mathsf{X}^b\} \,.\, \{y *= \mathsf{Z}^a\} \,. \\
(\mathsf{new}\ e)(e![x] \,.\, \mathbf{0} \mid e?[x] \,.\, \{x,y *= \mathsf{CNot}\} \,.\, \{x *= \mathsf{H}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0})
\end{aligned}
\tag{5.13}
$$

Again using the expansion law and the restriction rules, we get a summation of three terms of which one of them is an internal communication (τ transition). Performing

several iterations using R3 and followed by $(\mathsf{new}\ e)\,.\,\mathbf{0} = \mathbf{0}$, we arrive at:

$$(\mathsf{qbit}\ x, y)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x, y \ast\!= \mathsf{CNot}\}\,.\,c?[a, b]\,.\,\{x \ast\!= \mathsf{X}^b\}\,.\,\{y \ast\!= \mathsf{Z}^a\}\,.\\ \tau\,.\,\{x, y \ast\!= \mathsf{CNot}\}\,.\,\{x \ast\!= \mathsf{H}\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{5.14}$$

Removing the τ from Eq. 5.14 by using the rule Tau1, we get the sequentialised form of the definition of *SDC*:

$$(\mathsf{qbit}\ x, y)\,.\,\{x \ast\!= \mathsf{H}\}\,.\,\{x, y \ast\!= \mathsf{CNot}\}\,.\,c?[a, b]\,.\,\{x \ast\!= \mathsf{X}^b\}\,.\,\{y \ast\!= \mathsf{Z}^a\}\\ \{x, y \ast\!= \mathsf{CNot}\}\,.\,\{x \ast\!= \mathsf{H}\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{5.15}$$

Proposition 5.2. *SDC* $\leftrightarrows^c$ *CIdent*

Proof. We begin with Eq. 5.15, and using the rule Qi1 to combine the unitary actions to get:

$$(\mathsf{qbit}\ x, y)\,.\,\{x, y \ast\!= \mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,c?[a, b]\,.\,\{xy \ast\!= \mathsf{H}_x.\mathsf{CNot}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x\}\,.\\ d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{5.16}$$

Moving the input actions to the top by applying Qc7 and Qc11 on Eq. 5.16, we get:

$$c?[a, b]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{x, y \ast\!= \mathsf{CNot}_{xy}.\mathsf{H}_x\}\{xy \ast\!= \mathsf{H}_x.\mathsf{CNot}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x\}\\ d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{5.17}$$

Applying Qi1 on Eq. 5.17, we arrive at:

$$c?[a, b]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{x, y \ast\!= \mathsf{H}_x.\mathsf{CNot}_{xy}.\mathsf{Z}^a_{y.}\mathsf{X}^b_x.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0}$$

Rewriting the above as:

$$c?[a\!:\!\mathsf{Bit}, b\!:\!\mathsf{Bit}]\,.\,(\mathsf{Qbit}:x, y)\,.\,\{xy \ast\!= U^{ab}\}\,.\,d![\mathsf{measure}\ x, \mathsf{measure}\ y]\,.\,\mathbf{0} \tag{5.18}$$

Here, $U^{ab} = \mathsf{H}_x.\mathsf{CNot}_{xy}.\mathsf{Z}^a_y.\mathsf{X}^b_x.\mathsf{CNot}_{xy}.\mathsf{H}_x$, is a unitary operator which depends on the classical bits a and b.

Eq. 5.18 is a process, which is parameterised by the classical bits a and b. The original axioms of CQP [51] did not have the rule to analyse a process that depends on the classical bits. Hence, it was necessary to define a new rule Cv1 that enables us to analyse process parametrised by classical values.

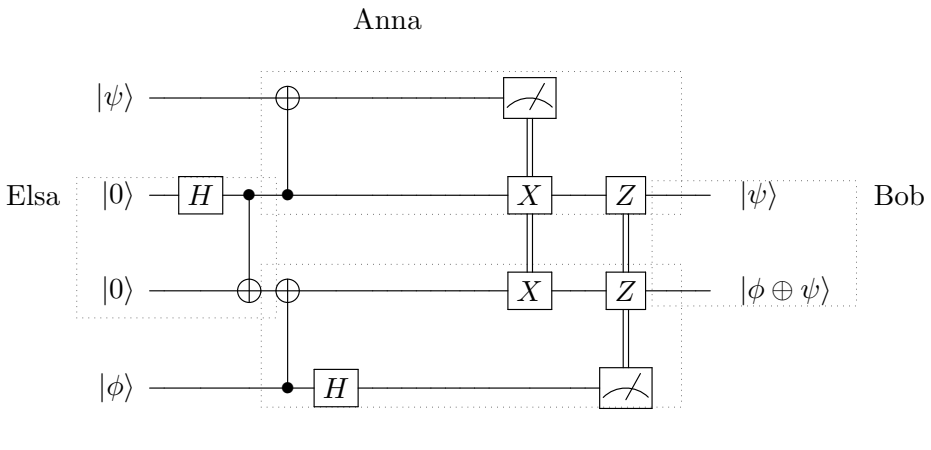


FIGURE 5.3: Remote CNOT

In order to do this, we define two processes P and Q that are parameterised by the classical bits a and b:

$$P(a,b) = (\mathsf{Qbit} : x, y) \,.\, \{xy \ast\!= U^{ab}\} \,.\, d![\mathsf{measure}\ x, \mathsf{measure}\ y] \,.\, \mathbf{0} \text{ and}$$
$$Q(a,b) = d![a,b] \,.\, \mathbf{0}$$

We evaluate the outcomes of the processes $P(a,b)$ and $Q(a,b)$ for all values of a and b. The results are provided in the Table 5.1.

a	b	$\|x\rangle$	$\|y\rangle$	$\{xy \ast\!= U^{ab}\}$	Result of $P(a,b)$	Result of $Q(a,b)$
0	0	$\|0\rangle$	$\|0\rangle$	$\|00\rangle$	00	00
0	1	$\|0\rangle$	$\|0\rangle$	$\|01\rangle$	01	01
1	0	$\|0\rangle$	$\|0\rangle$	$\|10\rangle$	10	10
1	1	$\|0\rangle$	$\|0\rangle$	$\|11\rangle$	11	11

TABLE 5.1: Outcomes of processes $P(a,b)$ and $Q(a,b)$

From Table 5.1, we find that the results for the processes P and Q, are the same for all values of a and b. Hence using the rule Cv1, we can confirm that $P(a,b) \leftrightarrows^c Q(a,b)$ for all possible values of a and b. There by, Eq. 5.18 $\leftrightarrows^c c?[a,b] \,.\, d![a,b] \,.\, \mathbf{0}$, which is the specification process *CIdent*. □

5.4 Remote CNOT (RCNOT)

The quantum circuit of the protocol (*RCNOT*), shown in Figure 5.3 [172], demonstrates the concept of teleporting a quantum logic gate. Here, the principle of quantum teleportation is extended to quantum gates. The concept involved is to act on remote qubits from a distance. This is referred to as *distributed quantum computation* [53] where a

CNOT from one user's state is transferred to other's, without communicating any quantum information between them. To perform this, we need to use qubit teleportation back and forth to perform computations as demonstrated in [172]. We assume that the protocol consists of four users: *Elsa*, *Anna*, *Iven* and *Bob* given by the respective CQP definitions.

$$
\begin{aligned}
Elsa(a,c,d) &= (\mathsf{qbit}\ x,y)a?[q:\mathsf{Qbit}, r:\mathsf{Qbit}]\,.\,\{x \mathrel{*}\!= \mathsf{H}\}\,.\,\{x,y \mathrel{*}\!= \mathsf{CNot}\}\,.\,c![q,x]\,.\,d![r,y]\,.\,\mathbf{0}\\
Anna(c,e,f,g) &= c?[q,x]\,.\,\{x,q \mathrel{*}\!= \mathsf{CNot}\}\,.\,e?[j:\mathsf{Bit}]\,.\,\{x \mathrel{*}\!= \mathsf{X}^{\mathsf{measure}\ q}\}\,.\,f![\mathsf{measure}\ q]\,.\\
&\quad \{x \mathrel{*}\!= \mathsf{Z}^{j}\}\,.\,g![x]\,.\,\mathbf{0}\\
Iven(d,f,e,h) &= d?[r,y]\,.\,\{r,y \mathrel{*}\!= \mathsf{CNot}\}\,.\,\{r \mathrel{*}\!= \mathsf{H}\}\,.\,e![\mathsf{measure}\ r]\,.\,f?[i:\mathsf{Bit}]\,.\,\{y \mathrel{*}\!= \mathsf{X}^{i}\}\,.\\
&\quad \{y \mathrel{*}\!= \mathsf{Z}^{\mathsf{measure}\ r}\}\,.\,h![y]\,.\,\mathbf{0}\\
Bob(g,h,b) &= g?[x]\,.\,h?[y]\,.\,b![x,y]\,.\,\mathbf{0}
\end{aligned}
$$

Anna and *Iven* have in their possession qubits q and r respectively, which they have received from *Elsa*. Also, *Elsa* has prepared an EPR pair with qubits x and y before sharing it with *Anna* and *Iven*. The objective of the protocol is that *Anna* and *Iven* would like to perform a CNot operation with their qubits q and r, without communicating any quantum information between them. *Anna* entangles her qubits q and x by performing a CNot. *Iven* performs the same with his qubits in addition to a H operation on r, before measuring it. He then sends the result to *Anna*. She measures her qubit q and performs certain unitary operations on x based on the outcomes of her and *Iven* measurements. Also, she sends her measurement outcome to *Iven*. Hence, *Anna* and *Iven* communicate only their classical results between them, which are used to perform unitary operation on their EPR pair. Essentially *Iven*'s qubit y is a CNot operation of q and r and they communicate their EPR pair qubits (x and y) to *Bob*. The specification of *RCNOT* is *SCNOT*. Let $\widetilde{k}$ be a list of channels that comprises c, d, e, f, g and h. The CQP definitions of the protocol and it's specification are:

$$
\begin{aligned}
RCNOT(a,b) &= (\mathsf{new}\ \widetilde{k})(Elsa(a,c,d) \mid Anna(c,e,f,g) \mid Iven(d,f,e,h) \mid Bob(g,h,b))\\
SCNOT(a,b) &= a?[q:\mathsf{Qbit}, r:\mathsf{Qbit}]\,.\,\{r,q \mathrel{*}\!= \mathsf{CNot}\}\,.\,b![q,r]\,.\,\mathbf{0}
\end{aligned}
$$

5.4.1 Analysis of *RCNOT*

Proposition 5.3. $RCNOT \leftrightarrows^{c} SCNOT$

Proof. We begin with the sequential form of the CQP definition of *RCNOT*. This is achieved like the previous cases by applying the expansion law and manipulation of

restrictions.

$$(\mathsf{qbit}\ x, y)\,.\,a?[q{:}\mathsf{Qbit}, r{:}\mathsf{Qbit}]\,.\,\{x \mathrel{*}= \mathsf{H}\}\,.\,\{x, y \mathrel{*}= \mathsf{CNot}\}\,.\,\{x, q \mathrel{*}= \mathsf{CNot}\}\,.\\ \{r, y \mathrel{*}= \mathsf{CNot}\}\,.\,\{r \mathrel{*}= \mathsf{H}\}\,.\,\{x \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ q}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{\mathsf{measure}\ q}\}\,.\,\{x \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ r}\}\,.\\ \{y \mathrel{*}= \mathsf{Z}^{\mathsf{measure}\ r}\}\,.\,b![x, y]\,.\,\mathbf{0}$$

Applying QI1 and QI2 to combine the unitary operations, we get:

$$(\mathsf{qbit}\ x, y)a?[q, r]\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,\{q, x, y \mathrel{*}= \mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}\}\,.\\ \{\mathsf{measure}\ q\}\,.\,\{r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}\}\,.\,\{\mathsf{measure}\ r\}\,.\,b![x, y]\,.\,\mathbf{0} \tag{5.19}$$

Now, we use the rules QC2, QC10, and QS1 on Eq. 5.19 to remove the measurements:

$$(\mathsf{qbit}\ x, y)a?[q, r]\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\\ \{q, x, y \mathrel{*}= \mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}\}\,.\,\{r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}\}\,.\,b![x, y]\,.\,\mathbf{0} \tag{5.20}$$

With the help of QC11 and QI1, we move the input action in the front of Eq. 5.20 and combine the unitary operations to get:

$$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}.\\ \mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,b![x, y]\,.\,\mathbf{0} \tag{5.21}$$

Applying QS2, QC8 and QI1 on Eq. 5.21, to add a Hadamard operation on qubit r to give:

$$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}.\mathsf{H}_r.\\ \mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,b![x, y]\,.\,\mathbf{0} \tag{5.22}$$

Now we apply the permutation operator to perform $\pi(q) = x$ and $\pi(x) = q$ by using the rule QP1 to give

$$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \Pi.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}.\\ \mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,b![q, y]\,.\,\mathbf{0} \tag{5.23}$$

By using QS2, QC8 and QI1 to add a Hadamard operation on qubit x. We get:

$$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \mathsf{H}_x.\Pi.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNot}_{qy}.\\ \mathsf{CNot}_{qx}.\mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,b![q, y]\,.\,\mathbf{0}$$

Applying QP1 a permutation operator as before to perform $\pi(r) = y$ and $\pi(y) = r$, we get:

$$a?[q, r]\,.\,(\mathsf{qbit}\ x, y)\,.\,\{q, r, x, y \mathrel{*}= \Pi.\mathsf{H}_x.\Pi.\mathsf{H}_r.\mathsf{CZ}_{ry}.\mathsf{CZ}_{rx}.\mathsf{CNot}_{qy}.\mathsf{CNot}_{qx}.\\ \mathsf{H}_r.\mathsf{CNot}_{ry}.\mathsf{CNot}_{xq}.\mathsf{CNot}_{xy}.\mathsf{H}_x\}\,.\,b![q, r]\,.\,\mathbf{0}$$

Then using QD1, to get:

$$a?[q,r]\,.\,(\mathsf{qbit}\ x,y)\,.\,\{r,q \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,y \mathrel{*}= \mathsf{I}\}\,.\,b![q,r]\,.\,\mathbf{0}$$

Finally, using the rules QC8, QS2, QC3 and QS3, we get:

$$a?[q,r]\,.\,\{r,q \mathrel{*}= \mathsf{CNot}\}\,.\,b![q,r]\,.\,\mathbf{0} \tag{5.24}$$

where Eq. 5.24 is the same as the specification process $SCNOT$. □

5.5 Verification of quantum error correction by equational reasoning

We have seen in Chapter 4 that the process $QECC$ is equivalent to $Identity$. This has been proved by using the bisimulation relations. Now, we show through equational theory that these processes are equivalent to each other. Recalling the CQP definitions of $QECC$ which consists of three processes: $Alice$, Bob and $Noise$. $Alice$ wishes to send a qubit to Bob over a noisy channel, represented by $Noise$. She uses a error correcting code based on threefold repetition [132]. The code is able to correct single bit-flip error in each block of three transmitted qubits. Bob uses the appropriate decoding procedure to recover $Alice$'s original qubit. The CQP definitions of the system are:

$$\begin{aligned}
Alice(a,b) &= (\mathsf{qbit}\ y,z)a?[x\!:\!\mathsf{Qbit}]\,.\,\{x,z \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,b![x,y,z]\,.\,\mathbf{0}\\
NoiseRnd(p) &= (\mathsf{qbit}\ u,v)\{u \mathrel{*}= \mathsf{H}\}\,.\,\{v \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u, \mathsf{measure}\ v]\,.\,\mathbf{0}\\
NoiseErr(b,p,c) &= b?[x\!:\!\mathsf{Qbit}, y\!:\!\mathsf{Qbit}, z\!:\!\mathsf{Qbit}]\,.\,p?[j\!:\!\mathsf{bit}, k\!:\!\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^{jk}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{j\bar{k}}\}\,.\\
&\quad \{z \mathrel{*}= \mathsf{X}^{\bar{j}k}\}\,.\,c![x,y,z]\,.\,\mathbf{0}\\
Noise(b,c) &= (\mathsf{new}\ p)(NoiseRnd(p) \mid NoiseErr(b,p,c))\\
BobRec(c,p) &= (\mathsf{qbit}\ s,t)c?[x,y,z]\,.\,\{x,s \mathrel{*}= \mathsf{CNot}\}\,.\,\{y,s \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,t \mathrel{*}= \mathsf{CNot}\}\,.\\
&\qquad \{z,t \mathrel{*}= \mathsf{CNot}\}\,.\,p![x,y,z,\mathsf{measure}\ s, \mathsf{measure}\ t]\,.\,\mathbf{0}\\
BobCorr(p,d) &= p?[x,y,z,j\!:\!\mathsf{bit}, k\!:\!\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^{jk}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{j\bar{k}}\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\bar{j}k}\}\,.\\
&\qquad \{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,z \mathrel{*}= \mathsf{CNot}\}\,.\,d![x]\,.\,\mathbf{0}\\
Bob(c,d) &= (\mathsf{new}\ p)(BobRec(c,p) \mid BobCorr(p,d))\\
QECC(a,d) &= (\mathsf{new}\ b,c)(Alice(a,b) \mid Noise(b,c) \mid Bob(c,d))
\end{aligned}$$

Proposition 5.4. $QECC \leftrightarrows^{c} Identity$

Proof. We begin with the configuration which is obtained after eliminating the parallel composition and after the application of the rule QI1.

$$\begin{aligned}(\mathsf{qbit}\ y,z)\,.\,a?[x]\,.\,\{x,y,z \ast\!\!= \mathsf{CNot}_{xy}.\mathsf{CNot}_{xz}\}\,.\,(\mathsf{qbit}\ u,v)\{u,v \ast\!\!= \mathsf{H}_v.\mathsf{H}_u\}\,.\\ \{x \ast\!\!= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\}\,.\,\{y \ast\!\!= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\}\,.\,\{z \ast\!\!= \mathsf{X}^{\mathsf{measure}\ u.\mathsf{measure}\ v}\}\,.\\ (\mathsf{qbit}\ s,t)\,.\,\{x,y,z,s,t \ast\!\!= \mathsf{CNot}_{zt}.\mathsf{CNot}_{xt}.\mathsf{CNot}_{ys}.\mathsf{CNot}_{xs}\}\,.\,\{x \ast\!\!= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\}\,.\\ \{y \ast\!\!= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\}\,.\,\{z \ast\!\!= \mathsf{X}^{\mathsf{measure}\ s.\mathsf{measure}\ t}\}\,.\,\{x,y,z \ast\!\!= \mathsf{CNot}_{xz}.\mathsf{CNot}_{xy}\}\,.\,d![x]\,.\,\mathbf{0}\end{aligned} \tag{5.25}$$

Applying the rules QI3, QC2, QS1 and QC3 one after other on Eq. 5.25, we get:

$$\begin{aligned}(\mathsf{qbit}\ y,z)a?[x]\,.\,(\mathsf{qbit}\ u,v)\,.\,(\mathsf{qbit}\ s,t)\{x,y,z,u,v \ast\!\!= \mathsf{CNot}_{(uv)z}.\mathsf{CNot}_{(uv)y}.\\ \mathsf{CNot}_{(uv)x}.\mathsf{H}_v.\mathsf{H}_u.\mathsf{CNot}_{xy}.\mathsf{CNot}_{xz}\}\,.\,\{x,y,z,s,t \ast\!\!= \mathsf{CNot}_{xz}.\\ \mathsf{CNot}_{xy}.\mathsf{CNot}_{(st)z}.\mathsf{CNot}_{(st)y}.\mathsf{CNot}_{(st)x}.\mathsf{CNot}_{zt}.\mathsf{CNot}_{xt}.\mathsf{CNot}_{ys}.\mathsf{CNot}_{xs}\}\,.\,d![x]\,.\,\mathbf{0}\end{aligned} \tag{5.26}$$

Using rules QC11 and QI1 on Eq. 5.26 to get:

$$\begin{aligned}a?[x]\,.\,(\mathsf{qbit}\ y,z,u,v,s,t)\,.\,\{x,y,z,u,v,s,t \ast\!\!= \mathsf{CNot}_{xz}.\mathsf{CNot}_{xy}.\mathsf{CNot}_{(st)z}.\\ \mathsf{CNot}_{(st)y}.\mathsf{CNot}_{(st)x}.\mathsf{CNot}_{zt}.\mathsf{CNot}_{xt}.\mathsf{CNot}_{ys}.\mathsf{CNot}_{xs}.\mathsf{CNot}_{(uv)z}.\mathsf{CNot}_{(uv)y}.\\ \mathsf{CNot}_{(uv)x}.\mathsf{H}_v.\mathsf{H}_u.\mathsf{CNot}_{xy}.\mathsf{CNot}_{xz}\}\,.\,d![x]\,.\,\mathbf{0}\end{aligned} \tag{5.27}$$

Applying QS2, QI1 and QD1 on Eq. 5.27 to give

$$a?[x]\,.\,(\mathsf{qbit}\ y,z,u,v,s,t)\,.\,\{x,y,z,u,v,s,t \ast\!\!= \mathsf{I}\}\,.\,d![x]\,.\,\mathbf{0} \tag{5.28}$$

Then using QI1 on the above equation, we get:

$$a?[x]\,.\,(\mathsf{qbit}\ y,z,u,v,s,t)\,.\,\{x \ast\!\!= \mathsf{I}\}\,.\,\{y,z,u,v,s,t \ast\!\!= \mathsf{I}\}\,.\,d![x]\,.\,\mathbf{0} \tag{5.29}$$

Finally after applying the rules QC10, QS1, QC12 andQS3 on the Eq. 5.29, we arrive at the desired result: $a?[x]\,.\,d![x]\,.\,\mathbf{0}$. □

Proposition 5.5. *Teleport* $\leftrightarrows^c$ *QSS* $\leftrightarrows^c$ *QECC*

Proof. Quantum teleportation (*Teleport*) is a protocol, which allows two users who share an entangled pair of qubits, to exchange an unknown quantum state by communicating only two classical bits. There is a proof in [51] that *Teleport* $\leftrightarrows^c$ *Identity*. We prove the proposition easily by using the *transitivity* of $\leftrightarrows^c$ as we have seen that *QECC* and *QSS* are equivalent to *Identity* through Propositions 5.1 and 5.4. The congruence property helps to analyse a combination of systems. For example, if we consider a process defined as *System* = *Teleport* | *QECC*. We can consider this equivalent to a process *Teleport* | *Identity* by using Proposition 5.4. This is also equivalent to *Identity* | *Identity* which is equivalent to *Identity*. □

5.6 Proof of Soundness of axioms

The equational axioms except for Cv1, Qi3 and Tau1, are proved to be sound in [51]. Now, we will prove the soundness of the new rules that are introduced with respect to full probabilistic branching bisimilarity. This proof holds for any arbitrary quantum states and substitutions.

Lemma 5.6 (Classical rule Cv1)**.** *Let P and Q be two processes that are parameterised by a classical bit x. Then for all values of $x \in \{0,1\}$, we have $P(x) \leftrightarrows^c Q(x)$. Therefore, there exists bisimulations $\mathcal{R}_0$ and $\mathcal{R}_1$ such that $(P(0), Q(0)) \in \mathcal{R}_0$ and $(P(1), Q(1)) \in \mathcal{R}_1$, and we have*

$$c?[x : \mathsf{Bit}] \,.\, P(x) \leftrightarrows^c c?[x : \mathsf{Bit}] \,.\, Q(x)$$

Proof. For any arbitrary states σ, let $\sigma = [\widetilde{p} \mapsto |\psi\rangle]$ and

$$\begin{aligned} s_1 &= (\sigma; \widetilde{p}; c?[x : \mathsf{Bit}] \,.\, P(x)), \\ s_2 &= (\sigma; \widetilde{p}; P(x)), \\ s_3 &= (\sigma; \widetilde{p}; c?[x : \mathsf{Bit}] \,.\, Q(x)), \\ s_4 &= (\sigma; \widetilde{p}; Q(x)) \ . \end{aligned}$$

We define an equivalence relation $\mathcal{R}$ as

$$\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1 \cup \{(s_1, s_3))\} \cup \mathcal{I}$$

where $\mathcal{I}$ is the identity relation. We now prove easily that $\mathcal{R}$ is a probabilistic branching bisimulation by the transitions of s_1 and s_3.

If $s_1 \xrightarrow{c?[x]} s_2$ then we have $s_3 \xrightarrow{c?[x]} s_4$ where $(s_2, s_4) \in \mathcal{R}$. With some formal definitions and inductions, we achieve $P(x)\mathcal{R}Q(x)$. □

Lemma 5.7 (Deferred measurement Qi3)**.** *Assume $x, z \notin \widetilde{y}$. If U is a unitary operator and CU is the corresponding controlled operator then*

$$\{\widetilde{y} *= \mathsf{U}^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P \leftrightarrows^c \{(x,z), \widetilde{y} *= \mathsf{CU}\}.\{\mathsf{measure}\ x\}.\{\mathsf{measure}\ z\}.P.$$

Proof. This is a straightforward adaptation from [51]. Assume that $\kappa = \{p, \widetilde{q}/x, \widetilde{r}/z, \widetilde{y}\}$. Let

$$
\begin{aligned}
s_1 &= ([p\widetilde{q}\widetilde{r} \mapsto |\psi_1\rangle]; p, \widetilde{q}, \widetilde{r}; (\{\widetilde{y} *= U^{\mathsf{measure}\ x.\mathsf{measure}\ z}\}.P)\kappa),\\
s_2 &= \oplus_{i\in I}\ g_i\ ([p\widetilde{q}\widetilde{r} \mapsto |\psi_{2_i}\rangle]; p, \widetilde{q}, \widetilde{r}; \lambda i \bullet (\{\widetilde{y} *= U^{i.\mathsf{measure}\ z}\}.P)\kappa; i),\\
s_3 &= \oplus_{\substack{i\in I\\ j\in J}}\ g_i\ h_{ij}([p\widetilde{q}\widetilde{r} \mapsto |\psi_{3_{ij}}\rangle]; p, \widetilde{q}, \widetilde{r}; \lambda ij \bullet (\{\widetilde{y} *= U^{i.j}\}.P)\kappa; i, j),\\
s_4 &= \oplus_{\substack{i\in I\\ j\in J}}\ g_i\ h_{ij}([p\widetilde{q}\widetilde{r} \mapsto |\psi_{4_{ij}}\rangle]; p, \widetilde{q}, \widetilde{r}; P\kappa),\\
s_5 &= ([p\widetilde{q}\widetilde{r} \mapsto |\psi_1\rangle]; p, \widetilde{q}, \widetilde{r}; (\{(x, z), \widetilde{y} *= CU\}.\{\mathsf{measure}\ x\}.\{\mathsf{measure}\ z\}.P)\kappa),\\
s_6 &= ([p\widetilde{q}\widetilde{r} \mapsto |\psi_5\rangle]; p, \widetilde{q}, \widetilde{r}; (\{\mathsf{measure}\ x\}.\{\mathsf{measure}\ z\}.P)\kappa),\\
s_7 &= \oplus_{i\in I}\ g_i\ ([p\widetilde{q}\widetilde{r} \mapsto |\psi_{6i}\rangle]; p, \widetilde{q}, \widetilde{r}; (\{\mathsf{measure}\ z\}.P)\kappa),\\
s_8 &= \oplus_{\substack{i\in I\\ j\in J}}\ g_i\ h_{ij}([p\widetilde{q}\widetilde{r} \mapsto |\psi_{7_{ij}}\rangle]; p, \widetilde{q}, \widetilde{r}; P\kappa)
\end{aligned}
$$

where $I = \{0, 1\}$ and $J = \{0, 1\}$.

Let M_i and M_j be the measurement operators corresponding to the measurement of x and z respectively. Then $|\psi_{2_i}\rangle = M_i|\psi_1\rangle$, $|\psi_{3_{ij}}\rangle = M_j|\psi_{2_i}\rangle$ and $|\psi_{4_{ij}}\rangle = U^{ij}|\psi_{3_{ij}}\rangle = U^{ij}M_i.M_j|\psi_1\rangle$ and $|\psi_5\rangle = CU|\psi_1\rangle$ and $|\psi_{6i}\rangle = M_i|\psi_5\rangle = M_iCU|\psi_1\rangle$ and $|\psi_{7_{ij}}\rangle = M_j.|\psi_{6i}\rangle$. For each $i \in I$ and $j \in J$, a straightforward calculation shows $U^{ij}M_j.M_i = M_j.M_iCU$, therefore $|\psi_{4_{ij}}\rangle = |\psi_{7_{ij}}\rangle$ and $s_4 = s_8$.

Now define an equivalence relation where

$$\mathcal{R} = \{(s_1, s_5), (s_2, s_6), (s_3, s_7), (s_4, s_8)\} \cup \mathcal{I}\ .$$

We have $s_1 \xrightarrow{\tau} s_2 \xrightarrow{\tau} s_3 \xrightarrow{\tau} s_4$ and $s_5 \xrightarrow{\tau} s_6 \xrightarrow{\tau} s_7 \xrightarrow{\tau} s_8$. Therefore it is straightforward to see that $\mathcal{R}$ is a probabilistic branching bisimulation. □

Lemma 5.8 (Tau rule TAU1)**.**

$$\alpha\,.\,\tau\,.\,P \leftrightarrows^c \alpha\,.\,P$$

Proof. Let σ_1 be an arbitrary quantum state, and let

$$s_1 = (\sigma_1; \omega; (\alpha\,.\,\tau\,.\,P) \text{ and } s_2 = (\sigma_1; \omega; (\alpha\,.\,P))\ .$$

Then define an equivalence relation $\mathcal{R}$ where $\mathcal{R} = \{(s_1, s_2)\}$ Then we have the transition $s_1 \xrightarrow{\tau} s_2$ and $\rho_E(s_1) = \rho_E(s_2)$. □

5.7 Discussion

Davidson [51] has introduced the equational theory in CQP. By using the axioms, he has verified the quantum teleportation protocol. In this chapter, we show that by defining three additional axioms, we have taken a step further in analysing various other quantum protocols like superdense coding, quantum secret sharing, remote CNOT and quantum error correction. It is interesting to note that the new axioms Cv1 and Qi3 were not required in the analysis of the quantum teleportation protocol. This is because the teleportation protocol did not involve processes that are parameterised by classical bits and also did not have operators that are controlled by two qubits. The proofs of the soundness of the axioms, which are defined in this thesis, are presented. The soundness of the other remaining equational laws are not presented in this work as it is given in complete detail in [51].

Verification of the quantum protocols using the bisimulation relations requires hard work. First, we need to perform the computations of the *System* and the *Specification*, and then we need to establish a bisimulation relation based on the requirements of the bisimulation definition. Like for example, the transitions of the processes should be compared according to the type of bisimulation and also checking the reduced density matrices for the resulting states. Because of equational reasoning, we show that we can reduce the need to explicitly construct bisimulation relations.

Abramsky and Coecke [2] have developed an approach for analysing quantum protocols by using the mathematical tools of category theory. Their approach is based on recasting the standard axiomatisation of quantum mechanics by employing category theory to describe the protocols at a more abstract level. They have verified the correctness of quantum teleportation, logic gate teleportation and entanglement swapping. Ying *et al.* [171] defined an automata model for reasoning about information-flow security of quantum systems which provides a quantitative description of quantum information flow.

An important advantage of using the equational axioms is that it helps in automated reasoning. Automated reasoning has been applied to many quantum protocols. Ardeshir-Larijani *et al.* developed a model checking tool [12] called the equivalence checker for the verification of quantum protocols. The tool uses the stabilizer formalism and is restricted to use the operators that are only in the Clifford Group. This puts a limitation in solving problems that cannot be defined by the Clifford group operators.

The equational theory of CQP is not based on the stabiliser formalism and hence not restricted to Clifford group operations. One of the future tasks is to prove the completeness of the axioms. Following from the recent work on equivalence checker, our

long-term goal is to develop an automated tool based on the equational theory of CQP that allows the possibility to verify quantum programs beyond the stabilizer formalism.

Chapter 6

Quantum Process Calculus for Linear Optical Quantum Computing

In this chapter, we describe the use of quantum process calculus (CQP) to model a realistic experimental system that demonstrates quantum computing such as linear optical quantum computing (LOQC). We begin by providing the foundations to understand linear optics and then extending CQP to describe the basic linear optical elements that are used in LOQC. In all previous work on quantum process calculus, we have seen that a qubit was considered as a *localised* unit of information. Now, we present our first attempt to model realistic ideal systems and the associated experimental processes.

6.1 Linear optical quantum computing (LOQC)

Many different architectures for quantum computers based on different physical systems have been proposed. These include atom/ion trap quantum computing, nuclear magnetic resonance, nuclear spin quantum computing and optical quantum computing. A detailed review is provided in [156]. Each of these systems has its own advantages and disadvantages. For example, in an ion trap, two qubit gates are relatively easy to implement but isolating the ions from the environment is difficult. This is due to the motion of ions being susceptible to decoherence.

We focus our attention on optical quantum computing which uses single photons as qubits and energy-preserving optical elements (linear optics). A photon is an elementary particle or called the quantum of light. Optical implementations offer to date the most

advanced system for quantum information processing (QIP), and photons naturally allow to integrate quantum computation and quantum communication. Photons possess large coherence times and can easily be generated, manipulated and detected thereby making them suitable candidates for computation and communications. The downside is that photons do not naturally interact with each other, and in order to apply two-bit quantum gates such interactions are essential.

LOQC is one potential way for implementing small-scale quantum computing [106]. The basic building blocks of linear optics are beam splitters, phase shifters and detectors. The difficulty in using these elements in the experiments is in the alignment in order to ensure interference of photons. In order to overcome these drawbacks, linear optical circuits can be miniaturised using optical fibre and integrated waveguide circuits. These waveguide circuits follow the same principle as that of the macroscopic laboratory setups and have more stability [141]. The computation is based on *spatial encoding* where a quantum bit is represented by two optical or spatial modes containing a single photon. Precise manipulation of the quantum information inscribed in the internal (polarisation) and external (path) states of a photon are routinely achieved using linear optical elements [134].

In the following paragraphs we will provide the basic concepts which will be used in the subsequent sections when we discuss LOQC.

Qubits and modes A *qubit* or a *quantum bit* is one of the fundamental unit of quantum information processing (QIP). We have seen in Chapter 3 that a qubit is represented by the quantum state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ where α_0 and α_1 are complex values. The state $|\psi\rangle$ is a superposition of the basis states $|0\rangle$ and $|1\rangle$ having the respective amplitudes of probability, $|\alpha_0|^2$ to be in $|0\rangle$ state and $|\alpha_1|^2$ to be in $|1\rangle$ state.

In LOQC, a qubit is represented by a single photon where the states $|0\rangle$ and $|1\rangle$ could represent the polarisation state of a photon (i.e. $|0\rangle = |H\rangle$ and $|1\rangle = |V\rangle$ where H and V are the respective horizontal and vertical polarisations of the photon). We can then write the state $|\psi\rangle$ as:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0|H\rangle + \alpha_1|V\rangle \tag{6.1}$$

We refer to the qubit represented by Eq. (6.1) as a *polarisation qubit*. Here polarisation is a distinguishable property of the qubit. In general, we say that any distinguishable property of a photon is defined as a *mode* and the two most common examples for a mode in LOQC that we concentrate in this thesis are *polarisation* and *spatial path* traversed by a photon. A qubit in LOQC has generally the choice of two different modes [107].

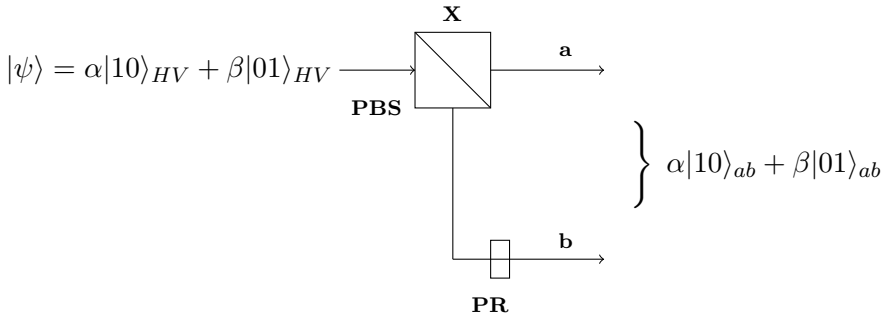


FIGURE 6.1: Conversion of a polarisation qubit to a spatially encoded qubit by using the linear optical elements, polarisation beam splitter (PBS) and phase shifter (PR). X is the unused port of PBS, a and b are the optical paths.

For example, we say that the state $|H\rangle$ is equivalent to the state $|1\rangle_H|0\rangle_V$, which means a state with 1 photon in polarisation mode H and 0 photon in polarisation mode V.

Fock States. We use the notation $|n\rangle_x$ for a state that represents the number of photons n (where $n = 0, 1, 2...$) of the given optical mode x (i.e indicated by the subscript). We refer this as the number states or *Fock states*. The standard basis of Fock states ($|n\rangle$) are $|0\rangle, |1\rangle, |2\rangle$ and so on. The general quantum state represented in Eq. (6.1) can be considered as a linear combination of these basis states given by:

$$\alpha_0|H\rangle + \alpha_1|V\rangle = \alpha_0|10\rangle_{HV} + \alpha_1|01\rangle_{HV} \tag{6.2}$$

where the entries in the ket states on the RHS of Eq. (6.2) represent the number of photons and the subscripts indicate the optical *mode*, which in this case is polarisation. Using the above concepts, we can generalise the notation to more than one photon. For example, two photons can then be encoded in polarisation mode given by $\alpha|20\rangle_{HV} + \beta|02\rangle_{HV} + \gamma|11\rangle_{HV}$, if they are indistinguishable in all other parameters.

Spatial Encoding. Apart from polarisation, qubits can also be considered to be encoded in different optical paths 'a' and 'b' in LOQC [106]. This is known as *spatial encoding* also referred to as *dual rail logic*. Here, we denote the entries in the kets as the number of photons travelling along the different paths (i.e. indicated in the subscripts). The basis states in dual rail logic are then $|0\rangle = |1\rangle_a|0\rangle_b \equiv |10\rangle_{ab}$, and similarly for $|1\rangle = |0\rangle_a|1\rangle_b \equiv |01\rangle_{ab}$. Therefore, $|10\rangle_{ab}$ means 1 photon travelling in path a and no photon in path b.

Coding Conversion. In experiments, the conversion of a *polarisation* qubit into a *dual rail* qubit is accomplished by the combination of a polarising beam splitter (PBS)

and a phase shifter (PR) as shown in Figure 6.1. The PBS has two input ports and two output ports, where the unused input port is denoted by X. The superposition of $|H\rangle$ and $|V\rangle$ is converted into a superposition of paths a and b. The polarising beam splitter changes the path of the incident photon if it is vertically (V) polarised. That is, the photon is reflected and comes out of PBS in a direction perpendicular to the original path of the photon. But, a horizontal polarised (H) photon would come out of the PBS in the same direction or we say that the photon is transmitted. The PBS therefore links polarisation information with path information. A subsequent phase shifter (PR) rotates the polarisation of the vertical output by 90° so that the components of the dual rail qubit are indistinguishable in their polarisations and can interfere [134]. We define the combined operation of PBS and PR as a unitary operation, PS, which converts a polarisation encoded qubit into a dual rail or spatial encoded qubit.

Definition 6.1 (PS operator)**.** A PS is an operator that transforms a polarisation qubit $|\psi\rangle \in \mathbb{H}_1$ to a dual rail qubit $|\phi\rangle \in \mathbb{H}_2$ represented by spatial modes (a, b).The action of PS is defined by

$$\mathsf{PS}|H\rangle = |10\rangle_{ab} \text{ and } \mathsf{PS}|V\rangle = |01\rangle_{ab}$$

The evolution of a lossless closed quantum system can be described by *unitary transformations* that is performed by the linear optical elements such as phase shifters and beam splitters. The total photon number is preserved by these transformations. If the state of a qubit is represented by a column vector then a unitary transformation U can be represented by a matrix.

Transformation on Fock states. Operations on number states or *Fock states* ($|n\rangle$) are described in terms of the *creation* operator ($\hat{a}^\dagger$) and *destruction* operator ($\hat{a}$), which satisfy the following commutation relations:

$$[\hat{a}_i, \hat{a_j}^\dagger] = \delta_{i,j}$$

$$[\hat{a}_i, \hat{a_j}] = [\hat{a_i}^\dagger, \hat{a_j}^\dagger] = 0$$

The action of the abive operators on the number states $|n\rangle$ increase or decrease the photon number (n) by one. This is given as:

$$\begin{aligned} \hat{a}|n\rangle &= \sqrt{n}|n-1\rangle \\ \hat{a}^\dagger|n\rangle &= \sqrt{n+1}|n+1\rangle. \end{aligned}$$

In this scheme, $|0\rangle$ corresponds to the vacuum state (i.e. absence of photon), $|1\rangle$ corresponds to single photon and so on. Therefore, each Fock state can be built up from creation operators given by

$$|n\rangle = \frac{(\hat{a}^\dagger)^n}{\sqrt{n!}}|0\rangle.$$

Post-selection. This plays a vital role in LOQC, where one considers only a subset of all experimental runs that fulfil predefined criteria, e.g. given by the desired number of detected photons in particular channels. Therefore the computation succeeds with a certain probability, and with the complementary probability it is aborted with no result. We describe post-selection in CQP by modelling a linear optical CNOT gate.

6.1.1 Unitary transformation in LOQC

An optical component is defined to be *linear* if the output mode operators of the component are a linear combination of its input mode operators [128]. If $\hat{b}_j^\dagger$ are the output mode operators and $\hat{c}_k^\dagger$ are the input mode operators, then

$$\hat{b}_j^\dagger \quad = \quad \textstyle\sum_k M_{jk}\hat{c}_k^\dagger \tag{6.3}$$

A unitary transformation in LOQC [128] can be described by its effect on each mode's creation operator. The basic and common linear optical components are phase shifters and beam splitters.

A non polarising beam splitter (BS) is defined by the transformation matrix [107, 146]

$$U(BS) \quad = \quad \begin{pmatrix} \cos\theta & e^{i\phi}\sin\theta \\ e^{-i\phi}\sin\theta & -\cos\theta \end{pmatrix} \tag{6.4}$$

where the input mode operators ($\hat{c}^\dagger$) are related to the output mode operators ($\hat{b}^\dagger$) by

$$(\hat{b}_j^\dagger)^n|0\rangle \quad = \quad \textstyle\sum_k U_{jk}(\hat{c}_k^\dagger)^n|0\rangle. \tag{6.5}$$

The reflectivity of BS is given by $\eta = \cos^2\theta$, where $\cos\theta$ and $\sin\theta$ are the probability amplitudes for reflection and transmission, and ϕ is the relative phase. Here we consider $\phi = 0$, which is the case for BSs in integrated circuits.

If we consider the state $|mn\rangle_{pq}$ incident on a beam splitter with m photons along path p and n photons along path q, the transformation is [128]:

$$|mn\rangle_{pq} = \frac{(\hat{a}_p^\dagger)^m}{\sqrt{m!}}\frac{(\hat{a}_q^\dagger)^n}{\sqrt{n!}}|00\rangle_{pq} = \frac{1}{\sqrt{m!n!}}(\hat{a}_p^\dagger\cos\theta + \hat{a}_q^\dagger\sin\theta)^m(\hat{a}_p^\dagger\sin\theta - \hat{a}_q^\dagger\cos\theta)^n|00\rangle_{pq}. \tag{6.6}$$

For example, we get:

$$\begin{aligned}
|10\rangle &\rightarrow \cos\theta|10\rangle + e^{-i\phi}\sin\theta|01\rangle \\
|01\rangle &\rightarrow e^{i\phi}\sin\theta|10\rangle - \cos\theta|01\rangle \\
|11\rangle &\rightarrow \sqrt{2}e^{i\phi}\cos\theta.\sin\theta|20\rangle + (\sin^2\theta - \cos^2\theta)|11\rangle - \sqrt{2}e^{-i\phi}\cos\theta.\sin\theta|02\rangle \\
|20\rangle &\rightarrow \cos^2\theta|20\rangle + \sqrt{2}e^{-i\phi}\cos\theta.\sin\theta|11\rangle + e^{-2i\phi}\sin^2\theta|02\rangle \\
|02\rangle &\rightarrow e^{2i\phi}\sin^2\theta|20\rangle - \sqrt{2}e^{i\phi}\cos\theta.\sin\theta|11\rangle + \cos^2\theta|02\rangle
\end{aligned}$$

In general, for a beam splitter of reflectivity $\frac{1}{2}$ (assuming $\phi = 0$), the transformation matrix is given as:

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

The output modes of this beam splitter are then given as:

$$\begin{aligned}
|10\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \\
|01\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(|10\rangle - |01\rangle) \\
|11\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(|20\rangle - |02\rangle) \\
|20\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(\tfrac{1}{\sqrt{2}}|20\rangle + |11\rangle + \tfrac{1}{\sqrt{2}}|02\rangle) \\
|02\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(\tfrac{1}{\sqrt{2}}|20\rangle - |11\rangle + \tfrac{1}{\sqrt{2}}|02\rangle)
\end{aligned}$$

Similarly for a beam splitter of reflectivity $\frac{1}{3}$ with $\cos\theta = \frac{1}{\sqrt{3}}$, we get the transformation matrix as:

$$U = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \\ \frac{\sqrt{2}}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{pmatrix}.$$

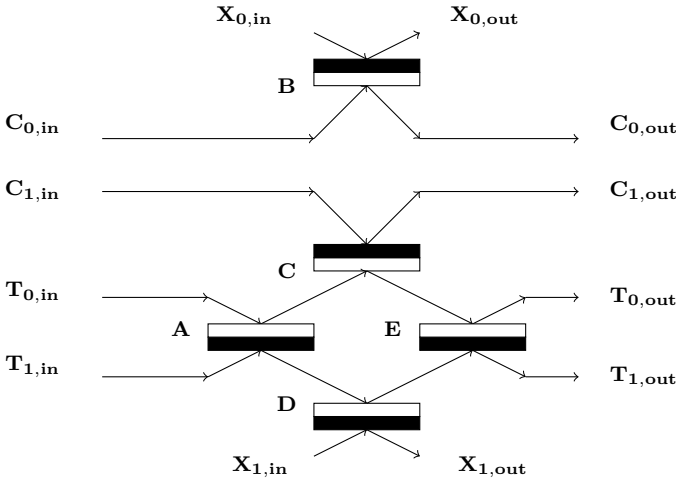


FIGURE 6.2: LOQC CNOT Gate. A sign change occurs upon reflection of the optically thicker side (indicated in black) of the BSs.

The output modes are given as:

$$
\begin{aligned}
|10\rangle &\rightarrow \tfrac{1}{\sqrt{3}}|10\rangle + \tfrac{\sqrt{2}}{\sqrt{3}}|01\rangle \\
|01\rangle &\rightarrow \tfrac{\sqrt{2}}{\sqrt{3}}|10\rangle - \tfrac{1}{\sqrt{3}}|01\rangle \\
|11\rangle &\rightarrow \tfrac{1}{3}(2|20\rangle + |11\rangle - 2|02\rangle) \\
|20\rangle &\rightarrow \tfrac{1}{3}(|20\rangle + 2|11\rangle + 2|02\rangle) \\
|02\rangle &\rightarrow \tfrac{1}{3}(2|20\rangle - 2|11\rangle + |02\rangle)
\end{aligned}
$$

Using the transformations of the above beam splitters, we will discuss the operation of the LOQC CNOT gate in the next section.

6.1.2 Working of LOQC CNOT gate

We consider the CNOT gate of O' Brien *et. al* [134, 141], depicted in Figure 6.2, which is an implementation of the gate proposed by Ralph, Langford, *et al.* [146]. This is a postselected two-photon gate where two polarised qubits are created in a spontaneous parametric down-conversion (also known as SPDC). This is used especially to create entangled photon pairs, and of single photons. To perform this task, a laser beam is incident on a non linear crystal which is used to split photons into pairs of photons that have combined energy and momentum equal to the energy and momentum of the original photon. The polarisation qubits can be converted to dual-rail qubits with the

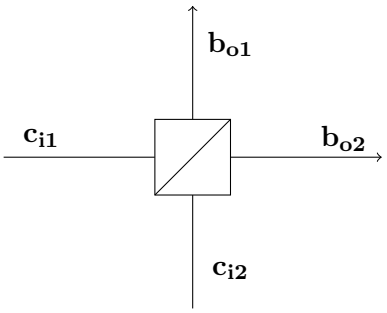


FIGURE 6.3: The beam splitter

help of a polarising beam splitter and a phase shifter combination. Both the control and target qubits can be prepared in an arbitrary pure superposition of the computational basis states. The LOQC CNOT gate shown in Figure 6.2 is a combination of five beam splitters (BSs) A, B, C, D and E.

Experiment: In [134], the experiment is analysed as follows: if the control qubit is in the state where the photon enters the top input port $C_{0,inp}$, there is no interaction between the control and target qubits. On the other hand, when the control photon enters the lower input port $C_{1,inp}$, the control and target photons interfere non classically at the central beam splitter giving two photon interference which causes a π phase shift in the upper arm of the target $T_{0,inp}$, and as a result the target photon is switched from one output mode to the other. Otherwise we can say that, the target state experiences a bit flip. The control qubit remains unaffected, hence the interpretation of this experiment as a CNOT gate. We do not always observe a single photon in each of the control and target outputs. But, when a control and a target photon are detected we know that the CNOT operation has been correctly realised.

Theory: All the beam splitters are assumed to be asymmetric in phase. This means that if $\hat{c}^\dagger_{i1}$ and $\hat{c}^\dagger_{i2}$ are the two input mode operators and $\hat{b}^\dagger_{o1}$ and $\hat{b}^\dagger_{o2}$ are the corresponding output operators as shown in Figure 6.3, then the relations between the input and output operators are given by:

$$\begin{aligned} \hat{b}^\dagger_{o1} &= \cos\theta \hat{c}^\dagger_{i1} + \sin\theta \hat{c}^\dagger_{i2} \\ \hat{b}^\dagger_{o2} &= \sin\theta \hat{c}^\dagger_{i1} - \cos\theta \hat{c}^\dagger_{i2} \end{aligned} \tag{6.7}$$

Eq. (6.7) is obtained by using Eq. (6.5) and the unitary matrix of beam splitter given by Eq. (6.4).

The theory and operation of the gate are provided in [146] and are used here for our understanding. Beam splitters A and E are of reflectivity $\frac{1}{2}$ and the rest B, C and D are of reflectivity $\frac{1}{3}$. X indicates the respective input port of the BS that is not used and also

the photons coming out of output ports $X_{1,out}$ and $X_{0,out}$ are not considered. $C_{0,in}, C_{1,in}$ are the control (C) input ports and $T_{0,in}, T_{1,in}$ are the target (T) input ports. The output ports are $C_{0,out}, C_{1,out}, T_{0,out}, T_{1,out}$. The relationships between the corresponding input and output operators are the following:

$$\begin{aligned}
\hat{c}^\dagger_{0,out} &= \tfrac{1}{\sqrt{3}}(\sqrt{2}\hat{x}^\dagger_{0,in} + \hat{c}^\dagger_{0,in}) \\
\hat{c}^\dagger_{1,out} &= \tfrac{1}{\sqrt{3}}(-\hat{c}^\dagger_{1,in} + \hat{t}^\dagger_{0,in} + \hat{t}^\dagger_{1,in}) \\
\hat{t}^\dagger_{0,out} &= \tfrac{1}{\sqrt{3}}(\hat{c}^\dagger_{1,in} + \hat{t}^\dagger_{0,in} + \hat{x}^\dagger_{1,in}) \\
\hat{t}^\dagger_{1,out} &= \tfrac{1}{\sqrt{3}}(\hat{c}^\dagger_{1,in} + \hat{t}^\dagger_{1,in} - \hat{x}^\dagger_{1,in}) \\
\hat{x}^\dagger_{0,out} &= \tfrac{1}{\sqrt{3}}(-\hat{x}^\dagger_{0,in} + \sqrt{2}\hat{c}^\dagger_{0,in}) \\
\hat{x}^\dagger_{1,out} &= \tfrac{1}{\sqrt{3}}(\hat{t}^\dagger_{0,in} - \hat{t}^\dagger_{1,in} - \hat{x}^\dagger_{1,in})
\end{aligned} \tag{6.8}$$

Consider the general two photon input state given by Eq. (6.9), where the notation e.g. $|HV\rangle$ refers to a control photon with horizontal polarisation and a vertically polarised target photon

$$\begin{aligned}
|\phi\rangle &= (\alpha|HH\rangle + \beta|HV\rangle + \gamma|VH\rangle + \delta|VV\rangle)|00\rangle \\
&= (\alpha\hat{c}^\dagger_{0,in}\hat{t}^\dagger_{0,in} + \beta\hat{c}^\dagger_{0,in}\hat{t}^\dagger_{1,in} + \gamma\hat{c}^\dagger_{1,in}\hat{t}^\dagger_{0,in} + \delta\hat{c}^\dagger_{1,in}\hat{t}^\dagger_{1,in})|0000\rangle|00\rangle
\end{aligned} \tag{6.9}$$

where the ordering in the kets is $|c_0c_1t_0t_1\rangle|x_0x_1\rangle$. Here c_0, c_1 are the number states for the control qubit, t_0, t_1 are for the target qubit and x_0, x_1 are the vacuum states and we use the shorthand $|1010\rangle = |HH\rangle$, etc., where appropriate. Using the operators as discussed in Eq. (6.8) and applying it to Eq. (6.9) by substituting input operators for the output operators, we get the number of photons in the respective output ports ($C_{1,out}, C_{0,out}, T_{1,out}, T_{0,out}, X_{1,out}$ and $X_{0,out}$) of the CNOT gate as shown in Figure 6.2.

$$\begin{aligned}
|\phi\rangle_{out} = (\alpha\hat{c}^\dagger_{0,out}\hat{t}^\dagger_{0,out} + \beta\hat{c}^\dagger_{0,out}\hat{t}^\dagger_{1,out} + \gamma\hat{c}^\dagger_{1,out}\hat{t}^\dagger_{0,out} + \delta\hat{c}^\dagger_{1,out}\hat{t}^\dagger_{1,out})|0000\rangle|00\rangle \\
= \tfrac{1}{3}\{(\alpha|HH\rangle + \beta|HV\rangle + \gamma|VV\rangle + \delta|VH\rangle)|00\rangle + \sqrt{2}(\alpha+\beta)|0100\rangle|10\rangle + \\
\sqrt{2}(\alpha-\beta)|0000\rangle|11\rangle + (\alpha+\beta)|1100\rangle|00\rangle + (\alpha-\beta)|1000\rangle|01\rangle + \sqrt{2}\alpha|0010\rangle|10\rangle + \\
\sqrt{2}\beta|0001\rangle|10\rangle - \sqrt{2}(\gamma+\delta)|0200\rangle|00\rangle - (\gamma-\delta)|0100\rangle|01\rangle + \sqrt{2}\gamma|0020\rangle|00\rangle \\
+(\gamma-\delta)|0010\rangle|01\rangle + (\gamma+\delta)|0011\rangle|00\rangle + (\gamma-\delta)|0001\rangle|01\rangle + \sqrt{2}\delta|0002\rangle|00\rangle\}.
\end{aligned} \tag{6.10}$$

From these states we post-select only those where one photon is found in the target and one in the control state, giving

$$|\phi\rangle_{ps} = \alpha|HH\rangle + \beta|HV\rangle + \gamma|VV\rangle + \delta|VH\rangle. \tag{6.11}$$

Because of the BSs (that have reflectivity $\frac{1}{3}$), we essentially do not always have a single photon in each of the control and target outputs. But, when a single photon is detected at each of the outputs, it is recorded as a coincidence count that occurs with a probability

$$
\begin{array}{rcl}
T & ::= & \mathsf{Int} \mid \mathsf{Qbit} \mid \mathsf{NS} \mid \widehat{\ }[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v & ::= & 0 \mid 1 \mid \cdots \mid \mathsf{H} \mid \mathsf{PS} \mid \cdots \\
e & ::= & v \mid x \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} *\!= e \mid e + e \mid x : \mathsf{NS}, y : \mathsf{NS} *\!= \mathsf{PS}(z) \\
P & ::= & \mathbf{0} \mid (P|P) \mid P + P \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid [e].P \mid (\mathsf{qbit}\ x)P \mid (\mathsf{ns}\ x)P \mid \\
 & & (\mathsf{new}\ x : \widehat{\ }[T])P \mid \mathsf{if}\ \ e\ \ \mathsf{then}\ \ P\ \ \mathsf{else}\ \ Q
\end{array}
$$

FIGURE 6.4: Syntax of CQP.

of one-ninth and the relationship between Eq. (6.10) and Eq. (6.11) is a controlled-NOT transformation.

With the understanding of the above theory and operation of the gate, we provide the extensions of CQP in the next section to describe LOQC.

6.2 Extensions of CQP for LOQC

Modelling in CQP provides us an abstract view of the quantum system. Our aim is to model realistic (non-localised) systems and the associated experimental processes. CQP assumes that a qubit is a *localised* unit of information. This view works well with QKD but not with LOQC, as it cannot describe *spatial encoding*. In this section we extend CQP in order to model LOQC. We illustrate this by defining various linear optical elements such as beam splitters and phase shifters in CQP and by modelling an LOQC CNOT gate.

6.2.1 Syntax

The syntax of CQP for LOQC is defined by the grammar as shown in Figure 6.4. This is very similar to the previous version of CQP as shown in Figure 4.1. The framework of CQP helps us to extend or generalise the language to be suited for other applications. The description of the syntax is the same as discussed in the earlier chapters which consists of types T, values v, expressions e (including quantum measurements and the conditional application of unitary operators $\tilde{e} *\!= e^e$), and processes P. We have a new type called NS for number state. Values v consist of variables (x,y,z etc), literal values of data types (0,1,..), and a new unitary operator PS which is provided by the Definition 6.1. An important addition to the expression is the unitary operation of PS that converts a polarisation qubit, say z, to spatially encoded number states x and y. Processes now include the following $[e].P$ (typically for PS operation), **if** . . . **else** conditions and number

state declaration $(\mathsf{ns}\ x)P$, that are needed for the extension of the language to model LOQC.

In order to define the operational semantics we provide the *internal syntax* in Figure 6.5.

$$
\begin{array}{lcl}
v & ::= & \ldots \mid q \mid s \mid c \\
E & ::= & [] \mid \mathsf{measure}\ E, \tilde{e} \mid \mathsf{measure}\ v, E, \tilde{e} \mid \ldots \mid \mathsf{measure}\ \tilde{v}, E \mid E + e \mid v + E \\
F & ::= & []?[\tilde{x}].P \mid []![\tilde{e}].P \mid v![[].\tilde{e}].P \mid v![v, [], \tilde{e}].P \mid \cdots \mid v![\tilde{v}, []].P \mid \{[]\}.P
\end{array}
$$

FIGURE 6.5: Internal syntax of CQP.

The addition to the values are the number state names s that are generated at run-time and substituted for the variables used in ns declarations respectively.

6.2.2 Type System for LOQC

In this section, we introduce the extensions of the type system of CQP that is presented in the previous chapter (section 4.1.2) and [51]. This is a straight forward approach as we introduce the number states (NS). The modified typing rules for the syntax defined in Figure 6.4 are shown in Figure 6.6. Environments Γ are mappings from variables to types in the usual way.

Definition 6.2 (Addition of Environments)**.** [51, 75] The partial operation of adding a typed variable to an environment, $\Gamma + x{:}T$, is defined by

$$
\Gamma + x{:}T = \begin{cases} \Gamma, x{:}T & \text{if } x \notin dom(\Gamma) \\ \Gamma & \text{if } T \neq (\mathsf{Qbit}, \mathsf{NS}) \text{ and } x{:}T \in \Gamma \\ \text{undefined} & \text{otherwise.} \end{cases}
$$

6.2.3 Linear optical elements in CQP

We have seen earlier that the combination of a PBS and PR converts a polarisation qubit to a dual rail qubit as shown in Figure 6.1. We define the combination as a process *PolSe* which provides the input to the LOQC CNOT gate.

$$
\begin{array}{rcl}
PolSe(a{:}\hat{}[\mathsf{Qbit}], c{:}\hat{}[\mathsf{NS}], d{:}\hat{}[\mathsf{NS}]) & = & a?[q_0 : \mathsf{Qbit}]\,.\,[s_0 : \mathsf{NS}, s_1 : \mathsf{NS} *{=} \mathsf{PS}(q_0)] \\
& & .\,c![s_0]\,.\,d![s_1]\,.\,\mathbf{0}
\end{array}
$$

$$\frac{\forall i(\Gamma \vdash x_i : \mathsf{NS}) \quad x_1, \ldots, x_n \text{ distinct}}{\Gamma \vdash \mathsf{measure}\ x_1, \ldots, x_n : \mathsf{Int}} \qquad \text{(T-Msure-Ns)}$$

$$\frac{\Gamma, x : \mathsf{ns} \vdash P}{\Gamma \vdash (\mathsf{ns}\ x)P} \qquad \text{(T-NS)}$$

$$\frac{\begin{array}{c}\Gamma \vdash x : \hat{}[T_1, \ldots, T_m, \mathsf{NS}, \ldots, \mathsf{NS}] \quad \forall i.(T_i \neq \mathsf{NS}) \\ \forall i.(\Gamma \vdash e_i : T_i) \quad y_i \text{ distinct} \quad \Gamma \vdash P\end{array}}{\Gamma, y_1 : \mathsf{NS} \ldots, y_n : \mathsf{NS} \vdash x![e_1, \ldots, e_m, y_1, \ldots, y_n].P} \qquad \text{(T-Out)}$$

$$\frac{\forall i(\Gamma \vdash x_i : \mathsf{NS}) \quad x_1 \ldots x_n \text{ distinct} \quad \Gamma \vdash U : \mathsf{Op}(n) \quad \Gamma \vdash e : \mathsf{Int} \quad \Gamma \vdash P}{\Gamma \vdash x_1, \ldots, x_n \mathrel{*}= U^e : \mathsf{Unit}} \qquad \text{(T-Trans)}$$

Figure 6.6: Modified typing rules for the syntax of CQP needed for LOQC.

PolSe is parameterized by three channels, a,c and d. The polarisation qubit (say q_0) is received through channel a whose type is $\hat{}[\mathsf{Qbit}]$. The qubit q_0 will be encoded in terms of the number of photons (s_0 and s_1) travelling along channels c and d respectively.

The right hand side of the definition specifies the behaviour of the process *PolSe*. The first term, $a?[q_0 : \mathsf{Qbit}]$ specifies that the qubit is received from channel a and given the local name q_0. The following sequence of terms, separated by dots, indicate temporal sequencing. The term $[s_0 : \mathsf{NS}, s_1 : \mathsf{NS} \mathrel{*}= \mathsf{PS}(q_0)]$ specifies that the PS operation is applied to qubit q_0 thereby generating s_0 and s_1 of type number states (NS). PS corresponds to the transformation produced by the combination of PBS and PR, introduced by Definition 6.1. The last two terms ($c![s_0]$ and $d![s_1]$) indicate that the respective values of the number states are sent through the respective output channels. The term $\mathbf{0}$ simply indicates termination.

The CQP definition of the beam splitter *BS* is

$$BS(e : \hat{}[\mathsf{NS}], f : \hat{}[\mathsf{NS}], h : \hat{}[\mathsf{NS}], i : \hat{}[\mathsf{NS}], \eta) \quad = \quad e?[s_2 : \mathsf{NS}] \,.\, f?[s_3 : \mathsf{NS}] \,.\, \{s_2, s_3 \mathrel{*}= \mathsf{B}_\eta\} \,. \\ h![s_2] \,.\, i![s_3] \,.\, \mathbf{0}$$

where η is the reflectivity. Process *BS* has input channels e and f, and output channels h and i, all of type $\hat{}[\mathsf{NS}]$. After receiving inputs s_2 and s_3 from e and f, the unitary operation of *BS* represented by $\{s_2, s_3 \mathrel{*}= \mathsf{B}_\eta\}$ is carried out on the input number states as defined by Eq. 6.6. Here B_η is the unitary operation represented by the matrix $U(BS)$ for $\phi = 0$. The number states are then output on h and i.

Finally, we define the process *Det* which encapsulates measurement of a number state as a detector component. This will be used for the post-selecting measurement of the

outputs of the CNOT gate.

$$Det(l{:}\hat{}[\mathsf{NS}], u{:}\hat{}[\mathsf{Val}]) \quad = \quad l?[s_0 : \mathsf{NS}] \,.\, u![\mathsf{measure}\ s_0] \,.\, \mathbf{0}$$

The expression $\mathsf{measure}\ s_0$ probabilistically evaluates to a positive integer which is the number of photons detected.

6.2.4 Semantics

In this section we will explain the operational semantics of CQP. Our focus is to provide the new additions to the previously defined formal syntax and semantics of CQP [51], in order to describe the behaviour of the linear optical CNOT gate. We have seen that the execution of a system is not completely described by the process term (which is the case for classical process calculus) but also depends on the quantum state. Hence the operational semantics are defined using *configurations*, which represent both the quantum state and the process term.

In LOQC, a qubit can be encoded with respect to the polarisation of a photon which we denote as Qbit and also it can be encoded with respect to the path traversed by the photon which we denote as NS. Therefore, our quantum state now can comprise both Qbit and NS. Hence, we have an additional term in our configuration which gives the list of elements and their types that are associated to the quantum state.

Definition 6.3 (Configuration)**.** A configuration is defined as a tuple of the form $(\tilde{x} : \tilde{T}; \sigma; \omega; P)$ where $\tilde{x}$ is a list of names (qubits $\tilde{q}$, number states $\tilde{s}$ or both) associated with their types $\tilde{T}$, σ is a mapping from names $(\tilde{x})$ to the quantum state and ω is a list of names associated with the process P

We operate with configurations such as

$$(q_1 : \mathsf{Qbit}, s_0 : \mathsf{NS}, s_1 : \mathsf{NS}; [q_1, s_0, s_1 \mapsto (|0\rangle|10\rangle + |1\rangle|01\rangle)]; q_1; c![q_1] \,.\, P)$$

We interpret the NS variables as dual-rail representations of qubits, which were in the initial configuration. For example, in this case, s_0 and s_1 represent the original qubit q_0. There is a fixed relationship between the indices of qubits and number state variables: q_i is represented by s_{2i}, s_{2i+1}. There may be additional NS variables, introduced by the ns declarations, representing vacuum states. This configuration means that the global quantum state consists of a qubit, q_1, number states s_0 and s_1, in the specified state; that the process term under consideration has access to qubit q_1 but not to the number states; and that the process itself is $c![q_1] \,.\, P$. The suffix is important as it not only

$$(\tilde{x} : \tilde{T}; \sigma; \omega; u + v) \longrightarrow_v (\tilde{x} : \tilde{T}; \sigma; \omega; w) \text{if } u \text{ and } v \text{ are integer literals and } w = u + v \qquad \text{(R-Plus)}$$

$$(\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s}} \alpha_{\tilde{s}} |\beta\rangle|\gamma\rangle]; \omega; \mathsf{measure}\ s_r) \longrightarrow_v \qquad \text{(R-Measure-NS)}$$

$$\boxplus_{u \geq 0} p_u \bullet (\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s'}} \frac{\alpha_{\tilde{s'}}}{p_u} |\beta\rangle|\gamma'\rangle]; \omega; u) \text{ where } p_u = \sum_{\tilde{i}} |\alpha_{\tilde{s'}}|^2,$$

$$\tilde{s} = s_0, \ldots, s_{2n-1}, \tilde{s'} = s_0, \ldots, s_{r-1}, u, s_{r+1}, \ldots, s_{2n-1}, \tilde{i} = s_0, \ldots, s_{r-1}, s_{r+1}, \ldots, s_{2n-1}$$

$$(\tilde{q} : \mathsf{Qbit}, \tilde{s} : \mathsf{NS}; [\tilde{q}, \tilde{s} \mapsto |\beta\rangle|\gamma\rangle]; \omega; s_0, \ldots, s_{2r-1} *= U) \longrightarrow_v \qquad \text{(R-Trans-NS)}$$

$$(\tilde{q} : \mathsf{Qbit}, \tilde{s} : \mathsf{NS}; [\tilde{q}, s_0, \ldots, s_{2n-1} \mapsto |\beta\rangle(U \otimes I_{(n-r)})|\gamma\rangle]; \omega; \mathsf{unit})$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega; e) \longrightarrow_v \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega_i; e_i)}{(\tilde{x} : \tilde{T}; \sigma; \omega; E[e]) \longrightarrow_e \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega_i; E[e_i])} \qquad \text{(R-Context)}$$

Figure 6.7: Transition rules for values and expressions.

indicates the position of the qubit or number state in the quantum state but it shows the relationship between them, that is $q_i = s_{2i}, s_{2i+1}$. Here the configuration shows that the number states s_0 and s_1 are associated with the qubit q_0 which in this case is not accessible by the process term.

For the evaluation of expressions we have *expression configurations* $(\tilde{x} : \tilde{T}; \sigma; \omega; e)$, which are similar to configurations, but include an expression in place of the process. The semantics of expressions is defined by the reduction relations $\longrightarrow_v$ (on values) and $\longrightarrow_e$ (on expressions), given in Figure 6.7. Rules R-Plus, R-Measure-NS and R-Trans-NS deal with the evaluation of terms that result in values, including measurement which produces a probabilistic distribution over the possible measurement outcomes u, and unitary transformations which result in literal unit. The rules R-Measure-NS and R-Trans-NS are the new rules which are added to the semantics to operate with number states. R-Trans-NS defines the action of the unitary operators that operate on number states listed first in the state.

Example 6.1.

$(q_1 : \mathsf{Qbit}, s_0 : \mathsf{NS}, s_1 : \mathsf{NS}; [s_0, s_1, q_1 \mapsto \alpha_1|10\rangle|0\rangle + \alpha_0|01\rangle|1\rangle]; s_0, s_1; \{s_0, s_1 *= \mathsf{B}_{\frac{1}{2}}\}\,.\,P) \xrightarrow{\tau}$

$(q_1 : \mathsf{Qbit}, s_0 : \mathsf{NS}, s_1 : \mathsf{NS}; [s_0, s_1 \mapsto \frac{1}{\sqrt{2}}((\alpha_1 + \alpha_0)|10\rangle|0\rangle + (\alpha_1 - \alpha_0)|01\rangle|1\rangle)]; s_0, s_1; P).$

The above example shows the effect of the unitary operation of a beam splitter (reflectivity $\eta = \frac{1}{2}$) on the number states s_0 and s_1. The important aspect of R-Trans-NS and R-Measure-NS is the effect they have on the quantum state. R-Measure-NS is a rule defined for the measurement of number states.

$$\boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i) \xrightarrow{p_i} (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i) \quad \text{(L-Prob)}$$

$$(\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{v}; c![\tilde{v}].P) \xrightarrow{c![\tilde{v}]} (\tilde{x} : \tilde{T}; \sigma; \omega; P) \quad \text{(L-Out)}$$

$$(\tilde{x} : \tilde{T}; \sigma; \omega; c?[\tilde{y}].Q) \xrightarrow{c?[\tilde{v}]} (\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{v}; Q\{\tilde{v}/\tilde{y}\}) \quad \text{(L-In)}$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{v}; P) \xrightarrow{c![\tilde{v}]} (\tilde{x} : \tilde{T}; \sigma; \omega; P') \quad (\tilde{x} : \tilde{T}; \sigma; \omega; Q) \xrightarrow{c?[\tilde{v}]} (\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{y}; Q')}{(\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{v}; P|Q) \xrightarrow{\tau} (\tilde{x} : \tilde{T}; \sigma; \omega, \tilde{v}; P'|Q')} \quad \text{(L-Com)}$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i)}{(\tilde{x} : \tilde{T}; \sigma; \omega; P + Q) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i)} \quad \text{(L-Sum)}$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega; P) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i)}{(\tilde{x} : \tilde{T}; \sigma; \omega; P|Q) \xrightarrow{\alpha} \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; P_i|Q)} \quad \text{(L-Par)}$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega; P) \xrightarrow{\alpha} (\tilde{x} : \tilde{T}; \sigma'; \omega; P')}{(\tilde{x} : \tilde{T}; \sigma; \omega; (\mathsf{new}\ c{:}\widehat{\ }[T]).P) \xrightarrow{\alpha} (\tilde{x} : \tilde{T}; \sigma'; \omega; (\mathsf{new}\ c{:}\widehat{\ }[T]).P')} \quad \text{if } \alpha \notin \{c?[.], c![.]\} \quad \text{(L-Res)}$$

$$(\tilde{x} : \tilde{T}; \sigma; \omega; \{v\}.P) \xrightarrow{\tau} (\tilde{x} : \tilde{T}; \sigma; \omega; P) \quad \text{(L-Act)}$$

$$(\tilde{x} : \tilde{T}; [x \mapsto |\phi\rangle]; \omega; (\mathsf{ns}\ s)P) \xrightarrow{\tau} (\tilde{x} : \tilde{T}, s : \mathsf{NS}; [\tilde{x}, s \mapsto |\phi\rangle|0\rangle]; \omega, s; P) \text{ if } s \text{ is fresh} \quad \text{(L-Ns)}$$

$$(\tilde{x}, \tilde{y} : \tilde{\mathsf{Qbit}}, q_c : \mathsf{Qbit}, \tilde{z} : \tilde{\mathsf{NS}}; [\tilde{x}, q_c, \tilde{y}, \tilde{z} \mapsto |\phi\rangle]; \omega; [s_{2c}, s_{2c+1} *\!= \mathsf{PS}(q_c)]\,.\,P) \quad \text{(L-PS)}$$
$$\xrightarrow{\tau} (\tilde{x}, \tilde{y} : \tilde{\mathsf{Qbit}}, \tilde{z} : \tilde{\mathsf{NS}}, s_{2c} : \mathsf{NS}, s_{2c+1} : \mathsf{NS}; [\tilde{x}, \tilde{y}, \tilde{z}, s_{2c}, s_{2c+1} \mapsto |\psi\rangle]; \omega'; P)$$

$$\frac{(\tilde{x} : \tilde{T}; \sigma; \omega; e) \longrightarrow_e \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; e_i)}{(\tilde{x} : \tilde{T}; \sigma; \omega; F[e]) \xrightarrow{\tau} \boxplus_i p_i \bullet (\tilde{x} : \tilde{T}; \sigma_i; \omega; F[e_i])} \quad \text{(L-Expr)}$$

Figure 6.8: Transition Relation Rules.

Example 6.2.
$(s_0 : \mathsf{NS}, s_1 : \mathsf{NS}; [s_0, s_1 \mapsto \alpha_1|10\rangle + \alpha_0|01\rangle]; s_0, s_1; \mathsf{measure}\ s_0\,.\,P) \xrightarrow{\tau}$
$\boxplus_{i,j\in\{0,1\},i\neq j} |\alpha_i|^2 (s_0 : \mathsf{NS}, s_1 : \mathsf{NS}; [s_0, s_1 \mapsto |ij\rangle]; s_0, s_1; P)$.

Example 6.2 shows the effect of measurement(R-Measure-NS) within a process. On the right of the transition, we have a probabilistic configuration in which the $\boxplus$ ranges over the possible outcomes i of the measurement and the $|\alpha_i|^2$ are the weights of the components of the mixture. The measurement outcomes are classical values which are the number of photons detected.

The labelled transitions between configurations are defined given by the set of rules shown in Figure 6.8. The rule L-Prob is a probabilistic transition in which p_i is the probability of the transition. The rules L-In and L-Out represent the input and output actions respectively, which are the *visible* interactions with the environment. $Q\{\tilde{v}/\tilde{y}\}$ in rule L-In indicates that Q with a list of values $\tilde{v}$ substituted for the list of variables $\tilde{y}$. When the two processes of the input and output actions are put in parallel then each

has a partner for its potential interaction, and the input and output can synchronise, resulting in a τ transition which is given by the rule L-Com. The rule L-Act just removes actions. This is a reduction of the action expression to v which would involve effects like measurement or transformation of the quantum state. The rules discussed are similar to the rules in [51] with the modification of introducing the number states into the configuration in order to describe the behaviour of LOQC.

Rule L-PS describes the PS operation, which is the conversion of a polarisation qubit (q_c) to the number states (s_{2c} and s_{2c+1}). Here $\tilde{x}$, $\tilde{y}$ and $\tilde{z}$ means a list of names of the form q_i, q_j and s_k where $k \neq (2i, 2i{+}1, 2j$, and $2j{+}1)$. The quantum state of the system before the operation is given as $|\phi\rangle = |\alpha\rangle|0\rangle|\beta\rangle|\gamma\rangle + |\alpha'\rangle|1\rangle|\beta'\rangle|\gamma'\rangle$. The initial configuration shows that $q_c \in \omega$ and $s_{2c}, s_{2c+1} \notin \omega$ where ω is a list of names that is owned by the process P and after the operation we have a new list ω' (where $q_c \notin \omega'$ and $s_{2c}, s_{2c+1} \in \omega'$) and the quantum state of the system is given as $|\psi\rangle = |\alpha\rangle|\beta\rangle|\gamma\rangle|10\rangle + |\alpha'\rangle|\beta'\rangle|\gamma'\rangle|01\rangle$.

Example 6.3.
$(q_0 : \mathsf{Qbit}, q_2 : \mathsf{Qbit}, s_2 : \mathsf{NS}, s_3 : \mathsf{NS}; [q_0, q_2, s_2, s_3 \mapsto \alpha|00\rangle|10\rangle + \beta|11\rangle|01\rangle];$
$q_0, q_2, s_2, s_3; [s_0, s_1 *= \mathsf{PS}(q_0)] \,.\, P) \xrightarrow{\tau}$
$(q_2 : \mathsf{Qbit}, \tilde{s'} : \tilde{\mathsf{NS}}; [q_2, s_0, s_1, s_2, s_3 \mapsto \alpha|0\rangle|1010\rangle + \beta|1\rangle|0101\rangle]; q_2, s_0, s_1, s_2, s_3; P).$

Example 6.3 shows the effect of PS operation on qubit q_0. The qubit is converted to the number states s_0, s_1 and $\tilde{s'}$ indicates that it is a list of names comprising s_0, s_1, s_2 and s_3 of type NS.

In the next section, we will discuss to model an LOQC CNOT gate using the CQP definitions of the linear optical elements that has been explained earlier.

6.3 CQP model of an LOQC CNOT gate

The structure of the system is shown in Figure 6.9. The system receives two polarisation qubits (control and target) as inputs through the channels a and b. The qubits are then converted to number states by the process $PolSe_{CT}$, and these are provided as the input to the CNOT gate represented by process $CNOT$. The output of $CNOT$ is then *post-selected* by the process PSM. We demonstrate this by removing the unsuccessful outcomes of the gate and recording a coincidence count for every successful outcome. The output of the system are the classical values of the CNOT gate output for which a coincidence count is obtained. The whole system is then defined as a parallel composition of $PolSe_{CT} \mid CNOT \mid PSM$, which means that the processes can proceed simultaneously

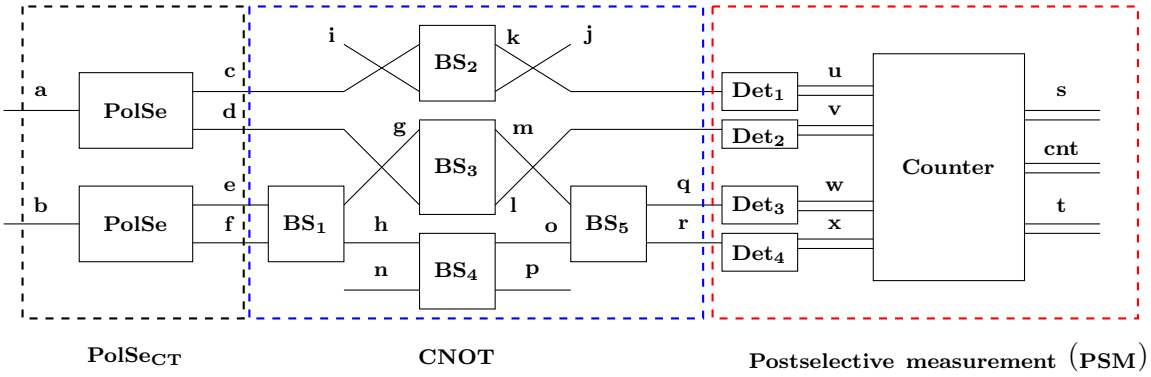


FIGURE 6.9: Model of LOQC CNOT gate: The dashed lines enclose the subsystems which are defined in the text.

and interact with each other, and the CQP definition of the system is

$$System(a,b,s,t,cnt) = (\mathsf{new}\ c,d,e,f,g,h,i,j,k,l,m,n,o,p,u,v,w,x,q,r)$$
$$(PolSe_{CT}(a,b,c,d,e,f) \mid CNOT(c,d,e,f,i,j,n,j,k,l,p,q,r) \mid$$
$$PSM(k,l,q,r,s,t,cnt))$$

where the channels (a,b) are of type ^[Qbit], channels (c,...,r) are of type ^[NS], channels (s,...,x) are of type ^[Val] and the channel *cnt* is of type ^[Bit]. We have omitted the types from our definitions, for brevity. Each process is parameterised by the channels on which it interacts with other processes.

$PolSe_{CT}$ represents the conversion of the control and target qubits from polarisation encoding to spatial encoding or number states given by the definition:

$$PolSe_{CT}(a,b,c,d,e,f) = PolSe(a,c,d) \mid PolSe(b,e,f)$$

Recall from Section 6.2.3 that *PolSe* represents the combination of a PBS and PR. The number states are then provided as inputs to the CNOT gate.

The CNOT gate, represented by the process *CNOT*, is a combination of five beam splitters. Each BS is represented by a process *BS* and is annotated to show the correspondence with Figure 6.9. The process *CNOT* consists of all BSs in parallel. BS_2 and BS_3 have their inputs crossed over, corresponding to their orientation in Figure 6.2. Vacuum states y and z (which means absence of a photon) are created by ($\mathsf{ns}\ y,z$) and communicated to BS_2 and BS_4 respectively through the channels i and n. The CQP definition of *CNOT* is:

$$CNOT(c,d,e,f,i,n,j,k,l,p,q,r) = (\mathsf{new}\ g,h,m,o)(\mathsf{ns}\ y,z)(BS_1(e,f,g,h,\tfrac{1}{2})| \\ i![y]\,.\,\mathbf{0} \mid BS_2(i,c,j,k,\tfrac{1}{3}) \mid j?[y:\mathsf{NS}]\,.\,\mathbf{0} \mid BS_3(d,g,l,m,\tfrac{1}{3}) \mid n![z]\,.\,\mathbf{0}| \\ BS_4(h,n,o,p,\tfrac{1}{3}) \mid p?[z:\mathsf{NS}]\,.\,\mathbf{0} \mid BS_5(m,o,q,r,\tfrac{1}{2}))$$

The parallel composition of processes in $CNOT$ permits interaction between processes. This means that the output on the channels g,h,m and o of the respective processes BS_1, BS_3 and BS_4 synchronises with the input on channels g,h,m and o of processes BS_3, BS_4 and BS_5. The outputs (number states) of $CNOT$ are communicated through the channels k,l,q and r, to the process PSM. The unused BS outputs j and p are absorbed by $j?[y:\mathsf{NS}]$ and $p?[z:\mathsf{NS}]$.

$$PSM(k,l,q,r,s,t,cnt) = (\mathsf{new}\ u,v,w,x)(Det_1(k,u) \mid Det_2(l,v) \mid Det_3(q,w)| \\ Det_4(r,x) \mid Counter(u,v,w,x,s,t,cnt))$$

PSM performs the *post-selective* measurement. This is achieved with the parallel composition of detectors and a process $Counter$. Detectors $Det_1, Det_2, Det_3, Det_4$ are annotated to match Figure 6.9 and measure the number states associated with the control and target qubits. The output of a detector is a classical value which represents the measurement outcome, that is the number of photons detected. The outcomes of the detector processes are given as inputs to the process $Counter$.

$$Counter(u,v,w,x,s,t,cnt) = u?[c_0:\mathsf{Val}]\,.\,v?[c_1:\mathsf{Val}]\,.\,w?[t_0:\mathsf{Val}]\,.\,x?[t_1:\mathsf{Val}]\,. \\ \textbf{if}\ (c_0+c_1=1 \text{ and } t_0+t_1=1)\ \textbf{then}\ s![c_1]\,.\,t![t_1]\,.\,cnt![1]\,.\,\mathbf{0}\ \textbf{else}\ cnt![0]\,.\,\mathbf{0}$$

$Counter$ is a process which represents the coincidence measurement. Coincidence is observed by detecting two photons, one at channels u or v and the other at w or x. It also provides the correct output of the CNOT gate in terms of classical values through the channels s and t. The output is received only for coincidence. This is determined by the **if** ... **else** conditions in the definition. When the condition is satisfied, then a count is registered by outputting a value 1 through the channel cnt. If the condition is not satisfied then a value 0 is given as output, which signifies no coincidence and we don't get any values from the channels s and t. Thus, we achieve post-selection in the coincidence basis in our model.

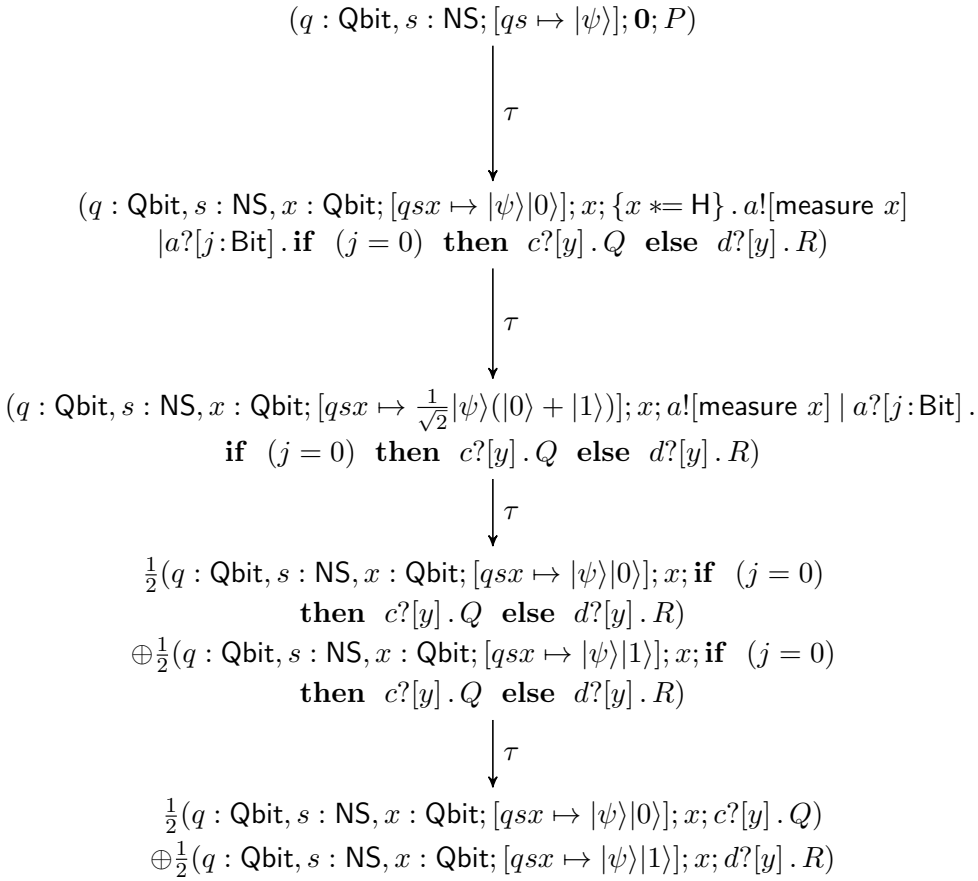


FIGURE 6.10: Example 6.4

6.4 Discussion

In this section, we present an analysis of the labelled transition system of CQP that has been extended to model LOQC. We worked towards in presenting our first model of an LOQC CNOT gate. The present semantics of CQP excludes the idea of mixed configurations, as our intention was to develop CQP in order to describe or model LOQC. But in order for the equivalence of processes to have the important property of*congruence*, the semantics must include mixed configurations as it plays a vital role in the analysis of the measurement, which is explained in the earlier part of the thesis.

Our next task would be to extend the theory of equivalence in CQP to LOQC. This would help us to verify systems but also would provide us a more physical understanding of the property of equivalence. Extending the semantics of CQP to verify LOQC is not a straightforward task. For example in the previous section, the CQP definition of the experimental system that demonstrates a LOQC CNOT gate consists of a process called *Counter*. This records or represents the coincidence measurement of two photons. We perform this task by employing an $\mathbf{if} - \mathbf{then}$ condition. The presence of mixed

configurations appears to make it difficult to introduce the **if** $-$ **then** statement into CQP.

A mixed configuration is a mixture of pure configurations with the *same process term*. In an if-then statement, different processes would follow depending on the value of the condition. Suppose the value of the condition comes from a quantum measurement as in the case of the process *Counter* and there is some probability that the condition is true. Then it requires that the mixed configurations needs to evolve to become a mixture with different process terms, which is not possible with the present definition of the process *Counter*. To illustrate this, we will define a simple process P given by the following example:

Example 6.4.
$P = (\mathsf{qbit}\ x)\,.\,\{x{*}{=}\mathsf{H}\}\,.\,a![\mathsf{measure}\ x]|a?[j:\mathsf{Bit}]\,.\,\mathbf{if}\ \ (j=0)\ \ \mathbf{then}\ \ c?[y]\,.\,Q\ \ \mathbf{else}\ \ d?[y]\,.\,R$

Here, Q and R are different processes which are executed depending on the outcome of a quantum measurement with a probability $\frac{1}{2}$ resulting in 0 and a probability $\frac{1}{2}$ resulting in 1.

Let the initial configuration be $(q:\mathsf{Qbit}, s:\mathsf{NS}; [qs \mapsto |\psi\rangle]; \mathbf{0}; P)$. The execution of the process shown in Figure 6.10. We find that the final mixed configuration consists of different processes which would not be allowed.

With this initial investigation, we will discuss in detail the semantics that are required for the extension of the theory of equivalence of CQP for LOQC.

Chapter 7

Formal verification of LOQC using CQP

Chapter 6 presented an initial attempt at modelling a realistic experimental system associated with quantum computing. In this chapter, we improve our semantics by including the concept of mixed configurations and there by extend the theory of equivalence in CQP in order to analyse and verify LOQC. This provides us for the first time with a more physical understanding of the property of equivalence. We present two models of an experimental system that demonstrates a LOQC CNOT gate and prove that they are equivalent to their specification. In our second model, we describe the process of *post-selection*, which plays an important role in LOQC, where one considers only a subset of all experimental runs that fulfil predefined criteria.

7.1 Modified syntax and semantics of CQP for LOQC

7.1.1 Syntax

The syntax of CQP for LOQC is defined by the grammar as shown in Figure 7.1. This is very similar to the syntax (Figure 6.4) as described in the previous chapter with some changes. We have a new addition to the expression called post-selective measurement psmeasure $e_1, \ldots, e_n$ and the **if** $\ldots$ **then** conditions are introduced into the expression and not in the processes. As discussed in the previous chapter, this is due to the fact that the presence of mixed configuration makes it hard to employ **if** $-$ **then** conditions in processes. The conditions allow different processes to be computed in a mixed configuration which should not happen. The reason being that the mixed

$$
\begin{array}{lcl}
T & ::= & \mathsf{Int} \mid \mathsf{Qbit} \mid \mathsf{NS} \mid \mathsf{Bit} \mid \hat{}[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v & ::= & x \mid 0 \mid 1 \mid \cdots \mid \mathsf{H} \mid \cdots \\
e & ::= & v \mid \mathsf{measure}\ \tilde{e} \mid \mathsf{psmeasure}\ \tilde{e} \mid \tilde{e} *\!= e \mid e + e' \mid (e, e) \mid \text{if } e \text{ then } e \text{ else } e \mid x : \mathsf{NS}, y : \mathsf{NS} *\!= \mathsf{PS}(z) \\
P & ::= & \mathbf{0} \mid (P|P) \mid P + P \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid (\mathsf{qbit}\ x)P \mid (\mathsf{ns}\ x)P \mid (\mathsf{new}\ x : \hat{}[T])P
\end{array}
$$

FIGURE 7.1: Syntax of CQP for LOQC

$$
\begin{array}{lcl}
v & ::= & \ldots \mid q \mid s \mid c \\
E & ::= & [] \mid \mathsf{measure}\ E, \tilde{e} \mid \mathsf{measure}\ v, E, \tilde{e} \mid \ldots \mid \mathsf{measure}\ \tilde{v}, E \mid E + e \mid v + E \mid \text{if } E \text{ then } e \text{ else } e \\
F & ::= & []?[\tilde{x}].P \mid []![\tilde{e}].P \mid v![[].\tilde{e}].P \mid v![v, [], \tilde{e}].P \mid \cdots \mid v![\tilde{v}, []].P \mid \{[]\}.P
\end{array}
$$

FIGURE 7.2: Internal syntax of CQP for LOQC.

configuration is defined as components that differ in values but has the same process structure. The *internal syntax* is provided in Figure 7.2

7.1.2 Linear Optical Elements in CQP

Recall the definitions from Section 6.2.3, we define the process *PolSe* which provides the input to the LOQC CNOT gate.

$$PolSe(a{:}\hat{}[\mathsf{Qbit}], c{:}\hat{}[\mathsf{NS}], d{:}\hat{}[\mathsf{NS}]) = a?[q_0{:}\mathsf{Qbit}]\,.\,\{s_0{:}\mathsf{NS}, s_1{:}\mathsf{NS} *\!= \mathsf{PS}(q_0)\}\,.\,c![s_0]\,.\,d![s_1]\,.\,\mathbf{0}$$

PS corresponds to the transformation produced by the combination of PBS and PR, introduced by Definition 6.1. The CQP definition of the beam splitter *BS* is

$$BS(e{:}\hat{}[\mathsf{NS}], f{:}\hat{}[\mathsf{NS}], h{:}\hat{}[\mathsf{NS}], i{:}\hat{}[\mathsf{NS}], \eta) = e?[s_2{:}\mathsf{NS}]\,.\,f?[s_3{:}\mathsf{NS}]\,.\,\{s_2, s_3 *\!= \mathsf{B}_\eta\}\,.\,h![s_2]\,.\,i![s_3]\,.\,\mathbf{0}$$

Now, we define two types of detectors, *Det* and *PDet*. In the previous chapter we defined the detector to only measure a single number state. Since, a polarisation qubit is represented by a pair of number states, we define the detector *Det* to perform measurement of a pair of number states. This also makes a simpler and easier analysis of the CQP models of LOQC. We define another type of detector represented by process *PDet* that performs post-selective measurement. Both *Det* and *PDet* are used for the measurement of the outputs of CNOT gate.

$$Det(l{:}\hat{}[\mathsf{NS}], m{:}\hat{}[\mathsf{NS}], u{:}\hat{}[\mathsf{Val}, \mathsf{Val}]) = l?[s_0{:}\mathsf{NS}]\,.\,m?[s_1{:}\mathsf{NS}]\,.\,u![\mathsf{measure}\ s_0, s_1]\,.\,\mathbf{0}$$

$$PDet(l{:}\hat{}[\mathsf{NS}], m{:}\hat{}[\mathsf{NS}], u{:}\hat{}[\mathsf{Val}]) = l?[s_0{:}\mathsf{NS}]\,.\,m?[s_1{:}\mathsf{NS}]\,.\,u![\mathsf{psmeasure}\ s_0, s_1]\,.\,\mathbf{0}$$

The expression measure s_0, s_1 probabilistically evaluates to a pair of positive integers which is the number of photons detected in the respective channels and psmeasure s_0, s_1 produces a zero or one which is a result of post-selection.

7.1.3 Semantics of CQP

We will now explain the formal semantics of CQP. The *pure configuration* has the same form as the configurations that are defined in Chapter 6. We have seen in Chapter 4 that a mixed configuration is defined as a weighted sum of pure configurations. Mixed configurations arise from measurements whose results are not made visible to an observer. In a similar case, we also define a mixed configuration where the list of elements that forms a quantum state can comprise of qubits or number states.

Definition 7.1 (Mixed Configuration)**.** A *mixed configuration* is a weighted distribution of pure configurations, written as

$$\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \widetilde{x} = |\psi_i\rangle; \omega; \lambda\widetilde{y} \bullet P; \widetilde{v_i})$$

with weights g_i where $\sum_{i\in I} g_i = 1$ and for each $i \in I, 0 < g_i \leq 1$ and $|\psi_i\rangle \in \mathbb{H} = \mathbb{H}_q \otimes \mathbb{H}_s$ and $|\widetilde{v_i}| = |\widetilde{y}|$.

This is required for the equivalence of processes to have the important property of *congruence*. We now present the different types of *labelled transition rules* of CQP that are extended from the previous work [51] in order to verify LOQC, which is the focus of this Chapter.

Expression Transition Rules. Earlier, we have seen the expression transition rules of CQP for qubits that are given in Figure 4.3. Now, we present the expression transition rules of CQP that are applicable to qubits and number states. The rules are shown in Figure 7.3. Rules R-Measure-NS-2, R-PS-Measure and R-Measure-QBIT are measurement rules which produces a mixed configuration. The first two measurement rules measure a pair of number states and the last rule measures qubit. R-Measure-NS-2 produces a mixed configuration over the possible measurement outcomes k and l. The measurement outcomes are classical values which are the number of photons detected. R-PS-Measure is a *post-selective* measurement rule which produces a mixed configuration over the possible measurement outcome l. Rule R-Trans-NS deals with unitary transformations which result in literal unit. We introduce new rules called R-IfThen-T and R-IfThen-F that is necessary for the **if** ... **else** conditions in the expression configurations.

$$(\tilde{x} : \tilde{T}; \sigma; \omega; u + v) \longrightarrow_v (\tilde{x} : \tilde{T}; \sigma; \omega; w) \text{if } u \text{ and } v \text{ are integer literals and } w = u + v \quad \text{(R-Plus)}$$

$$(\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s} \geq 0} \alpha_{\tilde{s}} |\beta_{\tilde{s}}\rangle |\tilde{s}\rangle]; \omega; \mathsf{measure}\ s_a, s_b) \longrightarrow_v \quad \text{(R-Measure-NS-2)}$$

$$\oplus_{k,l \geq 0} g_{kl} (\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s'} \geq 0} \frac{\alpha_{\tilde{s'}}}{\sqrt{g_{kl}}} |\beta_{\tilde{s'}}\rangle |\widetilde{s'}\rangle]; \omega; \lambda yz \bullet (y, z); k, l)$$

$$\text{where } g_{kl} = \sum_{\tilde{i}} |\alpha_{\tilde{s'}}|^2, \tilde{s} = s_0, \ldots, s_{n-1}, \tilde{s'} = s_0, \ldots, s_{a-1}, k, \ldots, l, s_{b+1}, \ldots, s_{n-1},$$

$$\tilde{i} = s_0, \ldots, s_{n-1} \setminus (s_a, s_b) \text{and } (a, b) \in \{0, \ldots, n-1\} \text{ and } a \neq b$$

$$(\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s} \geq 0} \alpha_{\tilde{s}} |\beta_{\tilde{s}}\rangle |\tilde{s}\rangle]; \omega; \mathsf{psmeasure}\ s_a, s_b) \longrightarrow_v \quad \text{(R-PS-Measure)}$$

$$\oplus_{k,l \in \{0,1\}, k \neq l} h_{kl} (\tilde{x} : \tilde{T}; [\tilde{x} \mapsto \sum_{\tilde{s'} \geq 0} \frac{\alpha_{\tilde{s''}}}{\sqrt{h_{kl}}} |\beta_{\tilde{s'}}\rangle |\widetilde{s'}\rangle]; \omega; \lambda z \bullet z; l)$$

$$\text{where } h_{kl} = \sqrt{g_{op}} \frac{1}{\sum_{\tilde{j}} |\alpha_{\tilde{s''}}|^2} \text{ and } g_{op} = \sum_{\tilde{i}} |\alpha_{\tilde{s'}}|^2,\ o, p \geq 0,\ \tilde{s} = s_0, \ldots, s_{n-1},$$

$$\tilde{s'} = s_0, \ldots, s_{a-1}, o, \ldots, p, s_{b+1}, \ldots, s_{n-1},$$

$$\tilde{i} = s_0, \ldots, s_{n-1} \setminus (s_a, s_b)\ \tilde{s''} = s_0, \ldots, s_{a-1}, k, \ldots, l, s_{b+1}, \ldots, s_{n-1},$$

$$\text{and } \tilde{j} = s_0, \ldots, s_{a-1}, k, \ldots, l, s_{b+1}, \ldots, s_{n-1} \text{ and } (a, b) \in \{0, \ldots, n-1\} \text{ and } a \neq b$$

$$(q_0, \ldots, q_{n-1} = \alpha_0 |\phi_0\rangle + \cdots + \alpha_{2^n-1} |\phi_{2^n-1}\rangle; \omega; \mathsf{measure}\ q_0, \ldots, q_{r-1}) \longrightarrow_v \quad \text{(R-Measure-QBIT)}$$

$$\oplus_{0 \leq m < 2^r}\ g_m\ (q_0, \ldots, q_{n-1} = \frac{\alpha_{l_m}}{\sqrt{g_m}} |\phi_{l_m}\rangle + \cdots + \frac{\alpha_{u_m}}{\sqrt{g_m}} |\phi_{u_m}\rangle; \omega; \lambda x \bullet x; m)$$

$$\text{where } l_m = 2^{n-r} m, u_m = 2^{n-r}(m+1) - 1, g_m = |\alpha_{l_m}|^2 + \cdots + |\alpha_{u_m}|^2$$

$$(\tilde{q} : \mathsf{Qbit}, \tilde{s} : \mathsf{NS}; [\tilde{q}, \tilde{s} \mapsto |\psi\rangle]; \omega; s_0, \ldots, s_{2r-1} *= U) \longrightarrow_v \quad \text{(R-Trans-NS)}$$

$$(\tilde{q} : \mathsf{Qbit}, \tilde{s} : \mathsf{NS}; [\tilde{q}, s_0, \ldots, s_{n-1} \mapsto (I_{|\tilde{q}|} \otimes U \otimes I_{(n-2r)}) |\psi\rangle]; \omega; \mathsf{unit})$$

$$(\tilde{x} : \tilde{T}; \sigma; \omega; \text{if true then } e \text{ else } e') \longrightarrow_v (\tilde{x} : \tilde{T}; \sigma; \omega; e) \quad \text{(R-IfThen-T)}$$

$$(\tilde{x} : \tilde{T}; \sigma; \omega; \text{if false then } e \text{ else } e') \longrightarrow_v (\tilde{x} : \tilde{T}; \sigma; \omega; e') \quad \text{(R-IfThen-F)}$$

$$\frac{\forall i \in I. (\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |\psi_i\rangle]; \omega; e\{\widetilde{u_i}/\widetilde{y}\}) \longrightarrow_v \oplus_{j \in J_i}\ g_{ij}\ (\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |\psi_{ij}\rangle]; \omega; \lambda \widetilde{z} \bullet e'\{\widetilde{u_i}/\widetilde{y}\}; \widetilde{v}_{ij})}{\oplus_{i \in I}\ h_i\ (\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |\psi_i\rangle]; \omega; \lambda \widetilde{y} \bullet E[e]; \widetilde{u_i}) \longrightarrow_e \oplus_{\substack{i \in I \\ j \in J_i}} h_i g_{ij}\ (\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |\psi_{ij}\rangle]; \omega; \lambda \widetilde{y}\widetilde{z} \bullet E[e']; \widetilde{u_i}, \widetilde{v}_{ij})} \quad \text{(R-Context)}$$

Figure 7.3: Transition rules for values and expressions.

Pure Configuration Transition Rules. The transition rules for pure process configurations are given in Figure 7.4. This is a straightforward extension of the rules represented in Figure 4.4, with the inclusion of number states.

$$(\widetilde{p}, \widetilde{q} : \tilde{\mathsf{Qbit}}, \widetilde{r}, \widetilde{s} : \tilde{\mathsf{NS}}, [\widetilde{pqrs} \mapsto |\psi\rangle]; \widetilde{p}, \widetilde{q}, \widetilde{r}, \widetilde{s}; c![\widetilde{v}, \widetilde{q}, \widetilde{s}].P) \xrightarrow{c![\widetilde{v},\widetilde{q},\widetilde{s}]}_p (\widetilde{p}, \widetilde{q} : \tilde{\mathsf{Qbit}}, \widetilde{r}, \widetilde{s} : \tilde{\mathsf{NS}}, [\widetilde{pqrs} \mapsto |\psi\rangle]; \widetilde{p}, \widetilde{r}; P) \quad \text{(P-Out)}$$

$$(\widetilde{q} : \tilde{\mathsf{Qbit}}, \widetilde{s} : \tilde{\mathsf{NS}}, [\widetilde{qs} \mapsto |\psi\rangle]; \omega; c?[\widetilde{y}, \widetilde{x}].P) \xrightarrow{c?[\widetilde{v},\widetilde{p},\widetilde{r}]}_p (\widetilde{q} : \tilde{\mathsf{Qbit}}, \widetilde{s} : \tilde{\mathsf{NS}}, [\widetilde{qs} \mapsto |\psi\rangle]; \omega, \widetilde{p}, \widetilde{r}; P\{\widetilde{v}, \widetilde{r}/\widetilde{y}, \widetilde{p}/\widetilde{x}\}) \quad \text{(P-In)}$$

$$\frac{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega'; P')}{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P \mid Q) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega'; P' \mid Q)} \quad \text{(P-Par)}$$

$$\frac{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega'; P')}{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P + Q) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega'; P')} \quad \text{(P-Sum)}$$

$$\frac{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; P')}{(\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; (\mathsf{new}\ c)P) \xrightarrow{\alpha}_p (\widetilde{x} : \tilde{T}, [\widetilde{x} \mapsto |\psi\rangle]; \omega; (\mathsf{new}\ c)P')} \quad \text{if } \alpha \notin \{c?[\cdot], c![\cdot]\} \quad \text{(P-Res)}$$

Figure 7.4: Transition rules for pure process configurations.

Mixed Configuration Transition Rules. The transition rules on mixed configurations are defined in Figures 7.5 and 7.6. The rules L-In, L-Out-Qbit and L-Out-Ns represent the input and output actions respectively, which are the visible interactions with the environment. L-Com and L-Act perform the same function irrespective of qubits and number states. Rules L-Qbit and L-Ns are for introducing additional Qbit and NS variables respectively. ns declarations represents vacuum states. Since the values associated with the an input action are determined by the environment, this action is identical across all components in a mixed configuration. The rule L-PS describes the PS operation, which is the conversion of a polarisation qubit (q_c) to the number states (s_a and s_b). The quantum state of the system before the operation is given as $|\phi\rangle = |\alpha\rangle|0\rangle|\beta\rangle|\gamma\rangle + |\alpha'\rangle|1\rangle|\beta'\rangle|\gamma'\rangle$. The initial configuration shows that $q_c \in \omega$ and $s_a, s_b \notin \omega$ where ω is a list of names that is owned by the process P and after the operation we have a new list ω' (where $q_c \notin \omega'$ and $s_a, s_b \in \omega'$) and the quantum state of the system is given as $|\psi\rangle = |\alpha\rangle|\beta\rangle|\gamma\rangle|10\rangle + |\alpha'\rangle|\beta'\rangle|\gamma'\rangle|01\rangle$. The rule L-Out-Qbit and L-Out-Ns is the point at which mixed configurations are combined with probabilistic branching. Branching occurs only when there is information to distinguish the components. This information is represented by the classical values that are outputs, which may vary between the components.

Next we illustrate with a few examples of some of the *labelled transition rules* of CQP.

Example 7.1.
$(q, s, t : \widetilde{T}; [q, s, t \mapsto \alpha_{10}|0\rangle|10\rangle + \alpha_{01}|1\rangle|01\rangle + \alpha_{20}|0\rangle|20\rangle]; q, s, t; c![\mathsf{measure}\ s, t]\,.\,P) \xrightarrow{\tau}$
$\oplus_{i\in I, j\in J} \ |\alpha_{ij}|^2 (q, s, t : \widetilde{T}; [q, s, t \mapsto |\beta\rangle|ij\rangle]; q, s, t; \lambda yz \bullet c![y, z]\,.\,P; i, j)$.

$$\boxplus_j p_j(\oplus_i\ g_i\ (\widetilde{x} : \tilde{T}; \sigma_i; \omega; P_i)) \overset{p_i}{\rightsquigarrow} \oplus_i\ g_i\ (\widetilde{x} : \tilde{T}; \sigma_i; \omega; P_i) \quad \text{(L-Prob)}$$

$$\oplus_i\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet c?[\widetilde{q}, \widetilde{s}].P; \widetilde{v_i}) \xrightarrow{c?[\widetilde{p},\widetilde{r}]} \quad \text{(L-In)}$$

$$\oplus_i\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega, \widetilde{r}, \widetilde{p}; \lambda\widetilde{z} \bullet P\{\widetilde{p}/\widetilde{q}, \widetilde{r}/\widetilde{s}\}; \widetilde{v_i})$$

$$\forall i \in I.((\widetilde{p}, \widetilde{q}) : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{pqs} \mapsto |\alpha_i\rangle|\beta\rangle]; \widetilde{p}, \widetilde{s}; P\{\widetilde{v_i}/\widetilde{x}\}) \xrightarrow[p]{c![\widetilde{u}_i,\widetilde{r}]}$$

$$((\widetilde{p}, \widetilde{q}) : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{pqs} \mapsto |\alpha_i\rangle|\beta\rangle]; \widetilde{p}', \widetilde{s}; P'\{\widetilde{v_i}/\widetilde{x}\})$$

(L-Out-Qbit)

$$\oplus_{i\in I}\ g_i\ ((\widetilde{p}, \widetilde{q}) : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{pqs} \mapsto |\alpha_i\rangle|\beta\rangle]; \widetilde{p}, \widetilde{s}; \lambda\widetilde{x} \bullet P; \widetilde{v_i}) \xrightarrow{c![U,\widetilde{r}]}$$

$$\boxplus_{j\in J} p_j(\oplus_{i\in I_j} \frac{g_i}{p_j}\ ((\widetilde{p'}, \widetilde{q}) : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{p'rqs} \mapsto \Pi|\alpha_i\rangle|\beta\rangle]; \widetilde{p'}, \widetilde{s}; \lambda\widetilde{x} \bullet P'; \widetilde{v_i}))$$

where $U = \{\widetilde{u}_i \mid i \in I\} = \{\widetilde{w}_j \mid j \in J\}$ and $\forall j \in J, I_j = \{i | \widetilde{u}_i = \widetilde{w}_j\}, p_j = \sum_{i\in I_j} g_i$

and $\widetilde{r} \subseteq \widetilde{p}, \widetilde{p'} = \widetilde{p} \setminus \widetilde{r}, \Pi$ corresponds to the permutation $\pi : \widetilde{pqs} \mapsto \widetilde{p'rqs}$.

$$\forall i, j \in I.(\widetilde{p} : \widetilde{\mathsf{Qbit}}, (\widetilde{t}, \widetilde{s}) : \widetilde{\mathsf{NS}}; [\widetilde{pts} \mapsto |\alpha\rangle|\beta_{ij}\rangle]; \widetilde{p}, \widetilde{s}; P\{\widetilde{v}_{ij}/\widetilde{x}\}) \xrightarrow[p]{c![\widetilde{u}_{ij},\widetilde{r}]}$$

$$(\widetilde{p} : \widetilde{\mathsf{Qbit}}, (\widetilde{t}, \widetilde{s}) : \widetilde{\mathsf{NS}}; [\widetilde{pts} \mapsto |\alpha\rangle|\beta_{ij}\rangle]; \widetilde{p}, \widetilde{s}'; P'\{\widetilde{v}_{ij}/\widetilde{x}\})$$

(L-Out-Ns)

$$\oplus_{i,j\in I}\ g_{ij}\ (\widetilde{p} : \widetilde{\mathsf{Qbit}}, (\widetilde{t}, \widetilde{s}) : \widetilde{\mathsf{NS}}; [\widetilde{pts} \mapsto |\alpha\rangle|\beta_{ij}\rangle]; \widetilde{p}, \widetilde{s}; \lambda\widetilde{x} \bullet P; \widetilde{v}_{ij}) \xrightarrow{c![U,\widetilde{r}]}$$

$$\boxplus_{k\in J} p_k(\oplus_{i,j\in I_k} \frac{g_{ij}}{p_k}\ (\widetilde{p} : \widetilde{\mathsf{Qbit}}, (\widetilde{t}, \widetilde{s}' : \widetilde{\mathsf{NS}}); [\widetilde{pt}\widetilde{s}'\widetilde{r} \mapsto \Pi|\alpha\rangle|\beta_{ij}\rangle]; \widetilde{p}, \widetilde{s}'; \lambda\widetilde{x} \bullet P'; \widetilde{v}_{ij}))$$

where $U = \{\widetilde{u}_{ij} \mid i, j \in I\} = \{\widetilde{e}_k \mid k \in J\}$, and $\forall k \in J, I_k = \{i, j | \widetilde{u}_{ij} = \widetilde{e}_k\}, p_k = \sum_{i,j\in I_k} g_{ij}$

and $\widetilde{r} \subseteq \widetilde{s}, \widetilde{s}' = \widetilde{s} \setminus \widetilde{r}, \Pi$ corresponds to the permutation $\pi : \widetilde{pts} \mapsto \widetilde{ptrs}'$.

$$\frac{\forall i \in I.(\widetilde{x} : \widetilde{T}; \sigma_i; \omega, \widetilde{r}; P\{\widetilde{v_i}/\widetilde{z}\}) \xrightarrow[p]{c![\widetilde{u}_i,\widetilde{r}]} (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; P'\{\widetilde{v_i}/\widetilde{z}\}) \quad \forall i \in I.(\widetilde{x} : \widetilde{T}; \sigma_i; \omega; Q\{\widetilde{v_i}/\widetilde{z}\}) \xrightarrow[p]{c?[\widetilde{u}_i,\widetilde{r}]} (\widetilde{x} : \widetilde{T}; \sigma_i; \omega, \widetilde{r}; Q'\{\widetilde{v_i}/\widetilde{z}\})}{\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega, \widetilde{r}; \lambda\widetilde{z} \bullet P \mid Q; \widetilde{v_i}) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega, \widetilde{r}; \lambda\widetilde{z} \bullet P' \mid Q'; \widetilde{v_i})} \quad \text{(L-Com)}$$

$$\frac{\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet P; \widetilde{v_i}) \xrightarrow{\alpha} \oplus_{\substack{i\in I \\ j\in J_i}}\ g_i h_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega'; \lambda\widetilde{z}\widetilde{y} \bullet P'; \widetilde{v_i}, \widetilde{w_{ij}})}{\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet P \mid Q; \widetilde{v_i}) \xrightarrow{\alpha} \oplus_{\substack{i\in I \\ j\in J_i}}\ g_i h_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega'; \lambda\widetilde{z}\widetilde{y} \bullet P' \mid Q; \widetilde{v_i}, \widetilde{w_{ij}})} \quad \text{(L-Par)}$$

Figure 7.5: Transition rules for mixed process configurations

$$\oplus_{i\in I}\ g_i\ (\widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{qs} \mapsto |\beta_i\rangle|\gamma_i\rangle]; \omega; \lambda\widetilde{z} \bullet (\mathsf{qbit}\ : y)P; \widetilde{v_i}) \xrightarrow{\tau}$$
$$\oplus_{i\in I}\ g_i\ (\widetilde{q} : \widetilde{\mathsf{Qbit}}, q : \mathsf{Qbit}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{q}, q, \widetilde{s} \mapsto |\beta_i\rangle|0\rangle|\gamma_i\rangle]; \omega, q; \lambda\widetilde{z} \bullet P\{q/y\}; \widetilde{v_i}) \quad \text{where } q \text{ is fresh}$$
(L-QBIT)

$$\oplus_{i\in I}\ g_i\ (\widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{qs} \mapsto |\beta_i\rangle|\gamma_i\rangle]; \omega; \lambda\widetilde{z} \bullet (\mathsf{ns}\ : y)P; \widetilde{X}) \xrightarrow{\tau}$$
$$\oplus_{i\in I}\ g_i\ (\widetilde{q} : \widetilde{\mathsf{Qbit}}, r : \mathsf{NS}, \widetilde{s} : \widetilde{\mathsf{NS}}; [\widetilde{q}, r, \widetilde{s} \mapsto |\beta_i\rangle|0\rangle|\gamma_i\rangle]; \omega, r; \lambda\widetilde{m} \bullet P\{r/y\}; \widetilde{v_i}) \quad \text{where } r \text{ is fresh}$$
(L-NS)

$$\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet \{u\}.P_i; \widetilde{v_i}) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet P; \widetilde{v_i}) \qquad \text{(L-ACT)}$$

$$\oplus_{i\in I}\ g_i\ (\tilde{p}, \tilde{q} : \tilde{\mathsf{Qbit}}, q_c : \mathsf{Qbit}, \tilde{r} : \tilde{\mathsf{NS}}; [\tilde{p}, q_c, \tilde{q}, \tilde{r} \mapsto |\phi\rangle]; \omega; \lambda\widetilde{z} \bullet \{s_a, s_b *= \mathsf{PS}(q_c)\};\ .P, \widetilde{v_i})$$
(L-PS)

$$\xrightarrow{\tau} \oplus_{i\in I}\ g_i\ (\tilde{p}, \tilde{q} : \tilde{\mathsf{Qbit}}, \tilde{r} : \tilde{\mathsf{NS}}, s_a : \mathsf{NS}, s_b : \mathsf{NS}; [\tilde{p}, \tilde{q}, \tilde{r}, s_a, s_b \mapsto |\psi\rangle]; \omega'; \lambda\widetilde{z} \bullet P; ,\widetilde{v_i})$$

$$\frac{\oplus_{i\in I}\ h_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{y} \bullet e; \widetilde{v_i}) \longrightarrow_e \oplus_{\substack{i\in I \\ j\in J_i}}\ h_i g_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; \lambda\widetilde{yz} \bullet e'; \widetilde{v_i}, \widetilde{w_{ij}})}{\oplus_{i\in I}\ h_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{y} \bullet F[e]; \widetilde{v_i}) \xrightarrow{\tau} \oplus_{\substack{i\in I \\ j\in J_i}}\ h_i g_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; \lambda\widetilde{yz} \bullet F[e']; \widetilde{v_i}, \widetilde{w_{ij}})}$$
(L-EXPR)

$$\frac{\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet P; \widetilde{v_i}) \xrightarrow{\alpha} \oplus_{\substack{i\in I \\ j\in J_i}}\ g_i h_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega'; \lambda\widetilde{z}\widetilde{y} \bullet P'; \widetilde{v_i}, \widetilde{w_{ij}})}{\oplus_{i\in I}\ g_i\ (\widetilde{x} : \widetilde{T}; \sigma_i; \omega; \lambda\widetilde{z} \bullet P + Q; \widetilde{v_i}) \xrightarrow{\alpha} \oplus_{\substack{i\in I \\ j\in J_i}}\ g_i h_{ij}\ (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega'; \lambda\widetilde{z}\widetilde{y} \bullet P'; \widetilde{v_i}, \widetilde{w_{ij}})}$$
(L-SUM)

FIGURE 7.6: Transition rules for mixed process configurations

This transition represents the effect of a measurement of a pair of number states (s, t), within a process which is going to output the result of the measurement. The configuration on the left is a *pure configuration* and on the right we have a *mixed configuration* in which the $\oplus$ ranges over the possible outcomes of the measurement and the $|\alpha_{ij}|^2$ are the weights of the components in the mixture. Here, $I = \{0, 1, 2\}$ and $J = \{0, 1\}$. The quantum state $[q, s, t \mapsto |\beta\rangle|ij\rangle]$ corresponds to the measurement outcome. The expression $\lambda yz \bullet c![y, z].P$ represents the fact that the components of the mixed configuration have the same process structure and differ only in the values corresponding to measurement outcomes. The final terms in the configuration, i and j, shows how the abstracted variables y and z should be instantiated in each component. Thus the λyz represents a term into which expressions may be substituted, which is the reason for the λ notation. So the mixed configuration is essentially an abbreviation of

$$|\alpha_{10}|^2(q, s, t : \widetilde{T}; [q, s, t \mapsto |0\rangle|10\rangle]; q, s, t; c![1, 0]..P\{1/y, 0/z\})$$
$$\oplus|\alpha_{01}|^2(q, s, t : \widetilde{T}; [q, s, t \mapsto |1\rangle|01\rangle]; q, s, t; c![0, 1].P\{0/y, 1/z\})$$
$$\oplus|\alpha_{20}|^2(q, s, t : \widetilde{T}; [q, s, t \mapsto |0\rangle|20\rangle]; q, s, t; c![2, 0].P\{2/y, 0/z\}).$$

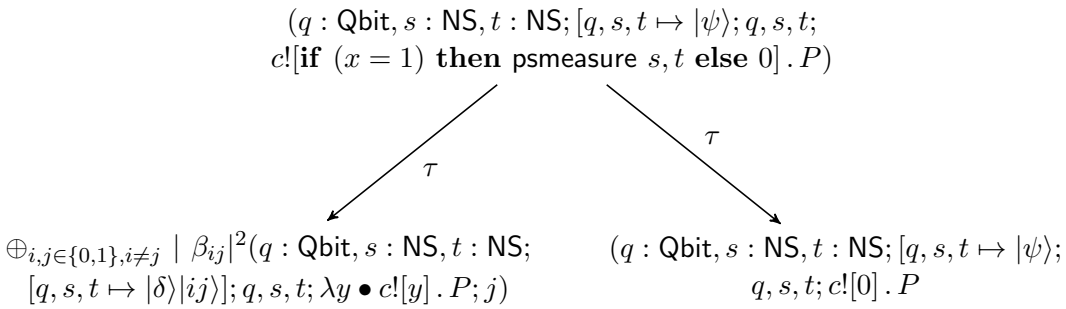
The next transition (R-PS-MEASURE) represents *post-selective* measurement which filters out the measurement values that satisfies a predefined criteria.

Example 7.2.

$(q : \mathsf{Qbit}, s : \mathsf{NS}, t : \mathsf{NS}; [q, s, t \mapsto \alpha_{10}|0\rangle|10\rangle + \alpha_{01}|1\rangle|01\rangle + \alpha_{20}|0\rangle|20\rangle]; q, s, t;$
$c![\mathsf{psmeasure}\ s, t]\,.\,P) \xrightarrow{\tau}$
$\oplus_{i,j\in\{0,1\},i\neq j}\ |\beta_{ij}|^2(q : \mathsf{Qbit}, s : \mathsf{NS}, t : \mathsf{NS}; [q, s, t \mapsto |\delta\rangle|ij\rangle]; q, s, t; \lambda y \bullet c![y]\,.\,P; j).$

This transition represents the effect of a *post-selective* measurement of a pair of number states (s, t), within a process which is going to output the result of the measurement. Here, we have a *mixed configuration* in which the $\oplus$ ranges over the possible outcomes of the measurement with $|\beta_{ij}|^2$ (where $|\beta_{ij}|^2 = \frac{|\alpha_{ij}|^2}{\sum_{ij\in\{0,1\}}|\alpha_{ij}|^2}$) representing the weights of the components in the mixture. Here i and j can have values either 0 or 1 and $i \neq j$, which filters out the measurement values. This is the criterion for *post-selection*. The quantum state $[q, s, t \mapsto |\delta\rangle|ij\rangle]$ corresponds to the measurement outcome. The post-selective measurement produces one value at the output whereas the normal measurement as seen in the above Example 7.1 produces two values at the output. If a measurement outcome is output then it becomes apparent to an observer which of the possible states the system is in, which is represented by probabilistic branching. Then the system is considered to be in one branch or the other and is no longer a mixture.

Example 7.3.



Example 7.3 is a demonstration of the transition rule . Here, $|\psi\rangle$ is $\alpha_{10}|0\rangle|10\rangle + \alpha_{01}|1\rangle|01\rangle + \alpha_{20}|0\rangle|20\rangle$. With the use of the **if** ... **else** conditions in the expression and not in the process, we would not have the conflict which was discussed in the previous chapter. In this example, the execution can proceed in two ways depending on the condition. If the condition is true, i.e $x = 1$, then we get a mixed configuration due to the measurement. If false then we would not get a mixed configuration as there is no measurement and we get a pure configuration. Example 7.4 shows the effect of the output from the mixed configuration of Example 7.3.

Example 7.4. $\oplus_{i,j\in\{0,1\},i\neq j} \mid \beta_{ij}|^2(q : \mathsf{Qbit}, s : \mathsf{NS}, t : \mathsf{NS}; [q,s,t \mapsto |\delta\rangle|ij\rangle]; q,s,t; \lambda y \bullet c![y] \,.\, P; j) \xrightarrow{c![j]} \boxplus_{ij\in\{0,1\},i\neq j}|\beta_{ij}|^2(q : \mathsf{Qbit}, s : \mathsf{NS}, t : \mathsf{NS}; [q,s,t \mapsto |\delta\rangle|ij\rangle]; q,s,t; \lambda y \bullet P; j) \overset{|\beta_{01}|^2}{\rightsquigarrow} (q : \mathsf{Qbit}, s : \mathsf{NS}, t : \mathsf{NS}; [q,s,t \mapsto |1\rangle|01\rangle]; q,s,t; \lambda y \bullet P; 1)$

The output transition produces the intermediate configuration, which is a probability distribution over pure configurations (in contrast to a mixed configuration; note the change from $\oplus$ to $\boxplus$). Because it comes from a mixed configuration, the output transition contains a *set* of possible values. From this intermediate configuration there are two possible probabilistic transitions, of which one is shown ($\overset{|\beta_{01}|^2}{\rightsquigarrow}$).

Example 7.5. $\oplus_{i,j\geq 0}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |q_0\rangle|ij\rangle]; \widetilde{x}; \lambda yz \bullet (c![y] \,.\, P \mid c?[y] \,.\, Q); i,j) \xrightarrow{\tau} \oplus_{i,j\geq 0}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; [\widetilde{x} \mapsto |q_0\rangle|ij\rangle]; \widetilde{x}; \lambda yz \bullet (P \mid Q); i,j)$

Measurement outcomes may be communicated between processes without creating a probability distribution. In these cases an observer must still consider the system to be in a mixed configuration as the outcomes are communicated internally and not to the environment. In Example 7.5 there is a mixed configuration on the left, with arbitrary weights g_{ij}, which we imagine to have been produced by a measurement. However, there is now a receiver for the output. Although there is no difference in process Q between the two components of the mixed configuration, we include it in the λ because the communication will propagate the different possible values for y to Q.

Example 7.6.
$(q : \mathsf{Qbit}, r : \mathsf{Qbit}, p : \mathsf{NS}, t : \mathsf{NS}; [q,r,p,t \mapsto \alpha|00\rangle|10\rangle + \beta|11\rangle|01\rangle]; q,r,p,t; \{u : \mathsf{NS}, v : \mathsf{NS}{*}{=}\mathsf{PS}(q)\} \,.\, P) \xrightarrow{\tau} (r : \mathsf{Qbit}, \tilde{s}' : \tilde{\mathsf{NS}}; [r,p,t,u,v \mapsto \alpha|0\rangle|1010\rangle + \beta|1\rangle|0101\rangle]; r,p,t,u,v; P)$.

Example 7.6 represents the transition which is the conversion of a polarisation qubit (q) to the number states (u and v). $\tilde{s}'$ indicates that it is a list of names comprising p, t, u and v of type NS.

7.2 Behavioural Equivalence of CQP for LOQC

In the previous section, we have explained the new operational semantics of CQP to describe LOQC. We have discussed the two measurement semantics and described an experimental model that demonstrates a LOQC CNOT gate. In this section, we begin our task to extend the theory of equivalence in CQP to apply it for LOQC. The process calculus approach to verification is to define a process *Model* which models the system of interest, another process *Specification* which expresses the specification that *Model* should satisfy, and then prove that *Model* and *Specification* are equivalent. Usually

Specification is defined in a sufficiently simple way that it can be taken as self-evident. We will now define probabilistic branching bisimilarity in full. The definitions in the remainder of this section are an extension from Davidson's thesis [51].

Notation: Let $\xrightarrow{\tau}{}^{+}$ denote zero or one τ transitions; let $\Longrightarrow$ denote zero or more τ transitions; and let $\overset{\alpha}{\Longrightarrow}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. We write $\widetilde{q}$ for a list of qubit names, and similarly for other lists.

Definition 7.2 (Density Matrix of Configurations)**.** Let $\sigma_{ij} = [\widetilde{x} \mapsto |\psi_{ij}\rangle]$ and $\widetilde{y} \subseteq \widetilde{x}$ and $t_{ij} = (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; \lambda\widetilde{w}\widetilde{z} \bullet P; \widetilde{v_{ij}}, \widetilde{o_{ij}})$ and $t = \oplus_{ij}\ g_{ij}\ t_{ij}$. Then

1. $\rho(\sigma_{ij}) = |\psi_{ij}\rangle\langle\psi_{ij}|$
2. $\rho^{\widetilde{y}}(\sigma_{ij}) = \mathrm{tr}_{\widetilde{x}\setminus\widetilde{y}}(|\psi_{ij}\rangle\langle\psi_{ij}|)$
3. $\rho(t_{ij}) = \rho(\sigma_{ij})$
4. $\rho^{\widetilde{y}}(t_{ij}) = \rho^{\widetilde{y}}(\sigma_{ij})$
5. $\rho(t) = \sum_{ij} g_{ij}\rho(t_{ij})$
6. $\rho^{\widetilde{y}}(t) = \sum_{ij} g_{ij}\rho^{\widetilde{y}}(t_{ij})$

Definition 7.3 (Probabilistic Branching Bisimulation)**.** An equivalence relation $\mathcal{R}$ on configurations is a *probabilistic branching bisimulation* on configurations if whenever $(t, u) \in \mathcal{R}$ the following conditions are satisfied.

I. If $t \in \mathcal{T}_n$ and $t \xrightarrow{\tau} t'$ then $\exists u', u''$ such that $u \Longrightarrow u' \xrightarrow{\tau}{}^{+} u''$ with $(t, u') \in \mathcal{R}$ and $(t', u'') \in \mathcal{R}$.

II. If $t \xrightarrow{c![V,\widetilde{X}_1]} t'$ where $t' = \boxplus_{j\in\{1\ldots m\}} p_j t'_j$ and $V = \{\widetilde{v}_1, \ldots, \widetilde{v}_m\}$ and $\widetilde{X}_1$ is either $\widetilde{q}_1$ or $\widetilde{s}_1$ then $\exists u', u''$ such that $u \Longrightarrow u' \xrightarrow{c![V,\widetilde{X}_2]} u''$ with

   a) $(t, u') \in \mathcal{R}$,

   b) $u'' = \boxplus_{j\in\{1\ldots m\}} p_j u''_j$,

   c) for each $j \in \{1, \ldots, m\}$, $\rho_E(t'_j) = \rho_E(u''_j)$.

   d) for each $j \in \{1, \ldots, m\}$, $(t'_j, u''_j) \in \mathcal{R}$.

III. If $t \xrightarrow{c?[\widetilde{v}]} t'$ then $\exists u', u''$ such that $u \Longrightarrow u' \xrightarrow{c?[\widetilde{v}]} u''$ with $(t, u') \in \mathcal{R}$ and $(t', u'') \in \mathcal{R}$.

IV. If $s \in \mathcal{T}_p$ then $\mu(t, D) = \mu(u, D)$ for all classes $D \in \mathcal{T}/\mathcal{R}$.

Definition 7.4 (Probabilistic Branching Bisimilarity)**.** Configurations t and u are *probabilistic branching bisimilar*, denoted $t \leftrightarrows u$, if there exists a probabilistic branching bisimulation $\mathcal{R}$ such that $(t, u) \in \mathcal{R}$.

Definition 7.5 (Probabilistic Branching Bisimilarity of Processes)**.** Processes P and Q are *probabilistic branching bisimilar*, denoted $P \leftrightarrows Q$, if and only if for all σ, $(\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \leftrightarrows (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; Q)$.

Lemma 7.6. *If $\mathcal{R}$ is a probabilistic branching bisimulation and $t\mathcal{R}u$, and $t \Longrightarrow t'$ then there exists u' such that $u \Longrightarrow u'$ and $(t', u') \in \mathcal{R}$.*

Proof. Follows directly from [51]. This is a straightforward induction. □

Lemma 7.7. *Probabilistic branching bisimilarity is an equivalence relation.*

Proof. [51]. This is a straightforward extension where we replace $(\sigma; \emptyset; P)$ by $(\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P)$ to show that probabilistic branching bismilarity is reflexive, symmetric and transitive. □

7.2.1 Preservation Properties

To consider the preservation properties of bisimilarity on processes, we begin by formally defining *contexts* and *congruence*, and their *non-input, non-qubit, non-number state* variants. The reason for considering variants without input, qubit and number state declaration prefixes, is that substitution must also be considered when these are included. We later define full probabilistic branching bisimilarity where we will also consider invariance under substitution.

Definition 7.8 (Context)**.** A *context* C is a process with a non-degenerate occurrence of $\mathbf{0}$ replaced by a hole, $[\cdot]$. Formally,

$$C ::= [] \mid (C \parallel P) \mid \alpha.C + P \mid \alpha.C \mid (\mathsf{new}\ x\widehat{\ }[T])C$$

for $\alpha \in \{e?[\tilde{x} : \tilde{T}], e![\tilde{e}], \{e\}, (\mathsf{qbit}\ x), (\mathsf{ns}\ r)\}$.

Definition 7.9 (Congruence)**.** An equivalence relation $\mathcal{R}$ on processes is a *congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P, Q) \in \mathcal{R}$ and C is a context.

Definition 7.10 (Non-input, non-qubit or non-number state context)**.** A *non-input, non-qubit or non-number state context* is a context in which the hole does not appear under an input or qubit and number state declaration.

Definition 7.11 (Non-input, non-qubit or non-number state congruence)**.** An equivalence relation $\mathcal{R}$ on processes is a *non-input, non-qubit or non-number state congruence* if $(C[P], C[Q]) \in \mathcal{R}$ whenever $(P, Q) \in \mathcal{R}$ and C is a non-input, non-qubit or non-number state context.

Next lemma provides a general form for representing mixed configurations related by internal transitions. The main purpose is to simplify the notations in the following proofs.

Lemma 7.12 (General form of internal transitions)**.** *If*
$t = \oplus_{\substack{ab \in I_{kl} \\ kl \in J}} g_{abkl}\ (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \widetilde{q}, \widetilde{s}; \lambda \widetilde{y} \widetilde{z} \bullet P; \widetilde{w}_{abkl})$ *and* $t \Longrightarrow t'$ *then there exist sets* I'_{kl} *such that* $t' = \oplus_{\substack{ab \in I'_{kl} \\ kl \in J}} g'_{abkl}\ (\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \widetilde{q}', \widetilde{s}'; \lambda \widetilde{y}' \widetilde{z}' \bullet P'; \widetilde{w}'_{abkl})$.

Proof. Adapted from [51], we replace $s = \oplus_{\substack{i\in I_j \\ j\in J}} g_{ij}\ (\sigma_{ij}; \widetilde{q}; \lambda\widetilde{x} \bullet P; \widetilde{v}_{ij})$ by t and by induction on the length of the sequence of τ-transitions. The inductive step is proved by a straightforward induction on the derivation of this transition. □

The following 3 lemmas prove that the state of qubits and number states that are not owned by a particular process is unaffected by any transitions of that process.

Lemma 7.13 (External state independence for $\longrightarrow_v$)**.** *If* $\Gamma;\ \widetilde{s} \vdash e : T$ *and* $t \longrightarrow_v t'$ *where* $t = (\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{s}\widetilde{q}\widetilde{r} \mapsto |\psi\rangle]; \widetilde{q}, \widetilde{s}; e)$ *then* $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(t')$

Proof. By case analysis.
R-Plus: The quantum state and distribution are unchanged.
R-Trans-NS: We have

$$t' = (\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; |\psi'\rangle; \widetilde{q}, \widetilde{s}; \mathsf{unit})$$

where $|\psi\rangle = \sum |\beta\rangle_{\widetilde{s}}|\alpha_\beta\rangle_{\widetilde{qr}}$ and $|\psi'\rangle = (U^m \otimes I)|\psi\rangle$ then $|\psi'\rangle = \sum |\beta'\rangle_{\widetilde{s}}|\alpha_{\beta'}\rangle_{\widetilde{qr}} = \sum U^m|\beta\rangle_{\widetilde{s}} I|\alpha_\beta\rangle_{\widetilde{qr}}$. Now, We have

$$\rho^{\widetilde{qr}}(t') = \sum \langle\beta'|\beta'\rangle_{\widetilde{s}}|\alpha_{\beta'}\rangle\langle\alpha_{\beta'}|_{\widetilde{qr}} = \sum \langle\beta|(U^m)^* U^m|\beta\rangle_{\widetilde{s}} I|\alpha_\beta\rangle\langle\alpha_\beta|_{\widetilde{qr}} I^*$$

$$\rho^{\widetilde{qr}}(t') = \sum \langle\beta|\beta\rangle_{\widetilde{s}}|\alpha_\beta\rangle\langle\alpha_\beta|_{\widetilde{qr}} = \rho^{\widetilde{qr}}(t)$$

R-Measure-NS-2: We have the transition

$$(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{s}\widetilde{q}\widetilde{r} \mapsto \sum \gamma_\beta |\beta\rangle_{\widetilde{s}}|\alpha_\beta\rangle_{\widetilde{qr}}]; \widetilde{q}, \widetilde{s}; \mathsf{measure}\ s_a s_b) \longrightarrow_v$$

$$\oplus_{k,l\geq 0}\ g_{\ kl}(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{s}\widetilde{q}\widetilde{r} \mapsto \sum \frac{\gamma_{\beta'}}{\sqrt{g_{kl}}} |\beta'\rangle_{\widetilde{s}}|\alpha_{\beta'}\rangle_{\widetilde{qr}}]; \widetilde{q}, \widetilde{s}; \lambda yz \bullet yz; k, l)$$

where $\beta = s_0, \ldots, s_{n-1}$ and $\beta' = s_0, \ldots, s_{a-1}, k, l, s_{b+1}, \ldots, s_{n-1}$.
Let $\{|\beta_{kl}\rangle\}$ and $\{|\beta'_i\rangle\}$ be an orthonormal basis for number states $\{s_a, s_b\}$ and $\{s_0, \ldots, s_{n-1}\}/\{s_a, s_b\}$ respectively. Then

$$|\psi\rangle = \sum_{(k,l)\geq 0, i\geq 0} \frac{\gamma_{ikl}}{\sqrt{g_{kl}}} |\beta_{kl}\rangle|\beta'_i\rangle|\alpha_{ikl}\rangle.$$

Now,

$$\begin{aligned} tr_{s_a,s_b}(|\psi\rangle) &= \textstyle\sum_{(k,l)\geq 0, i\geq 0} \sum_{(m,n)\geq 0, j\geq 0} \frac{\gamma_{ikl}}{\sqrt{g_{kl}}} \frac{\gamma^*_{jmn}}{\sqrt{g_{mn}}} \langle\beta_{kl}|\beta_{mn}\rangle|\beta'_j\rangle\langle\beta'_i||\alpha_{jmn}\rangle\langle\alpha_{ikl}| \\ &= \textstyle\sum_{(k,l)\geq 0, i\geq 0} \sum_{j\geq 0} \frac{\gamma_{ikl}}{\sqrt{g_{kl}}} \frac{\gamma^*_{jmn}}{\sqrt{g_{mn}}} |\beta'_j\rangle\langle\beta'_i||\alpha_{jmn}\rangle\langle\alpha_{ikl}| \end{aligned} \tag{7.1}$$

Since, $\langle\beta_{kl}|\beta_{mn}\rangle = 1$ if $kl = mn$ and 0 otherwise. Then,

$$tr_{s_a,s_b}(|\psi\rangle) = \frac{1}{g_{kl}} \sum_{i\geq 0, j\geq 0} \gamma_{ikl}\gamma *_{jmn} |\beta'_j\rangle\langle\beta'_i||\alpha_{jmn}\rangle\langle\alpha_{ikl}| \tag{7.2}$$

Let $t' = \oplus_{k,l\geq 0}\ g_{\ kl} t'_{kl}$, then we have

$$\rho^{\widetilde{qr}}(t') = \sum_{k,l\geq 0} g_{kl}\rho^{\widetilde{qr}}(t'_{kl}) = \sum_{k,l\geq 0} g_{kl} tr_{s_0\ldots s_{n-1}/\{s_a,s_b\}}(tr_{s_a,s_b}\{|\psi\rangle\}) \tag{7.3}$$

By substituting Eq. 7.2 in Eq. 7.3, we get

$$\begin{aligned}\rho^{\widetilde{qr}}(t') &= tr_{s_0\ldots s_{n-1}/\{s_a,s_b\}}(\textstyle\sum_{(k,l)\geq 0}\sum_{i\geq 0,j\geq 0}\gamma_{ikl}\gamma *_{jmn} |\beta'_j\rangle\langle\beta'_i||\alpha_{jmn}\rangle\langle\alpha_{ikl}| \\ &= tr_{s_0\ldots s_{n-1}/\{s_a,s_b\}}(tr_{s_a,s_b}(|\psi\rangle)) = \rho^{\widetilde{qr}}(t)\end{aligned} \tag{7.4}$$

R-PS-Measure: This is similar to the previous case and hence proved. □

Lemma 7.14 (External state independence for $\longrightarrow_e$)**.** *If* $\Gamma;\ \widetilde{s} \vdash e : T$ *and* $t \longrightarrow_v t'$ *where* $t = \oplus_{kl\in I}\ g_{\ kl}(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{sqr} \mapsto |\psi_{kl}\rangle]; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet e; \widetilde{w}_{kl})$ *then* $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(t')$

Proof. The transition $t \longrightarrow_e t'$ is derived by R-Context with a hypothesis where $(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{sqr} \mapsto |\psi_{kl}\rangle]; \widetilde{q}, \widetilde{s}; e\{\widetilde{w}_{kl}/\widetilde{y}\})$. For each $k, l \in I$ we have $\rho^{\widetilde{qr}}(t_{kl}) = \rho^{\widetilde{qr}}(t'_{kl})$ by Lemma 7.13. From definition 7.2, we have $\rho^{\widetilde{qr}}(t_{kl}) = \sum_{kl\in I}\rho^{\widetilde{qr}}(t_{kl})$ and $\rho^{\widetilde{qr}}(t') = \sum_{kl\in I}\rho^{\widetilde{qr}}(t_{kl})$. Hence, we arrive at the equality $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(t')$. □

Lemma 7.15 (External state independence for $\xrightarrow{\tau}$)**.** *If* $\Gamma;\ \widetilde{s} \vdash P$ *and* $t \xrightarrow{\tau} t'$ *where* $t = \oplus_{kl\in I}\ g_{\ kl}(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; [\widetilde{sqr} \mapsto |\psi_{kl}\rangle]; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet P; \widetilde{w}_{kl})$ *then* $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(t')$

Proof. By induction on the derivation of the transition $t \xrightarrow{\tau} t'$. Cases L-Par and L-Res are straight forward applications of the inductive hypothesis. The quantum state and distribution are unchanged for L-Com and L-Act. Therefore these cases are simple.

L-Qbit and L-Ns: We have the transition $\oplus_{kl\in I}\ g_{\ kl} t_{kl} \xrightarrow{\tau} \oplus_{kl\in I}\ g_{\ kl} t'_{kl}$, where for each $kl \in I$, $\rho(t'_{kl}) = \rho(t_{kl}) \otimes |0\rangle\langle 0|$. Therefore, $\rho^{\widetilde{qr}}(t'_{kl}) = \rho^{\widetilde{qr}}(t_{kl}) \otimes \langle 0|0\rangle = \rho^{\widetilde{qr}}(t_{kl})$.

L-PS: We have $|\psi_{kl}\rangle = |\alpha_{kl}\rangle_{\widetilde{s}}|\beta_{kl}\rangle_{\widetilde{q}}|\gamma_{kl}\rangle_{\widetilde{r}}$. Then

$$|\psi'_{kl}\rangle = \mathsf{PS}|\psi_{kl}\rangle = |\alpha_{kl}\rangle_{\widetilde{s}}\mathsf{PS}|\beta_{kl}\rangle_{\widetilde{q}}|\gamma_{kl}\rangle_{\widetilde{r}} = |\alpha'_{kl}\rangle_{\widetilde{s'}}|\beta'_{kl}\rangle_{\widetilde{q'}}|\gamma_{kl}\rangle_{\widetilde{r}}$$

Therefore, we have

$$\begin{aligned}\rho^{\widetilde{sr}}(t') &= \textstyle\sum\langle\beta'_{kl}||\beta'_{kl}\rangle|\alpha'_{kl}\rangle\langle\alpha'_{kl}||\gamma_{kl}\rangle\langle\gamma_{kl}| = \sum\langle\beta'_{kl}|(\mathsf{PS})^*\mathsf{PS}|\beta'_{kl}\rangle|\alpha'_{kl}\rangle\langle\alpha'_{kl}||\gamma_{kl}\rangle\langle\gamma_{kl}| \\ &= \textstyle\sum\langle\beta_{kl}||\beta_{kl}\rangle|\alpha'_{kl}\rangle\langle\alpha'_{kl}||\gamma_{kl}\rangle\langle\gamma_{kl}| = \rho^{\widetilde{sr}}(t).\end{aligned}$$

L-Expr: We have $P = F[e]$ and $P' = F[e']$ for some process context F and the hypothesis $u \longrightarrow_e u'$ where u is $\oplus_{kl\in I}\ g_{\ kl}(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; |\psi_{kl}\rangle; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet e; \widetilde{w}_{kl})$ and u' is $\oplus_{\substack{kl\in I \\ mn\in J_{kl}}}\ g_{\ kl}h_{klmn}(\widetilde{s} : \widetilde{\mathsf{NS}}, \widetilde{q} : \widetilde{\mathsf{Qbit}}, \widetilde{r} : \widetilde{\mathsf{Qbit}}; |\psi_{kl}\rangle; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{z} \bullet e'; \widetilde{w}_{kl}, \widetilde{w}_{klmn})$. By Lemma 7.14, we have $\rho^{\widetilde{qr}}(u) = \rho^{\widetilde{qr}}(u')$. It follows then from the definition that $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(u)$ and $\rho^{\widetilde{qr}}(t') = \rho^{\widetilde{qr}}(u')$, hence we get $\rho^{\widetilde{qr}}(t) = \rho^{\widetilde{qr}}(t')$ □

The next lemma proves that the action of a context on the quantum state is independent of the quantum subsystem owned by a process.

Lemma 7.16 (Independence of context transitions)**.** *Assume that* $\Gamma; \widetilde{s}_R \vdash R$*. Let* t *and* u *be configurations where*

$$t = \oplus_{kl\in I}\ g_{\ kl}(\widetilde{x} : \widetilde{T}; [\widetilde{q}_P\widetilde{q}_R\widetilde{q}_E\widetilde{s}_P\widetilde{s}_R\widetilde{s}_E \mapsto |\psi_{kl}\rangle]; \widetilde{q}_P, \widetilde{q}_R, \widetilde{s}_P, \widetilde{s}_R; \lambda\widetilde{y} \bullet R; \widetilde{w}_R)$$

$$u = \oplus_{mn\in J}\ h_{\ mn}(\widetilde{x} : \widetilde{T}; [\widetilde{q}_Q\widetilde{q}_R\widetilde{q}_E\widetilde{s}_Q\widetilde{s}_R\widetilde{s}_E \mapsto |\phi_{mn}\rangle]; \widetilde{q}_Q, \widetilde{q}_R, \widetilde{s}_Q, \widetilde{s}_R; \lambda\widetilde{y} \bullet R; \widetilde{w}_R)$$

If $\rho^{\widetilde{q}_P\widetilde{q}_E\widetilde{s}_P\widetilde{s}_E}(t) = \rho^{\widetilde{q}_Q\widetilde{q}_E\widetilde{s}_Q\widetilde{s}_E}(u)$ *and* $t \xrightarrow{\tau} t'$ *where*

$$t = \oplus_{\substack{kl\in I'_{ab} \\ ab\in K}}\ g'_{klab}(\widetilde{x} : \widetilde{T}; [\widetilde{q}_P\widetilde{q}'_R\widetilde{q}_E\widetilde{s}_P\widetilde{s}'_R\widetilde{s}_E \mapsto |\psi_{klab}\rangle]; \omega_P, \omega'_R; \lambda\widetilde{y}' \bullet R'; \widetilde{w}_{R_{ab}})$$

then there exists

$$u = \oplus_{\substack{mn\in J'_{ab} \\ ab\in K}}\ h'_{mnab}(\widetilde{x} : \widetilde{T}; [\widetilde{q}_Q\widetilde{q}'_R\widetilde{q}_E\widetilde{s}_Q\widetilde{s}'_R\widetilde{s}_E \mapsto |\phi_{mnab}\rangle]; \omega_Q, \omega'_R; \lambda\widetilde{y}' \bullet R'; \widetilde{w}_{R_{ab}})$$

such that $u \xrightarrow{\tau} u'$ *and* $\rho^{\widetilde{q}_P\widetilde{q}_E\widetilde{s}_P\widetilde{s}_E}(t') = \rho^{\widetilde{q}_Q\widetilde{q}_E\widetilde{s}_Q\widetilde{s}_E}(u')$

Proof. By induction on the derivation of $t \xrightarrow{\tau} t'$. □

The next two lemmas prove some simple results which are used in the proof of Theorem 7.19.

Lemma 7.17. *Let* $t = \oplus_{kl\in I}\ g_{\ kl}t_{kl}$ *and* $t' = \oplus_{kl\in I}\ g_{\ kl}t'_{kl}$ *then* $t \xrightarrow{\alpha} t'$ *if and only if* $\forall_{kl\in I}(t_{kl} \xrightarrow{\alpha} t'_{kl})$ *for* $\alpha \in \{.?[\cdot], \tau\}$

Proof. By induction on the derivation of $t \xrightarrow{\alpha} t'$. This is because process structure is constant for all $kl \in I$. □

Lemma 7.18. *Let* $t_{mn} = \oplus_{kl\in I_{mn}}\ g_{\ klmn}(\widetilde{x} : \widetilde{T}; \sigma_{klmn}; \omega; \lambda\widetilde{y} \bullet P; \widetilde{w}_{klmn})$ *and* $t_{klmn} = (\widetilde{x} : \widetilde{T}; \sigma_{klmn}; \omega; P\{\widetilde{w}_{klmn}/\widetilde{y}\})$ *then* $\forall_{mn\in J, kl\in I_{mn}}.(t_{klmn} \overset{c?[\widetilde{u}_{mn},\widetilde{q},\widetilde{s}]}{\longrightarrow_p} t'_{klmn})$ *if and only if* $\forall_{mn\in J}.(t_{mn} \overset{c?[\widetilde{u}_{mn},\widetilde{q},\widetilde{s}]}{\longrightarrow_p} t'_{mn})$

Proof. By induction on the derivation of $t_{mn} \overset{c?[\widetilde{u}_{mn},\widetilde{q},\widetilde{s}]}{\longrightarrow_p} t'_{mn}$. If the transition is derived from P-In then by L-In we have

$$\forall_{mn\in J,kl\in I_{mn}}.((\widetilde{x}:\widetilde{T};\sigma_{klmn};\omega;P) \overset{c?[\widetilde{u}_{mn},\widetilde{q},\widetilde{r}]}{\longrightarrow_p} (\widetilde{x}:\widetilde{T};\sigma_{klmn};\omega';P'))$$

and by Lemma 7.17, we have

$$\forall_{mn\in J} \oplus_{kl\in I_{mn}} \ g\ _{klmn}(\widetilde{x}:\widetilde{T};\sigma_{klmn};\omega;\lambda\widetilde{y}\bullet P;\widetilde{w}_{klmn}) \overset{c?[\widetilde{u}_{mn},\widetilde{q},\widetilde{r}]}{\longrightarrow_p}$$

$$\oplus_{kl\in I_{mn}}\ g\ _{klmn}(\widetilde{x}:\widetilde{T};\sigma_{klmn};\omega';\lambda\widetilde{y}\bullet P';\widetilde{w}_{klmn})$$

The cases for P-Par and P-Res are similar, making uses of L-Par and L-Res respectively. The argument is easily reversed to obtain the opposite direction. □

We are now in a position to prove that bisimilarity is preserved by parallel composition. To prove this, we define an equivalence relation that contains the pair $((\widetilde{x}:\widetilde{T};\sigma;\emptyset;P \mid R),(\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q \mid R))$ and that is closed under transitions from these configurations.

Theorem 7.19 (Parallel preservation for configurations)**.** *Assume that* $\Gamma \vdash P$, $\Gamma \vdash Q$, $\Gamma \vdash P \mid R$, *and* $\Gamma \vdash Q \mid R$. *If* $(\widetilde{x}:\widetilde{T};\sigma;\emptyset;P) \leftrightarrows (\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q)$ *then* $(\widetilde{x}:\widetilde{T};\sigma;\emptyset;P \mid R) \leftrightarrows (\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q \mid R)$.

Using this result, we prove that the bisimilarity of processes is preserved by parallel composition.

Proof. This proof is structured as follows. First, we introduce the notational conventions that will be used in this proof. We define an equivalence relation $\mathcal{R}$ on general configurations, in which the pair $(\widetilde{x}:\widetilde{T};\sigma;\emptyset;P \mid R),(\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q \mid R)$ from the statement is a particular case. The remainder of the proof is dedicated to proving that $\mathcal{R}$ is a probabilistic branching bisimulation.

Let P, Q and R be general processes and assume that $\Gamma;\widetilde{q}_P,\widetilde{s}_p \vdash P, \Gamma;\widetilde{q}_Q,\widetilde{s}_Q \vdash Q, \Gamma;$ $\widetilde{q}_P,\widetilde{s}_P,\widetilde{q}_R,\widetilde{s}_R \vdash P \mid R$, and $\Gamma;\widetilde{q}_Q,\widetilde{s}_Q,\widetilde{q}_R,\widetilde{s}_R \vdash Q \mid R$. Let, K be an arbitrary indexing set. For each $kl \in K$, let t_{kl} and u_{kl} be configurations given by

$$t_{kl} = \oplus_{ab\in I_{kl}}\ g\ _{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P;\lambda\widetilde{y}_P\bullet P;\widetilde{w}_{P_{abkl}})$$

$$u_{kl} = \oplus_{cd\in J_{kl}}\ h\ _{cdkl}(\widetilde{x}:\widetilde{T};\tau_{cdkl};\omega_Q;\lambda\widetilde{y}_Q\bullet Q;\widetilde{w}_{Q_{cdkl}})$$

where ω_P is $\widetilde{q}_P,\widetilde{s}_P$ and ω_Q is $\widetilde{q}_Q,\widetilde{s}_Q$ and $\sigma_{abkl} = [\widetilde{q}_P\widetilde{q}_R\widetilde{q}_E\widetilde{s}_P\widetilde{s}_R\widetilde{s}_E \mapsto |\psi_{abkl}\rangle]$, $\tau_{cdkl} = [\widetilde{q}_Q\widetilde{q}_R\widetilde{q}_E\widetilde{s}_Q\widetilde{s}_R\widetilde{s}_E \mapsto |\phi_{cdkl}\rangle]$ and $\widetilde{q}_E,\widetilde{s}_E$ are qubits and number states in the environment. For each $kl \in K$, we have $\rho_E(t_{kl}) = \rho_E(u_{kl})$.

We use the convention that configurations tw and uw are defined in relation to $\{t_{kl}\}$ and $\{u_{kl}\}$ where

$$tw = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; \lambda\widetilde{y}_P\widetilde{y}_R \bullet P \mid R; \widetilde{w}_{P_{abkl}}, \widetilde{w}_{R_{kl}})$$

$$uw = \oplus_{\substack{cd\in J_{kl}\\ kl\in K}} f_{\ kl} h_{cdkl}(\widetilde{x} : \widetilde{T}; \tau_{cdkl}; \omega_Q, \omega_R; \lambda\widetilde{y}_Q\widetilde{y}_R \bullet Q \mid R; \widetilde{w}_{Q_{cdkl}}, \widetilde{w}_{R_{kl}})$$

and $\{f_{kl}\}$ is a set of weights. Following this convention, the configurations $\{t'_{kl}\}, \{u'_{kl}\}, tw'$ and uw', for example, are related in same manner.

We use the convention that $\widetilde{y}_P$ (respectively $\widetilde{y}_Q, \widetilde{y}_R$) appear only in the process P (respectively Q, R). Therefore we are able to use the fact that the configurations $(\widetilde{x} : \widetilde{T}; \sigma; \omega; \lambda\widetilde{y}_P\widetilde{y}_R \bullet P; \widetilde{w}_P, \widetilde{w}_R)$ and $(\widetilde{x} : \widetilde{T}; \sigma; \omega; \lambda\widetilde{y}_P \bullet P; \widetilde{w}_P)$ are structurally congruent. This is used implicitly throughout the proof.

Now define the equivalence relation $\mathcal{R}_1$ as

$$\mathcal{R}_1 = \{(tw, uw) \mid \forall_{kl} \in K \,.\, (t_{kl} \leftrightarrows u_{kl})\}$$

Then define $\mathcal{R}$ to include probabilistic distributions, where

$$\mathcal{R} = \{(\boxplus_{m\in M} p_m \bullet t_m, \boxplus_{m\in M} p_m \bullet u_m) | \forall_{m\in M} \,.\, (t_m \leftrightarrows u_m \in \mathcal{R}_1)\}$$

Now we prove that $\mathcal{R}$ is a probabilistic branching bisimulation. By case analysis of the possible transitions of tw; we have an internal transition by P, output by P, input by P, communication from P, the respective transitions by R, and probabilistic transitions. In this proof we will use the convention that $t = \oplus_{kl\in K} f_{\ kl} t_{kl}$ and $tw = \oplus_{kl\in K} f_{\ kl} tw_{kl}$ in order to simplify the notation.

Internal transition by P:

If $tw \xrightarrow{\tau} tw'$ then by L-Par we have the hypothesis $t \xrightarrow{\tau} t'$ where

$$t' = \oplus_{\substack{ab\in I'_{kl}\\ kl\in K}} f_{\ kl} g'_{abkl}(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega'_P, \omega_R; \lambda\widetilde{y}'_P \bullet P'; \widetilde{w}'_{P_{abkl}})$$

and

$$tw' = \oplus_{\substack{ab\in I'_{kl}\\ kl\in K}} f_{\ kl} g'_{abkl}(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega'_P, \omega_R; \lambda\widetilde{y}'_P\widetilde{y}_R \bullet P' \mid R; \widetilde{w}'_{P_{abkl}}, \widetilde{w}_{R_{kl}})$$

Lemma 7.17 gives $\forall_{kl\in K}.(t_{kl} \xrightarrow{\tau} t'_{kl})$. Then, for each $kl \in K$, because $t_{kl} \leftrightarrows u_{kl}$ there exist configurations u'_{kl}, u''_{kl} such that $u_{kl} \Longrightarrow u'_{kl} \xrightarrow{\tau^+} u''_{kl}$ with $t_{kl} \leftrightarrows u'_{kl}$ and $t'_{kl} \leftrightarrows u''_{kl}$. Therefore by Lemma 7.17 we have $u \Longrightarrow u' \xrightarrow{\tau^+} u''$ where

$$u' = \oplus_{\substack{cd\in J'_{kl}\\ kl\in K}} f_{\ kl} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega'_Q, \omega_R; \lambda\widetilde{y}'_{Q'} \bullet Q; \widetilde{w}'_{Q_{cdkl}})$$

and

$$u'' = \oplus_{\substack{cd\in J''_{kl}\\ kl\in K}} f_{\ kl} h''_{cdkl}(\widetilde{x} : \widetilde{T}; \tau''_{cdkl}; \omega''_Q, \omega_R; \lambda\widetilde{y}''_Q \bullet Q'; \widetilde{w}'_{Q_{cdkl}}).$$

By L-Par we obtain the transitions $uw \Longrightarrow uw' \xrightarrow{\tau^+} uw''$ where

$$uw' = \oplus_{\substack{cd\in J'_{kl}\\ kl\in K}} f_{\ kl} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega'_Q, \omega_R; \lambda\widetilde{y}'_Q\widetilde{y}_R \bullet Q' \mid R; \widetilde{w}'_{Q_{cdkl}}, \widetilde{w}_{R_{kl}})$$

and

$$uw'' = \oplus_{\substack{cd\in J'_{kl}\\ kl\in K}} f_{\ kl} h''_{cdkl}(\widetilde{x} : \widetilde{T}; \tau''_{cdkl}; \omega''_Q, \omega_R; \lambda\widetilde{y}''_Q\widetilde{y}_R \bullet Q' \mid R; \widetilde{w}'_{Q_{cdkl}}, \widetilde{w}_{R_{kl}}).$$

Lemma 7.15 gives for each $kl \in K, \rho_E(t_{kl}) = \rho_E(t'_{kl})$ and $\rho_E(u_{kl}) = \rho_E(u'_{kl}) = \rho_E(u''_{kl})$ hence $\rho_E(t_{kl}) = \rho_E(u'_{kl})$ and $\rho_E(t'_{kl}) = \rho_E(u''_{kl})$. Therefore $(tw, uw') \in \mathcal{R}$ and $(tw', uw'') \in \mathcal{R}$.

Internal transition by R:

The transition $tw \xrightarrow{\tau} tw'$ has the hypothesis $w_1 \xrightarrow{\tau} w'_1$ where

$$w_1 = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; \lambda\widetilde{y}_R \bullet R; \widetilde{w}_{R_{kl}})$$

and

$$w'_1 = \oplus_{\substack{ab\in I'_{kl}\\ kl\in K}} f_{\ kl} g'_{abkl}(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega_P, \omega'_R; \lambda\widetilde{y}'_R \bullet R'; \widetilde{w}'_{R_{kl}}).$$

By Lemma 7.16 there exists w'_2 such that $w_2 \xrightarrow{\tau} w'_2$ where

$$w_2 = \oplus_{\substack{cd\in J_{kl}\\ kl\in K}} f_{\ kl} h_{cdkl}(\widetilde{x} : \widetilde{T}; \tau_{cdkl}; \omega_Q, \omega_R; \lambda\widetilde{y}_R \bullet R; \widetilde{w}_{R_{kl}})$$

and

$$w'_2 = \oplus_{\substack{cd\in J'_{kl}\\ kl\in K}} f_{\ kl} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega_Q, \omega'_R; \lambda\widetilde{y}'_R \bullet R'; \widetilde{w}'_{R_{kl}})$$

and $\rho^{\widetilde{q}_R\widetilde{q}_E\widetilde{s}'_R\widetilde{s}_E}(w'_1) = \rho^{\widetilde{q}_R\widetilde{q}_E\widetilde{s}'_R\widetilde{s}_E}(w'_2)$. By L-Par we have $uw \xrightarrow{\tau} uw'$. Let $t'_{kl} = \oplus_{ab\in I_{kl}} g_{\ abkl}(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega_P; \lambda\widetilde{y}_P \bullet P; \widetilde{w}_{abkl})$ and $u'_{kl} = \oplus_{cd\in J_{kl}} h_{\ cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega_Q; \lambda\widetilde{y}_Q \bullet$

$Q; \widetilde{w}_{cdkl})$. We must show that $\forall_{kl \in K} \,.\, (t'_{kl} \leftrightarrows u'_{kl})$. It is only necessary to consider the possible cases for the derivation of $w_i \xrightarrow{\tau} w'_i$ in which the quantum state is altered; these are R-Trans-NS, R-Measure-NS, R-PS-Measure, L-Qbit, L-Ns, L-PS (in all other cases $t_{kl} \leftrightarrows t'_{kl}$ and $u_{kl} \leftrightarrows u'_{kl}$).

R-Trans-NS: For σ'_{abkl}, we have $[|\psi'_{abkl}\rangle = (I_P \otimes U \otimes I_E)|\psi_{abkl}\rangle]$ for some unitary operator U, where I_P and I_E are identity operators on elements of P and E respectively. Similarly, for τ'_{cdkl}, we have $[|\phi'_{cdkl}\rangle = (I_Q \otimes U \otimes I_E)|\phi_{cdkl}\rangle]$. Now define a relation $\mathcal{R}_u$ such that $(t'_{kl}, u'_{kl}) \in \mathcal{R}_u$ if $t_{kl} \leftrightarrows u_{kl}$ and $\rho(t'_{kl}) = (I_P \otimes U \otimes I_E)^\dagger \rho(t_{kl})(I_P \otimes U \otimes I_E)$ and $\rho(u'_{kl}) = (I_Q \otimes U \otimes I_E)^\dagger \rho(u_{kl})(I_Q \otimes U \otimes I_E)$. If $t'_{kl} \xrightarrow{\tau} t''_{kl}$ then, by a straight forward induction on the derivation, we have $t_{kl} \xrightarrow{\tau} t'''_{kl}$ and $\rho(t''_{kl}) = (I'_P \otimes U \otimes I_E)^\dagger \rho(t'''_{kl})(I'_P \otimes U \otimes I_E)$. Because $t_{kl} \leftrightarrows u_{kl}$, we have $u_{kl} \Longrightarrow u''_{kl} \xrightarrow{\tau^+} u'''_{kl}$ and $t_{kl} \leftrightarrows u''_{kl}$ and $t''_{kl} \leftrightarrows u'''_{kl}$. By induction on the derivation of each transition in the sequence, we obtain $u'_{kl} \Longrightarrow u''''_{kl} \xrightarrow{\tau^+} u'''''_{kl}$ where $\rho(u''''_{kl}) = (I''_Q \otimes U \otimes I_E)^\dagger \rho(u''_{kl})(I''_Q \otimes U \otimes I_E)$ and $\rho(u'''''_{kl}) = (I'''_Q \otimes U \otimes I_E)^\dagger \rho(u'''_{kl})(I'''_Q \otimes U \otimes I_E)$. Therefore $(t'_{kl}, u''''_{kl}) \in \mathcal{R}_u$ and $(t''_{kl}, u'''''_{kl}) \in \mathcal{R}_u$. If $t'_{kl} \xrightarrow{c?[\widetilde{q},\widetilde{r},\widetilde{s}]} t''_{kl}$ then similar reasoning applies as in previous case. If $t'_{kl} \xrightarrow{c![V,\widetilde{X}]} \boxplus_m p_m \bullet t''_{klm}$ then $t_{kl} \xrightarrow{c![V,\widetilde{X}]} \boxplus_m p_m \bullet t'''_{klm}$ and $\rho(t''_{klm}) = (I'_P \otimes U \otimes I_E)^\dagger \rho(t'''_{klm})(I'_P \otimes U \otimes I_E)$ and then we have $\rho^{\widetilde{X}\widetilde{q}_R\widetilde{q}_E\widetilde{s}_R\widetilde{s}_E}(t''_{klm}) = (I_{\widetilde{X}} \otimes U \otimes I_E)^\dagger \rho^{\widetilde{X}\widetilde{q}_R\widetilde{q}_E\widetilde{s}_R\widetilde{s}_E}(t'''_{klm})(I_{\widetilde{X}} \otimes U \otimes I_E)$. Because $t_{kl} \leftrightarrows u_{kl}$, we have $u_{kl} \Longrightarrow u''_{kl} \xrightarrow{c![V,\widetilde{Y}]} \boxplus_m p_m \bullet u'''_{klm}$ and $t_{kl} \leftrightarrows u''_{kl}$ and $\forall_m \,.\, (t''_{klm} \leftrightarrows u'''_{kl})$. By induction on the derivation of each transition in this sequence, we obtain $u'_{kl} \Longrightarrow u''''_{kl} \xrightarrow{c![V,\widetilde{Y}]} \boxplus_m p_m \bullet u'''''_{klm}$ where $\rho(u''''_{kl}) = (I''_Q \otimes U \otimes I_E)^\dagger \rho(u''_{kl})(I''_Q \otimes U \otimes I_E)$ and $\forall_m \,.\, (\rho(u'''''_{klm}) = (I'''_Q \otimes U \otimes I_E)^\dagger \rho(u'''_{klm})(I'''_Q \otimes U \otimes I_E))$. Therefore $\rho^{\widetilde{Y}\widetilde{q}_R\widetilde{q}_E\widetilde{s}_R\widetilde{s}_E}(u'''''_{klm}) = (I_{\widetilde{X}} \otimes U \otimes I_E)^\dagger \rho^{\widetilde{Y}\widetilde{q}_R\widetilde{q}_E\widetilde{s}_R\widetilde{s}_E}(u'''_{klm})(I_{\widetilde{X}} \otimes U \otimes I_E)$ and because $\rho^{\widetilde{X}\ldots\widetilde{s}_E}(t'''_{klm}) = \rho^{\widetilde{Y}\ldots\widetilde{s}_E}(u'''_{klm})$ we have $\rho^{\widetilde{X}\ldots\widetilde{s}_E}(t''_{klm}) = \rho^{\widetilde{Y}\ldots\widetilde{s}_E}(u'''''_{klm})$. Therefore $(t'_{kl}, u''''_{kl}) \in \mathcal{R}_u$ and $\forall_m \,.\, (t''_{klm}, u'''''_{klm}) \in \mathcal{R}_u$. We find that $\mathcal{R}_u$ is a probabilistic branching bisimulation, hence $t'_{kl} \leftrightarrows u'_{kl}$.

R-Measure-NS-2/R-PS-Measure: We have a set of measurement operators $\{M_m\}$ such that $\rho(t'_{kl}) = \sum_m f_m (I_P \otimes M_m \otimes I_E)^\dagger \rho(t_{kl})(I_P \otimes M_m \otimes I_E)$ and $\rho(u'_{kl}) = \sum_m f_m (I_Q \otimes M_m \otimes I_E)^\dagger \rho(u'_{kl})(I_Q \otimes M_m \otimes I_E)$. We construct a relation $\mathcal{R}_m$ such that $(t'_{kl}, u'_{kl}) \in \mathcal{R}_m$ if $t_{kl} \leftrightarrows u_{kl}$ and $\rho(t'_{kl}) = \sum_m f_m (I_P \otimes M_m \otimes I_E)^\dagger \rho(t_{kl})(I_P \otimes M_m \otimes I_E)$ and $\rho(u'_{kl}) = \sum_m f_m (I_Q \otimes M_m \otimes I_E)^\dagger \rho(u_{kl})(I_Q \otimes M_m \otimes I_E)$. By similar reasoning to the previous case, we find that $\mathcal{R}_m$ is a bisimulation, hence $t_{kl} \leftrightarrows u_{kl}$.

L-Qbit/L-Ns: We have the relationships $\rho(t'_{kl}) = \rho(t_{kl}) \otimes |0\rangle\langle 0|$ and $\rho(u'_{kl}) = \rho(u_{kl}) \otimes |0\rangle\langle 0|$. We construct a relation and follow similar reasoning to the previous cases.

L-PS: We have the relationships $\rho(t'_{kl}) = \rho(t_{kl})$ and $\rho(u'_{kl}) = \rho(u_{kl})$ and follow similar reasoning.

Communication from P:

The derivation by L-Com is

$$\frac{\forall kl \in K, ab \in I_{kl} \,.\, ((t_{abkl} \overset{c![\widetilde{u}_{abkl},\widetilde{X}]}{\longrightarrow_p} t'_{abkl})(w_{abkl} \overset{c?[\widetilde{u}_{abkl},\widetilde{X}]}{\longrightarrow_p} w'_{abkl}))}{(tw \xrightarrow{\tau} tw')}$$

where

$$\begin{aligned}
t_{abkl} &= (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; P\{\widetilde{w}_{P_{abkl}}/\widetilde{y}_P\}),\\
t'_{abkl} &= (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega'_P, \omega_R; P'\{\widetilde{w}_{P_{abkl}}/\widetilde{y}_P\}),\\
w_{abkl} &= (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; R\{\widetilde{w}_{R_{kl}}/\widetilde{y}_R\}),\\
w'_{abkl} &= (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega'_R; P'\{\widetilde{w}_{R_{kl}}/\widetilde{y}_R\})
\end{aligned}$$

and

$$tw' = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega'_P, \omega'_R; \lambda\widetilde{y}_P\widetilde{y}_R \bullet P' \parallel R'; \widetilde{w}_{P_{abkl}}, \widetilde{w}_{R_{kl}}).$$

For each $kl \in K$, we derive by L-Out-Ns or L-Out-Qbit, the transition $(t_{kl} \overset{c![W_{kl},\widetilde{X}]}{\longrightarrow} t'_{kl_o})$ where $W_{kl} = \{w_{abkl} | ab \in I_{kl}\}$ and
$t'_{kl_o} = \boxplus_{m\in M_{kl}} p_m \bullet t_{klm_o}$ and $t_{klm_o} = (\oplus_{ab\in I_{klm}} (\ g_{abkl}/p_m)(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega'_P; \lambda\widetilde{y}_P \bullet P'; \widetilde{w}_{P_{abkl}})$.

For each $kl \in K$, because $t_{kl} \leftrightarrows u_{kl}$, we get u'_{kl}, u''_{kl_o} such that $u_{kl} \Longrightarrow u'_{kl} \overset{c![w_{kl},\widetilde{X}]}{\longrightarrow} u''_{kl_o}$ where

$$u'_{kl} = \oplus_{cd\in J'_{kl}} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega'_Q; \lambda\widetilde{y}'_Q \bullet Q'; \widetilde{w}'_{Q_{cdkl}}),$$

$$u''_{kl_o} \boxplus_{m\in M_{kl}} p_m \bullet u''_{klm_o},$$

$$u''_{klm_o} = \oplus_{cd\in J'_{kl}} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau''_{cdkl}; \omega''_Q; \lambda\widetilde{y}'_Q \bullet Q''; \widetilde{w}'_{cdkl})$$

and $t_{kl} \leftrightarrows u'_{kl}$ and for each $m \in M_{kl}, t'_{klm_o} \leftrightarrows u''_{klm_o}$ and $\rho_E(t_{klm_o}) = \rho_E(u'_{klm_o})$. Applying Lemma 7.17 to each step in $u_{kl} \Longrightarrow u'_{kl}$ gives $uw \Longrightarrow uw'$. By L-Com, we can derive the transition $uw' \xrightarrow{\tau} uw''$. Now by Lemma 7.15 we have for each $kl \in K$, $\rho_E(u_{kl}) = \rho_E(u'_{kl})$, therefore it follows that $\rho_E(t_{kl}) = \rho_E(u'_{kl})$ and because $t_{kl} \leftrightarrows u'_{kl}$ we have $(tw, uw') \in \mathcal{R}$.

By convention we have

$$t'_{kl} = \oplus_{ab\in I_{kl}} g_{\ abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega'_P; \lambda\widetilde{y}_P \bullet P'; \widetilde{w}_{P_{abkl}})$$

and

$$u''_{kl} = \oplus_{cd\in J'_{kl}} h'_{cdkl}(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega''_Q; \lambda\widetilde{y}_Q \bullet Q''; \widetilde{w}'_{Q_{cdkl}})$$

where σ_{abkl} and σ'_{abkl} (respectively τ'_{cdkl} and τ''_{cdkl}) differ by the permutation and renaming applied by L-Out-Ns/L-Out-Qbit. Because for each $m \in M_k, t'_{klm_o} \leftrightarrows u''_{klm_o}$, it follows that $t'_{kl} \leftrightarrows u''_{kl}$. It follows from $\rho_E(t_{kl}) = \rho_E(u'_{kl})$ and $\rho_E(t'_{kl}) = \rho_E(u''_{kl})$, therefore $(tw', uw'') \in \mathcal{R}$.

Communication from R:

The derivation by L-Com is

$$\frac{\forall kl \in K, ab \in I_{kl} \,.\, ((t_{abkl} \xrightarrow{c![\widetilde{o}_{kl},\widetilde{X}]}_p t'_{abkl})(w_{abkl} \xrightarrow{c?[\widetilde{o}_{kl},\widetilde{X}]}_p w'_{abkl}))}{(tw \xrightarrow{\tau} tw')}$$

Because the output is from R, the classical values $\widetilde{o}_{kl}$ that are transferred in the communication must be dependent on the index kl and be independent of ab. We rewrite the configurations so that $\widetilde{o}_{kl}$ are copies of the respective values in $\widetilde{w}_{R_{kl}}$; this helps us to maintain the distinction between variables appearing in the respective processes P, Q and R after the communication. Therefore we have

$$tw = \oplus_{\substack{ab \in I_{kl} \\ kl \in K}} f_{\ kl} g_{abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; \lambda \widetilde{y}_P \widetilde{y}_R \widetilde{z} \bullet P \mid R; \widetilde{w}_{P_{abkl}}, \widetilde{w}_{R_{kl}}, \widetilde{u}_{kl})$$

and

$$tw' = \oplus_{\substack{ab \in I_{kl} \\ kl \in K}} f_{\ kl} g_{abkl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega'_P, \omega'_R; \lambda \widetilde{y}_P \widetilde{y}_R \widetilde{z} \bullet P' \mid R'; \widetilde{w}_{P_{abkl}}, \widetilde{w}_{R_{kl}}, \widetilde{u}_{kl}).$$

For each $kl \in K$, because $\forall ab \in I_{kl} \,.\, (t_{abkl} \xrightarrow{c?[\widetilde{o_{kl}},\widetilde{X}]}_p t'_{abkl})$ we obtain by Lemma 7.18 that $(t_{kl} \xrightarrow{c?[\widetilde{o_{kl}},\widetilde{X}]}_p t'_{kl})$. Furthermore, because $t_{kl} \leftrightarrows u_{kl}$ there exist u'_{kl} and u''_{kl} such that $u_{kl} \Longrightarrow u'_{kl} \xrightarrow{c?[o_{kl},\widetilde{X}]} u''_{kl}$ where $t_{kl} \leftrightarrows u'_{kl}$ and $t'_{kl} \leftrightarrows u''_{kl}$. Then by applying L-Par to each step of the transition $u_{kl} \leftrightarrows u'_{kl}$ we obtain $uw \Longrightarrow uw'$, and by applying Lemma 7.18 to the transition $u'_{kl} \xrightarrow{c?[o_{kl},\widetilde{X}]} u''_{kl}$ gives $\forall cd \in J'_{kl} \,.\, (u'_{cdkl} \xrightarrow{c?[\widetilde{o_{kl}},\widetilde{X}]}_p u''_{cdkl})$. Therefore by L-Com we can derive the transition

$$\frac{\forall kl \in K, cd \in J'_{kl} \,.\, ((u'_{cdkl} \xrightarrow{c?[\widetilde{o}_{kl},\widetilde{X}]}_p u''_{cdkl})(w_{cdkl} \xrightarrow{c![\widetilde{o}_{kl},\widetilde{X}]}_p w'_{cdkl}))}{(uw \xrightarrow{\tau} uw')}$$

Using Lemma 7.15 we have $\rho_E(u_{kl}) = \rho_E(u'_{kl})$, hence $\rho_E(t_{kl}) = \rho_E(u'_{kl})$. Then we have $\rho_E(t'_{kl}) = tr_{\widetilde{X}} \rho_E(t_{kl})$ and $\rho_E(u''_{kl}) = tr_{\widetilde{X}} \rho_E(u'_{kl})$, from which we obtain $\rho_E(t'_{kl}) = \rho_E(u''_{kl})$. Therefore we have $(tw, uw') \in \mathcal{R}$ and $(tw', uw'') \in \mathcal{R}$ as required.

Output by P:

If $tw \xrightarrow{c![U,\widetilde{X}]} tw'$ where

$$tw' = \boxplus_{m\in M} p_m \bullet \oplus_{\substack{ab\in I_{klm}\\ kl\in K}} (\ f_{kl}g_{abkl}/p_m)(\widetilde{x}:\widetilde{T};\sigma'_{abkl};\omega'_P,\omega_R;\lambda\widetilde{y}_P\widetilde{y}_R \bullet P' \mid R;\widetilde{w}_{P_{abkl}},\widetilde{w}_{R_{kl}})$$

then the derivation by L-Out-Ns/L-Out-Qbit and P-Par has the hypothesis $\forall kl \in K, ab \in I_{kl}\,.\,(t_{abkl} \xrightarrow{c![\widetilde{o_{abkl}},\widetilde{X}]}_p t'_{abkl})$ where $U = \{\widetilde{o}_{abkl} | ab \in I_{kl}, kl \in K\}$ and

$$t_{abkl} = (\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P,\omega_R;P\{\widetilde{w}_{P_{abkl}}/\widetilde{y}_P\})$$

and

$$t'_{abkl} = (\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega'_P,\omega_R;P'\{\widetilde{w}_{P_{abkl}}/\widetilde{y}_P\}).$$

Then, for each $kl \in K$, by L-Out-Ns/L-Out-Qbit, we have $t_{kl} \xrightarrow{c![\widetilde{o_{kl}},\widetilde{X}]}_p t'_{kl}$ where

$$t_{kl} = \oplus_{ab\in I_{kl}}\ g_{\ abkl}t_{abkl},$$

$$t'_{kl} = \boxplus_{m\in M_{kl}} p_{klm} \bullet t'_{klm}$$

and

$$t'_{klm} = \oplus_{ab\in I_{klm}} \frac{g_{abkl}}{p_{klm}} t'_{abkl}.$$

For each $kl \in K$, because $t_{kl} \leftrightarrows u_{kl}$ then there exists u'_{kl}, u''_{kl} such that $u_{kl} \Longrightarrow u'_{kl} \xrightarrow{c![o_{kl},\widetilde{X}]} u''_{kl}$ and $t_{kl} \leftrightarrows u'_{kl}$ and $u''_{kl} = \boxplus_{m\in M_{kl}} p_{klm} \bullet u'''_{klm}$ and $\forall m \in M_{kl}\,.\,(t'_{klm} \leftrightarrows u''_{klm})$ and $\rho_E(t'_{klm}) = \rho_E(u''_{klm})$. Then for each $kl \in K$, the derivation of $u'_{kl} \xrightarrow{c![o_{kl},\widetilde{X}]} u''_{kl}$ gives the hypothesis $\forall cd \in J_{kl}\,.\,(u'_{cdkl} \xrightarrow{c![\widetilde{o_{abkl}},\widetilde{X}]}_p u''_{cdkl})$ where

$$\begin{aligned}
u'_{kl} &= \oplus_{cd\in J'_{kl}}\ h'_{\ cdkl}(\widetilde{x}:\widetilde{T};\tau'_{cdkl};\omega'_Q,\omega_R;\lambda\widetilde{y}'_Q \bullet Q';\widetilde{w}_{Q'_{cdkl}}),\\
u''_{klm} &= \oplus_{cd\in J'_{klm}}\ (\ h'_{cdkl}/p_{klm})(\widetilde{x}:\widetilde{T};\tau'_{cdkl};\omega''_Q,\omega_R;\lambda\widetilde{y}'_Q \bullet Q'';\widetilde{w}_{Q'_{cdkl}}),\\
u'_{cdkl} &= (\widetilde{x}:\widetilde{T};\tau'_{cdkl};\omega'_Q,\omega_R;Q'\{\widetilde{y}'_Q/\widetilde{w}_{Q'_{cdkl}}\}),\\
u''_{cdkl} &= (\widetilde{x}:\widetilde{T};\tau'_{cdkl};\omega''_Q,\omega_R;Q''\{\widetilde{y}'_Q/\widetilde{w}_{Q'_{cdkl}}\})
\end{aligned}$$

Now applying Lemma 7.17 to each step in the transitions $u_{kl} \Longrightarrow u'_{kl}$ gives $uw \Longrightarrow uw'$ where

$$u'_{kl} = \oplus_{\substack{cd\in J'_{kl}\\ kl\in K}}\ f_{\ kl}h'_{cdkl}(\widetilde{x}:\widetilde{T};\tau'_{cdkl};\omega'_Q,\omega_R;\lambda\widetilde{y}'_Q\widetilde{y}_R \bullet Q' \mid R;\widetilde{w}'_{Q_{cdkl}},\widetilde{R}_{kl})$$

Using P-Par and L-Out-Ns/L-Out-Qbit we can derive the transition $uw' \xrightarrow{c![U,\widetilde{X}]} uw''$ where

$$uw'' = \boxplus_{m\in M} p_m \bullet \oplus_{\substack{cd\in J'_{klm}\\ kl\in K}} (\ f_{kl}h'_{cdkl}/p_m)(\widetilde{x} : \widetilde{T}; \tau''_{cdkl}; \omega''_Q, \omega_R; \lambda\widetilde{y}'_Q\widetilde{y}_R \bullet Q'' \mid R; \widetilde{w}'_{Q_{cdkl}}, \widetilde{R}_{kl})$$

noting that $p_m = \sum_{kl\in K} p_{klm}$. Let

$$tw'_m = \oplus_{\substack{ab\in I_{klm}\\ kl\in K}} (\ f_{kl}g_{abkl}/p_m)(\widetilde{x} : \widetilde{T}; \sigma'_{abkl}; \omega'_P, \omega_R; \lambda\widetilde{y}_P\widetilde{y}_R \bullet P' \mid R; \widetilde{w}_{P_{abkl}}, \widetilde{R}_{kl}) \text{ and}$$
$$uw''_m = \oplus_{\substack{cd\in J'_{klm}\\ kl\in K}} \overline{f_{kl}h'_{cdkl}}\, p_m(\widetilde{x} : \widetilde{T}; \tau''_{cdkl}; \omega''_Q, \omega_R; \lambda\widetilde{y}'_Q\widetilde{y}_R \bullet Q'' \mid R; \widetilde{w}'_{Q_{cdkl}}, \widetilde{R}_{kl})$$

then for each $m \in M$ because $\forall kl \in K(t'_{klm} \leftrightarrows u''_{klm})$ and $\rho_E(t'_{klm}) = \rho_E(u'''_{klm})$ we have $(tw'_m, uw''_m) \in \mathcal{R}$. For each $kl \in K$, using Lemma 7.15 we have $\rho_E(u_{kl}) = \rho_E(u'_{kl})$, hence $\rho_E(t_{kl}) = \rho_E(u'_{kl})$ and therefore $(tw, uw') \in \mathcal{R}$ as required.

Output by R:

If $tw \xrightarrow{c![U,\widetilde{X}]} tw'$ then the derivation by L-Out-Ns/L-Out-Qbit and P-Par gives the hypothesis

$$\forall kl \in K, ab \in I_{kl}(\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega_R; R\{\widetilde{R}_{kl}/\widetilde{y}_R\}) \xrightarrow{c![\widetilde{o}_{kl},\widetilde{X}]}_p (\widetilde{x} : \widetilde{T}; \sigma_{abkl}; \omega_P, \omega'_R; R'\{\widetilde{R}_{mn}/\widetilde{y}_R\})$$

where $U = \{\widetilde{o}_{kl}\} = \{\widetilde{w}_m\}$ and each list of values $\widetilde{o}_{kl}$ is only dependent on kl since it must be continued within $\widetilde{w}_{R_{kl}}$. Because these transitions are independent of the quantum state, we get

$$\forall kl \in K, cd \in J_{kl}(\widetilde{x} : \widetilde{T}; \tau_{cdkl}; \omega_Q, \omega_R; R\{\widetilde{R}_{kl}/\widetilde{y}_R\}) \xrightarrow{c![\widetilde{o}_{kl},\widetilde{X}]}_p (\widetilde{x} : \widetilde{T}; \tau_{cdkl}; \omega_Q, \omega'_R; R'\{\widetilde{R}_{mn}/\widetilde{y}_R\})$$

By applying P-Par and L-Out-Ns/L-Out-Qbit we can derive the transition $uw \xrightarrow{c![U,\widetilde{X}]} uw'$ where

$$uw' = \boxplus_{m\in M} p'_m \bullet \oplus_{\substack{cd\in J'_{kl}\\ kl\in K_m}} (\ f_{kl}h_{cdkl}/p'_m)(\widetilde{x} : \widetilde{T}; \tau'_{cdkl}; \omega_Q, \omega_R; \lambda\widetilde{y}_Q\widetilde{y}_R \bullet Q \mid R; \widetilde{w}_{Q_{cdkl}}, \widetilde{R}_{kl})$$

For each $m \in M$ let $K_m = \{kl | \widetilde{o}_{kl} = \widetilde{w}_m\}$, then we have

$$p_m = \sum_{kl\in K_m} f_{kl} \sum_{ab\in I_{kl}} g_{abkl} = \sum_{kl\in K_m} f_{kl} = \sum_{kl\in K_m} f_{kl} \sum_{cd\in J_{kl}} h_{cdkl} = p'_m$$

Let $tw' = \boxplus_{m\in M} p_m \bullet tw'_m$ and $uw' = \boxplus_{m\in M} p_m \bullet uw'_m$ and let Π be the permutation operator corresponding to the permutation $\widetilde{q}_R\widetilde{s}_R\widetilde{q}_E\widetilde{s}_E \mapsto \widetilde{q}'_R\widetilde{s}'_R\widetilde{q}_E\widetilde{s}_E\widetilde{X}$ (this permutation is applied in the transformation from σ_{abkl} to σ'_{abkl} and from τ_{cdkl} to τ'_{cdkl} due to L-Out-Ns/L-Out-Qbit). Then we have

$$\rho^{\widetilde{q}'_R\cdots\widetilde{X}}(tw'_m) = \textstyle\sum_{kl\in K_m}(f_{kl}/p_m)(I_P\otimes\Pi)^\dagger\rho_E(t_{kl})(I_P\otimes\Pi) \text{ and}$$
$$\rho^{\widetilde{q}'_R\cdots\widetilde{X}}(uw'_m) = \textstyle\sum_{kl\in K_m}(f_{kl}/p_m)(I_Q\otimes\Pi)^\dagger\rho_E(u_{kl})(I_Q\otimes\Pi)$$

Because for each $kl \in K, \rho_E(t_{kl}) = \rho_E(u_{kl})$ and $\rho_E(tw'_m) = tr_{\widetilde{q}'_R\widetilde{s}'_R}(\rho^{\widetilde{q}'_R\cdots\widetilde{X}}(tw'_m))$ and $\rho_E(uw'_m) = tr_{\widetilde{q}'_R\widetilde{s}'_R}(\rho^{\widetilde{q}'_R\cdots\widetilde{X}}(uw'_m))$ we have $\rho_E(tw'_m) = \rho_E(uw'_m)$. Then, because for each $kl \in K, (t_{kl} \leftrightarrows u_{kl})$ we have $\forall m \in M\,.\,((tw'_m, uw'_m) \in \mathcal{R})$.

Input by P:

We have the transition $tw \xrightarrow{c?[\widetilde{o},\widetilde{X}]} tw'$ where

$$tw' = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega'_P,\omega_R,\widetilde{X};\lambda\widetilde{y}_P\widetilde{y}_R \bullet P' \mid R;\widetilde{w}_{P_{abkl}},\widetilde{w}_{R_{kl}})$$

The derivation of this transition by L-Par gives the hypothesis $t \xrightarrow{c?[\widetilde{o},\widetilde{X}]} t'$ where

$$t = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P;\lambda\widetilde{y}_P \bullet P;\widetilde{w}_{P_{abkl}}) \text{ and}$$
$$t' = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P,\widetilde{X};\lambda\widetilde{y}_P \bullet P';\widetilde{w}_{P_{abkl}})$$

Applying Lemma 7.17 gives $\forall kl \in K\,.\,(t_{kl} \xrightarrow{c?[\widetilde{o},\widetilde{X}]} t'_{kl})$ where

$$t'_{kl} = \oplus_{ab\in I_{kl}} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P.\widetilde{X};\lambda\widetilde{y}_P \bullet P';\widetilde{w}_{P_{abkl}}).$$

For each $kl \in K$, because $t_{kl} \leftrightarrows u_{kl}$ there exist configurations u'_{kl}, u''_{kl} such that $u_{kl} \Longrightarrow u'_{kl} \xrightarrow{c?[\widetilde{o},\widetilde{X}]} u''_{kl}$ where $t_{kl} \leftrightarrows u'_{kl}$ and $t'_{kl} \leftrightarrows u''_{kl}$. We now apply Lemma 7.17 to these transitions to get $u \Longrightarrow u' \xrightarrow{c?[\widetilde{o},\widetilde{X}]} u''$. Applying L-Par then gives the required transition $uw \Longrightarrow uw' \xrightarrow{c?[\widetilde{o},\widetilde{X}]} uw''$. For each $kl \in K$, we have $\rho_E(t'_{kl}) = tr_{\widetilde{X}}(\rho_E(u_{kl}))$ and $\rho_E(u''_{kl}) = tr_{\widetilde{X}}(\rho_E(u'_{kl}))$ and by Lemma 7.15 $\rho_E(u''_{kl}) = \rho_E(u'_{kl})$, then because $t_{kl} \leftrightarrows u'_{kl}$ and $t'_{kl} \leftrightarrows u''_{kl}$, we have $(tw, uw') \in \mathcal{R}$ and $(tw', uw'') \in \mathcal{R}$.

Input by R:

We have the transition $tw \xrightarrow{c?[\widetilde{o},\widetilde{X}]} tw'$ where

$$tw' = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P,\omega'_R,\widetilde{X};\lambda\widetilde{y}_P\widetilde{y}_R \bullet P \mid R';\widetilde{w}_{P_{abkl}},\widetilde{w}_{R_{kl}})$$

The derivation of this transition by L-Par gives the hypothesis $w_1 \xrightarrow{c?[\widetilde{o},\widetilde{X}]} w'_1$ corresponding to the action of R in isolation, where

$$w_1 = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P;\omega_R;\lambda\widetilde{y}_R \bullet R;\widetilde{w}_{R_{kl}}) \text{ and}$$
$$w'_1 = \oplus_{\substack{ab\in I_{kl}\\ kl\in K}} f_{\ kl} g_{abkl}(\widetilde{x}:\widetilde{T};\sigma_{abkl};\omega_P,\omega'_R;\lambda\widetilde{y}_R \bullet R';\widetilde{w}_{R_{kl}}).$$

Since this transition is independent from the quantum state we obtain the transition $w_2 \xrightarrow{c?[\widetilde{o},\widetilde{X}]} w'_2$ where

$$w_2 = \oplus_{\substack{cd\in J_{kl}\\ kl\in K}} f_{\ kl} h_{cdkl}(\widetilde{x}:\widetilde{T};\tau_{cdkl};\omega_Q;\omega_R;\lambda\widetilde{y}_R \bullet R;\widetilde{w}_{R_{kl}}) \text{ and}$$
$$w'_2 = \oplus_{\substack{cd\in J_{kl}\\ kl\in K}} f_{\ kl} h_{cdkl}(\widetilde{x}:\widetilde{T};\tau_{cdkl};\omega_Q,\omega'_R;\lambda\widetilde{y}_R \bullet R';\widetilde{w}_{R_{kl}})$$

Applying L-Par to this transition gives $uw \xrightarrow{c?[\widetilde{o},\widetilde{X}]} uw'$ where

$$uw' = \oplus_{\substack{cd\in J_{kl}\\ kl\in K}} f_{\ kl} h_{cdkl}(\widetilde{x}:\widetilde{T};\tau_{cdkl};\omega_Q,\omega_R;\lambda\widetilde{y}_Q\widetilde{y}'_R \bullet Q \mid R';\widetilde{w}_{Q_{cdkl}},\widetilde{w}_{R_{kl}}).$$

Because the elements $\widetilde{X}$ are contained within $\omega_E(\widetilde{q}_E,\widetilde{s}_E)$, we have $\widetilde{q}'_R = \widetilde{q}_R,\widetilde{X}$ or $\widetilde{s}'_R = \widetilde{s}_R,\widetilde{X}$ and $\widetilde{q}_E = \widetilde{q}'_E,\widetilde{X}$ or $\widetilde{s}_E = \widetilde{s}'_E,\widetilde{X}$. Therefore, $\rho^{\widetilde{q}_R\widetilde{s}_R\widetilde{q}_E\widetilde{s}_E}(tw) = \rho^{\widetilde{q}'_R\widetilde{s}'_R\widetilde{q}'_E\widetilde{s}'_E}(tw')$ and $\rho^{\widetilde{q}_R\widetilde{s}_R\widetilde{q}_E\widetilde{s}_E}(uw) = \rho^{\widetilde{q}'_R\widetilde{s}'_R\widetilde{q}'_E\widetilde{s}'_E}(uw')$. So $\forall kl \in K\,.\,(t_{kl} \leftrightarrows u_{kl})$, we have $(tw',uw') \in \mathcal{R}$. □

Theorem 7.20 (Parallel Preservation)**.** *If $P \leftrightarrows Q$ then for any process R such that $\Gamma \vdash P \mid R$ and $\Gamma \vdash Q \mid R$ then $P \mid R \leftrightarrows Q \mid R$.*

Proof. Because $P \leftrightarrows Q$ we have for all σ, $(\widetilde{x}:\widetilde{T};\sigma;\emptyset;P) \leftrightarrows (\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q)$. We define a relation $\mathcal{R}$, then we have for all σ, $((\widetilde{x}:\widetilde{T};\sigma;\emptyset;P \mid R),(\widetilde{x}:\widetilde{T};\sigma;\emptyset;Q \mid R)) \in \mathcal{R}$. By Theorem 7.19 $\mathcal{R}$ is a bisimulation, hence $P \mid R \leftrightarrows Q \mid R$. □

We now consider preservation with respect to other process constructions and can be shown that probabilistic branching bisimilarity is preserved by all process constructs except input and qubit or number state declarations.

Lemma 7.21. *Probabilistic branching bisimilarity is preserved by output prefix, action prefix, channel restriction and non-deterministic choice.*

Proof. This proof consists of a subset of the cases from the proof of Lemma 7.24. □

Theorem 7.22 (Probabilistic branching bisimilarity is a non-input, non-qubit congruence and non-number state congruence)**.** *If $P \leftrightarrows Q$ and for any non-input, non-qubit or non-number state context C if $\Gamma \vdash C[P]$ and $\Gamma \vdash C[Q]$ then $C[P] \leftrightarrows C[Q]$.*

Proof. Follows directly from Theorem 7.20 and Lemma 7.21. □

Definition 7.23 (Full probabilistic branching bisimilarity)**.** Processes P and Q are *full probabilistic branching bisimilar*, denoted $P \leftrightarrows^c Q$, if for all substitutions κ and all quantum states σ, $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; P\kappa) \leftrightarrows (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; Q\kappa)$.

We now show that full probabilistic branching bisimilarity is preserved by all process constructs. The following lemma is used in the proof of Lemma 7.25 which in turn is used in the proof of Theorem 7.26.

Lemma 7.24. *If $\forall ij \in I \,.\, ((\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; P) \leftrightarrows (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; Q))$ and $\sum_{ij \in I} g_{ij} = 1$ then $\oplus_{ij \in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; P) \leftrightarrows \oplus_{ij \in I}\ (\ \widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; Q)$*

Proof. There is a bisimulation $\mathcal{R}_1$ such that $\forall ij \in I, ((\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; P), (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; Q)) \in \mathcal{R}_1$. Now define a relation $\mathcal{R}_2$ as

$$\mathcal{R}_2 = \{(\oplus_{\substack{ij \in I \\ kl \in J_{ij}}}\ f_{\ ij} g_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \omega_P; \lambda \widetilde{y}_P \bullet P; \widetilde{w}_{P_{ijkl}}),$$
$$\oplus_{\substack{ij \in I \\ mn \in K_{ij}}}\ f_{\ ij} h_{ijmn}(\widetilde{x} : \widetilde{T}; \tau_{ijmn}; \omega_Q; \lambda \widetilde{y}_Q \bullet Q; \widetilde{w}_{Q_{ijmn}}),$$
$$\mid \forall ij \in I \,.\, (((\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \omega_P; \lambda \widetilde{y}_P \bullet P; \widetilde{w}_{P_{ijkl}}), (\widetilde{x} : \widetilde{T}; \tau_{ijmn}; \omega_Q; \lambda \widetilde{y}_Q \bullet Q; \widetilde{w}_{Q_{ijmn}})) \in \mathcal{R}_1)\}$$

Then extend this relation to include probabilistic configurations:

$$\mathcal{R}_3 = \{(\boxplus_{m \in M} p_m \bullet t_m, \boxplus_{m \in M} p_m \bullet u_m) \mid \forall m \in M \,.\, ((t_m, u_m) \in \mathcal{R}_2)\}$$

We now show that $\mathcal{R}_2 \cup \mathcal{R}_3$ is a bisimulation.

For $(t, u) \in \mathcal{R}_2$, if $t \xrightarrow{\alpha} t'$ where

$$t' = \oplus_{\substack{ij \in I \\ kl \in J'_{ij}}}\ f_{\ ij} g'_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma'_{ijkl}; \omega'_P; \lambda \widetilde{y}'_P \bullet P'; \widetilde{w}_{P'_{ijkl}})$$

then by Lemma 7.17 we have $\forall ij \in I \,.\, (t_{ij} \xrightarrow{\alpha} t'_{ij})$ where

$$t_{ij} = \oplus_{kl \in J_{ij}}\ g_{\ ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \omega_P; \lambda \widetilde{y}_P \bullet P; \widetilde{w}_{P_{ijkl}}) \text{ and}$$
$$t'_{ij} = \oplus_{kl \in J'_{ij}}\ g'_{\ ijkl}(\widetilde{x} : \widetilde{T}; \sigma'_{ijkl}; \omega'_P; \lambda \widetilde{y}'_P \bullet P'; \widetilde{w}'_{P_{ijkl}})$$

For each $ij \in I$, because $(t_{ij}, u_{ij}) \in \mathcal{R}_1$, there exists u'_{ij}, u''_{ij} such that $u_{ij} \Longrightarrow u'_{ij} \xrightarrow{\alpha} u''_{ij}$ where

$$u_{ij} = \oplus_{mn \in K_{ij}}\ h_{\ ijmn}(\widetilde{x} : \widetilde{T}; \tau_{ijmn}; \omega_Q; \lambda\widetilde{y}_Q \bullet Q; \widetilde{w}_{Q_{ijmn}}),$$
$$u'_{ij} = \oplus_{mn \in K'_{ij}}\ h'_{\ ijmn}(\widetilde{x} : \widetilde{T}; \tau'_{ijmn}; \omega'_Q; \lambda\widetilde{y}'_Q \bullet Q'; \widetilde{w}'_{Q_{ijmn}}),\ \text{and}$$
$$u''_{ij} = \oplus_{mn \in K''_{ij}}\ h''_{\ ijmn}(\widetilde{x} : \widetilde{T}; \tau''_{ijmn}; \omega''_Q; \lambda\widetilde{y}''_Q \bullet Q''; \widetilde{w}''_{Q_{ijmn}}).$$

By Lemma 7.17 we have $u \Longrightarrow u' \xrightarrow{\alpha} u''$ where

$$u' = \oplus_{\substack{ij \in I \\ mn \in K'_{ij}}}\ h'_{\ ijmn}(\widetilde{x} : \widetilde{T}; \tau'_{ijmn}; \omega'_Q; \lambda\widetilde{y}'_Q \bullet Q'; \widetilde{w}'_{Q_{ijmn}})$$
$$u'' = \oplus_{\substack{ij \in I \\ mn \in K''_{ij}}}\ h''_{\ ijmn}(\widetilde{x} : \widetilde{T}; \tau''_{ijmn}; \omega''_Q; \lambda\widetilde{y}''_Q \bullet Q''; \widetilde{w}''_{Q_{ijmn}})$$

and $(t, u') \in \mathcal{R}_1$ and $(t', u'') \in \mathcal{R}_2$.

If $t \xrightarrow{c![U,\widetilde{X}]} t'$ where $t' = \boxplus_{m \in M} p_m \bullet t'_m$ and

$$t'_m = \oplus_{\substack{ij \in I_m \\ kl \in J'_{ijm}}}\ (\ f_{ij}/p_m) g_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \omega'_P; \lambda\widetilde{y}'_P \bullet P'; \widetilde{w}'_{P_{ijkl}})$$

then by L-Out-Ns/L-Out-Qbit we can derive $\forall ij \in I\,.\,(t_{ij} \xrightarrow{c![\widetilde{U}_{ij},\widetilde{X}]} t'_{ij})$ where

$$t'_{ij} = \boxplus_{m \in M_{ij}} \bullet \oplus_{kl \in J_{ijm}}\ (\ g_{ijkl}/p_{ijm})(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \omega'_P; \lambda\widetilde{y}'_P \bullet P'; \widetilde{w}'_{P_{ijkl}})$$

and $U = \bigcup_{ij \in I} U_{ij}$ and $M = \bigcup_{ij \in I} M_{ij}$ and $p_m = \frac{\sum_{ij \in I_m} p_{ijm}}{\sum_{ij \in I} p_{ijm}}$.

For each $ij \in I$, because $(t_{ij}, u_{ij}) \in \mathcal{R}_1$ there exists u'_{ij} and u''_{ij} such that $u_{ij} \Longrightarrow u'_{ij} \xrightarrow{c![\widetilde{U}_{ij},\widetilde{X}]} u''_{ij}$. Using L-Out-Ns/L-Out-Qbit, we can derive the transitions $u \Longrightarrow u' \xrightarrow{c![\widetilde{U}_{ij},\widetilde{X}]} u''$, where $u'' = \boxplus_{m \in M} p_m \bullet u''_m$ and $(t, u') \in \mathcal{R}_2$ and $(t', u'') \in \mathcal{R}_3$ and $\forall m \in M\,.\,((t'_m, u''_m) \in \mathcal{R}_2)$. □

Lemma 7.25. *Full probabilistic branching bisimilarity is preserved by input prefix, output prefix, action prefix, qubit and number state declaration, channel restriction and non-deterministic choice.*

Proof. Because $P \leftrightarrows^c Q$, there exists a bisimulation $\mathcal{R}_1$ such that for all quantum states σ and for all substitutions k we have $((\widetilde{x} : \widetilde{T}; \sigma; \omega; P_k), (\widetilde{x} : \widetilde{T}; \sigma; \omega; Q_k)) \in \mathcal{R}_1$.

Input prefix: Let $\mathcal{R}_2$ be a relation such that $\forall \sigma, k' = \{\widetilde{v}, \widetilde{r}/\widetilde{y}\}$,

$$((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}]\,.\,P_{k'}), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}]\,.\,Q_{k'})) \in \mathcal{R}_2$$

We now show that $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ is a bisimulation. There is only one transition possibly, namely an input action. If $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}] \,.\, P_{k'}) \xrightarrow{c?[\widetilde{u},\widetilde{X}]} (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}, \widetilde{X}; P_{k'k}) = t'$ then we also have $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}] \,.\, Q_{k'}) \xrightarrow{c?[\widetilde{u},\widetilde{X}]} (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}, \widetilde{X}; Q_{k'k}) = u'$ and $(t', u') \in \mathcal{R}_1$.

L-PS: We define a relation $\mathcal{R}_2$ such that

$$((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{a : \mathsf{NS}, b : \mathsf{NS}{*}{=}\mathsf{PS}(c)\} \,.\, P_k), ((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{a : \mathsf{NS}, b : \mathsf{NS}{*}{=}\mathsf{PS}(c)\} \,.\, Q_k)) \in \mathcal{R}_2$$

We now show that $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ is a bisimulation. There is only one transition possibly, namely a τ transition.Then, If

$$(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{a : \mathsf{NS}, b : \mathsf{NS} *{=} \mathsf{PS}(c)\} \,.\, P_k) \longrightarrow (\widetilde{x'} : \widetilde{T'}; \sigma'; \widetilde{q'}, \widetilde{s'}; P_k)$$

we also have

$$(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{a : \mathsf{NS}, b : \mathsf{NS} *{=} \mathsf{PS}(c)\} \,.\, Q_k) \longrightarrow (\widetilde{x'} : \widetilde{T'}; \sigma'; \widetilde{q'}, \widetilde{s'}; Q_k)$$

where $((\widetilde{x'} : \widetilde{T'}; \sigma'; \widetilde{q'}, \widetilde{s'}; P_k), (\widetilde{x'} : \widetilde{T'}; \sigma'; \widetilde{q'}, \widetilde{s'}; Q_k) \in \mathcal{R}_1$. Hence, $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ where $\widetilde{s'} = \widetilde{s}, a, b$ and $\widetilde{q'} = \widetilde{q}/r$.

Output prefix: Define an equivalence relation $\mathcal{R}_2$ such that for all σ, k,

$$((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}] \,.\, P_{k'}), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{r}; c?[\widetilde{z}] \,.\, Q_{k'})) \in \mathcal{R}_2$$

whenever $P \leftrightarrows Q$. Then define $\mathcal{R}$ as the relation

$$\begin{aligned} \mathcal{R} = \{&(\boxplus_{m \in M} p_m \bullet \oplus_{ij \in I_m} g_{ij}(\widetilde{x} : \widetilde{T}; \sigma_{ijm}; \widetilde{q}, \widetilde{s}; \lambda \widetilde{y} \bullet P_k; \widetilde{w}_{ijm}), \\ &\boxplus_{m \in M} p_m \bullet \oplus_{ij \in I_m} g_{ij}(\widetilde{x} : \widetilde{T}; \sigma_{ijm}; \widetilde{q}, \widetilde{s}; \lambda \widetilde{y} \bullet Q_k; \widetilde{w}_{ijm})), \\ | \, \forall m \in M, ij \in I_m \,.\, &((\widetilde{x} : \widetilde{T}; \sigma_{ijm}; \widetilde{q}, \widetilde{s}; P_{kk'}), (\widetilde{x} : \widetilde{T}; \sigma_{ijm}; \widetilde{q}, \widetilde{s}; Q_{kk'})) \in \mathcal{R}_1 \cup \mathcal{R}_2\} \end{aligned}$$

where $k' = \{\widetilde{w}_{iim}/\widetilde{y}\}$. We also include non-probabilistic configurations in $\mathcal{R}$. The possible transitions are ultimately derived by either R-Plus, , R-PS-Measure, R-Measure-NS-2, R-Trans-NS, L-Out-Ns or L-Out-Qbit; we consider each case in turn:

R-Plus/: Let

$$\begin{aligned} t &= \oplus_{ij \in I} g_{ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda \widetilde{y} \bullet c![\widetilde{e}] \,.\, P_k; \widetilde{w}_{ij}) \text{ and} \\ t' &= \oplus_{ij \in I} g_{ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda \widetilde{y} z \bullet c![\widetilde{e'}] \,.\, P_k; \widetilde{w}_{ij}, \widetilde{u}_{ij}). \end{aligned}$$

If $t \xrightarrow{\tau} t'$ then $u \xrightarrow{\tau} u'$ where

$$u = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e}]\,.\,Q_k; \widetilde{w}_{ij}) \text{ and}$$
$$u' = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}z \bullet c![\widetilde{e'}]\,.\,Q_k; \widetilde{w}_{ij}, \widetilde{u}_{ij}).$$

We have

$$\forall ij \in I, ((\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; c![\widetilde{e'}]\{\widetilde{w}_{ij}u_{ij}/\widetilde{y}z\}P_k),$$
$$(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; c![\widetilde{e'}]\{\widetilde{w}_{ij}u_{ij}/\widetilde{y}z\}Q_k))\ \in \mathcal{R}_2.$$

Therefore $(t', u') \in \mathcal{R}$.

R-PS-Measure/R-Measure-NS-2: Let

$$t = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e}]\,.\,P_k; \widetilde{w}_{ij}) \text{ and}$$
$$t' = \oplus_{\substack{ij\in I \\ kl\in J_{ij}}}\ g_{\ ij}h_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{z} \bullet c![\widetilde{e'}]\,.\,P_k; \widetilde{w}_{ij}, \widetilde{u}_{ijkl}).$$

If $t \xrightarrow{\tau} t'$ then $u \xrightarrow{\tau} u'$ and as in previous case, we apply the same reasoning. Therefore $(t', u') \in \mathcal{R}$.

R-Trans-NS: Let

$$t = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e}]\,.\,P_k; \widetilde{w}_{ij}) \text{ and}$$
$$t' = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma'_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e'}]\,.\,P_k; \widetilde{w}_{ij}).$$

If $t \xrightarrow{\tau} t'$ then $u \xrightarrow{\tau} u'$ where

$$u = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e}]\,.\,Q_k; \widetilde{w}_{ij}) \text{ and}$$
$$u' = \oplus_{ij\in I}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma'_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \bullet c![\widetilde{e'}]\,.\,Q_k; \widetilde{w}_{ij}).$$

We have
$\forall ij \in I, ((\widetilde{x} : \widetilde{T}; \sigma'_{ij}; \widetilde{q}, \widetilde{s}; c![\widetilde{e'}]\{\widetilde{w}_{ij}/\widetilde{y}\}P_k), (\widetilde{x} : \widetilde{T}; \sigma'_{ij}; \widetilde{q}, \widetilde{s}; c![\widetilde{e'}]\{\widetilde{w}_{ij}/\widetilde{y}\}Q_k))\ \in \mathcal{R}_2.$

Therefore $(t', u') \in \mathcal{R}$.

L-Out-Ns: If

$$\oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{z}\,.\,c![\widetilde{y}, \widetilde{r}]\,.\,P_k; \widetilde{w}_{ij}, \widetilde{v}_{ij}) \xrightarrow{c![W,\widetilde{r}]} \boxplus_{m\in M} p_m \bullet t'_m$$

where $t'_m = \oplus_{ij\in I_m}\ (\ g_{ij}/p_m)(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}'; \lambda\widetilde{y}\widetilde{z}\,.\,P_k; \widetilde{w}_{ij}, \widetilde{v}_{ij})$ and $\widetilde{s} = \widetilde{s}'\widetilde{r}$ and $W = \{\widetilde{w}_{ij}\}$ then

$$\oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{z}\,.\,c![\widetilde{y}, \widetilde{r}]\,.\,Q_k; \widetilde{w}_{ij}, \widetilde{v}_{ij}) \xrightarrow{c![W,\widetilde{r}]} \boxplus_{m\in M} p_m \bullet u'_m$$

where $u'_m = \oplus_{ij\in I_m}\ (\ g_{ij}/p_m)(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}'; \lambda\widetilde{y}\widetilde{z}\,.\,Q_k; \widetilde{w}_{ij}, \widetilde{v}_{ij})$.

We have $P_k = P_{k'}$ and $Q_k = Q_{k'}$ where $k' = \{\widetilde{w}\widetilde{s}'/\widetilde{y}\}$. Then we have $\forall m \in M, ij \in I_m$

$$((\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}'; P_{k'}\{\widetilde{w}_{ij}\widetilde{v}_{ij}/\widetilde{y}\widetilde{z}\}), (\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}'; Q_{k'}\{\widetilde{w}_{ij}\widetilde{v}_{ij}/\widetilde{y}\widetilde{z}\}) \in \mathcal{R}_1$$

Therefore $\forall m \in M \,.\, (t'_m, u'_m) \in \mathcal{R}$.

Number state declaration: Define a relation

$$\mathcal{R}_2 = \{(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; (\mathsf{ns}\ y) \,.\, P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; (\mathsf{ns}\ y) \,.\, Q_k)) \mid (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; Q_k)) \in \mathcal{R}_1\}.$$

Then,

$$(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; (\mathsf{ns}\ y) \,.\, P_k) \xrightarrow{\tau} (\widetilde{x}' : \widetilde{T}'; \sigma'; \widetilde{q}, \widetilde{s}, r; P_{kk'})$$

where $k' = \{r/y\}$ and r is fresh. We also have $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; (\mathsf{ns}\ y) \,.\, Q_k) \xrightarrow{\tau} (\widetilde{x}' : \widetilde{T}'; \sigma'; \widetilde{q}, \widetilde{s}, r; Q_{kk'})$. Then

$$((\widetilde{x}' : \widetilde{T}'; \sigma'; \widetilde{q}, \widetilde{s}, r; P_{kk'}), (\widetilde{x}' : \widetilde{T}'; \sigma'; \widetilde{q}, \widetilde{s}, r; Q_{kk'})) \in \mathcal{R}_1\},$$

Hence, $\mathcal{R}_1 \cup \mathcal{R}_2$ is a bisimulation.

Restriction: Given a configuration $t = \oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; \lambda\widetilde{y} \bullet P; \widetilde{w}_{ij})$, lt t_n denote the corresponding configuration with a restriction $\oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \omega; \lambda\widetilde{y} \bullet (\mathsf{new}\ c)P; \widetilde{w}_{ij})$. Define a relation $\mathcal{R}_2 = \{(t_n, u_n) \mod (t, u) \in \mathcal{R}_1\}$.

If $t_n \xrightarrow{\alpha} t'_n$ then by L-Res we have $t \xrightarrow{\alpha} t'$. Because $(t, u) \in \mathcal{R}_1$ there exist u', u'' such that $u \Longrightarrow u' \xrightarrow{\alpha} u''$ and $(t, u') \in \mathcal{R}_1$ and $(t', u'') \in \mathcal{R}_1$. By L-Res we have $u_n \Longrightarrow u'_n \xrightarrow{\alpha} u''_n$ and $(t_n, u'_n) \in \mathcal{R}_2$ and $(t'_n, u''_n) \in \mathcal{R}_2$. We follow a similar reasoning for action prefix.

Action prefix: Define a relation

$$\mathcal{R}_2 = \{((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{e\} \,.\, P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{e\} \,.\, Q_k)) \mid ((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; Q_k)) \in \mathcal{R}_1\}$$

Then define

$$\mathcal{R}_3 = \{(\oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \,.\, \{e\} \,.\, P_k; \widetilde{w}_{ij}), \oplus_{ij}\ g_{\ ij}(\widetilde{x} : \widetilde{T}; \sigma_{ij}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y} \,.\, \{e\} \,.\, Q_k; \widetilde{w}_{ij})) \mid \forall ij((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{e\}\{\widetilde{w}_{ij}/\widetilde{y}\}P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \{e\}\{\widetilde{w}_{ij}/\widetilde{y}\}Q_k)) \in \mathcal{R}_2\}$$

Then for $(t, u) \in \mathcal{R}_3$, if $t \xrightarrow{\tau} t'$ where

$$t' = \{(\oplus_{ijkl}\ g_{\ ij}h_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{y}' \,.\, \{e'\} \,.\, P_k; \widetilde{w}_{ij}, \widetilde{w}_{ijkl}),$$

then $u \xrightarrow{\tau} u'$ where $u' = \{(\oplus_{ijkl}\ g_{\ ij} h_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; \lambda\widetilde{y}\widetilde{y}' \,.\, \{e'\} \,.\, Q_k; \widetilde{w}_{ij}, \widetilde{w}_{ijkl}),$ and for each i, j, k and l we have

$$\begin{aligned}&((\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; \{e\}\{\widetilde{w}_{ij}\widetilde{w}_{ijkl}/\widetilde{y}\widetilde{y}'\}P_k),\\ &(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; \{e\}\{\widetilde{w}_{ij}\widetilde{w}_{ijkl}/\widetilde{y}\widetilde{y}'\}Q_k)) \in \mathcal{R}_2\end{aligned}$$

Therefore $(t', u') \in \mathcal{R}_3$. If $t \xrightarrow{\tau} t'$ by L-Act where $t' = \{(\oplus_{ijkl}\ g_{\ ij} h_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; P_k)$ since variables $\widetilde{y}$ are not in P_k, then $u \xrightarrow{\tau} u'$ where $u' = \{(\oplus_{ijkl}\ g_{\ ij} h_{ijkl}(\widetilde{x} : \widetilde{T}; \sigma_{ijkl}; \widetilde{q}, \widetilde{s}; Q_k)$. By Lemma 7.24 we have $t' \leftrightarrows u'$.

Non-deterministic choice: There exists a bisimulation $\mathcal{R}_2$ such that $\forall \sigma, k \,.\, ((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; R), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; R)) \in \mathcal{R}_2$, and because $\alpha \,.\, P \leftrightarrows^c \alpha \,.\, Q$ from the previous cases, there is a bismiulation $\mathcal{R}_3$ such that $\forall \sigma, k \,.\, ((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, P_k), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, Q_k)) \in \mathcal{R}_3$. Now define a relation $\mathcal{R}_4$ such that

$$\mathcal{R}_4 = \{((\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, P_k + R), (\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, Q_k + R), \mid P \leftrightarrows^c Q\}$$

If we have the derivation

$$\frac{(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, P_k) \xrightarrow{\beta} t'}{(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, P_k + R) \xrightarrow{\beta} t'}$$

then $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, Q_k) \xrightarrow{\beta} u'$ and $(t', u') \in \mathcal{R}_3$. Therefore by L-Sum we have the transition $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, Q_k + R) \xrightarrow{\beta} u'$. The prefix α guarantees that this transition is strongly matched. If $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, P_k + R) \xrightarrow{\beta} t''$ is derived form the transition $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, R) \xrightarrow{\beta} t''$ then by L-Sum we have $(\widetilde{x} : \widetilde{T}; \sigma; \widetilde{q}, \widetilde{s}; \alpha \,.\, Q_k + R) \xrightarrow{\beta} t''$ where $t'' \leftrightarrows u''$, hence $(t'', u'') \in \mathcal{R}_2$. Therefore, $\mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$ is a bisimulation. □

Theorem 7.26 (Full probabilistic branching bisimilarity is a congruence)**.** *If $P \leftrightarrows^c Q$ then for any context $C[]$, if $C[P]$ and $C[Q]$ are typable then $C[P] \leftrightarrows^c C[Q]$.*

Proof. Follows directly from Theorem 4.14 and Lemma 7.25. □

7.3 Applications

7.3.1 The LOQC CNOT Gate in CQP : Revised first model

We have seen the CQP model of a experimental system that demonstrates the LOQC CNOT gate in Chapter 6 shown in Figure 6.9. Here, we present a revised model of the LOQC CNOT gate which is very similar to the previous model but with our new

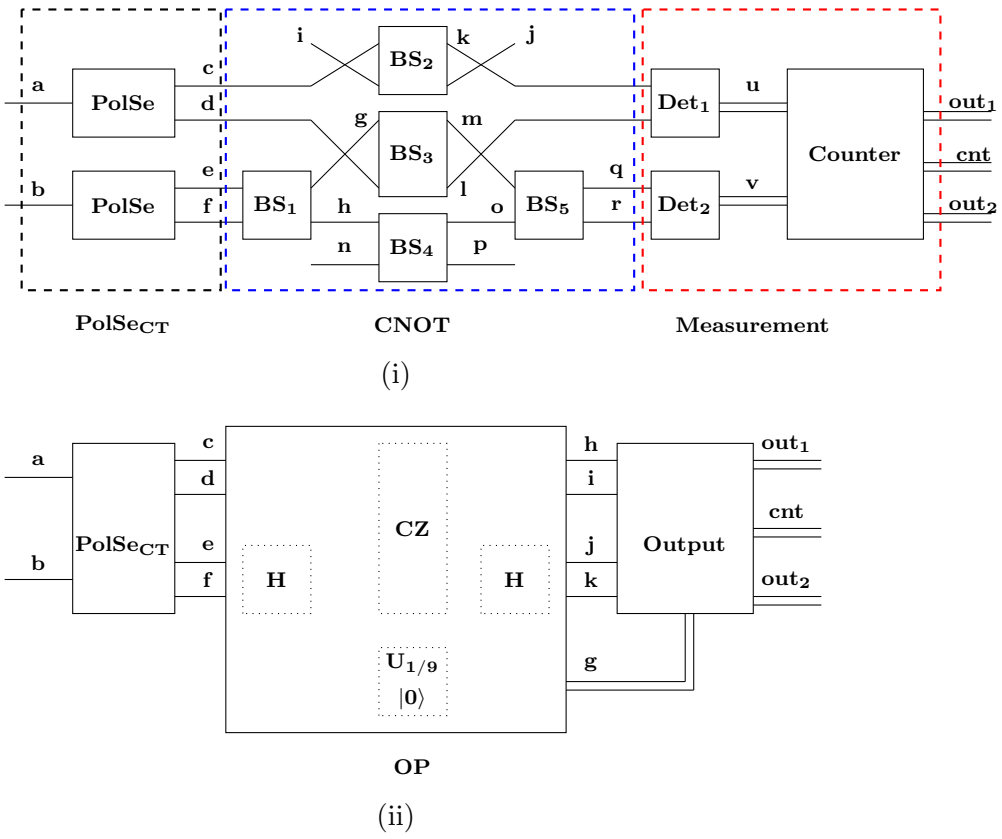


FIGURE 7.7: Model of LOQC CNOT gate: (i) $Model_1$.The dashed lines enclose the subsystems which are defined in the text. (ii) $Specification_1$. The dotted lines enclose the unitary operations involved in the system.

definitions. The structure of the new model is shown in Figure 7.7 (i). The differences between the two models are in the final two stages of the experimental system. First, in the older model, we used a detector which measures a number state or a presence of a photon in one path. The detector then sends the outcome in one channel. In the present model, we use a detector which measures a pair of number states, that is the presence of photon in two paths and sends the two measurement values in one channel. The second difference is in the definition of the process *Counter*.

We present the CQP definition of the new system:

$$Model_1(\widetilde{X}) = (\mathsf{new}\ \widetilde{Y})(PolSe_{CT}(\widetilde{U}) \mid CNOT(\widetilde{V}) \mid MMT(\widetilde{W}))$$

where each process is parameterised by their respective list of the channels ($\widetilde{X}, \widetilde{U}, \widetilde{V}$ and $\widetilde{W}$) on which it interacts with other processes. $\widetilde{X}$ contains channels a, b, out_1, cnt and out_2. $\widetilde{U}$ contains a, b, c, d, e, f and $\widetilde{W}$ contains $k, l, q, r, out_1, cnt, out_2$. The scope of the list of channels ($\widetilde{Y}$) is restricted, indicated by new in the definition. $\widetilde{Y}$ comprises of the channels $c, d, e, f, g, h, m, l, k, o, q, r, u$ and v. We have omitted the types from

our definitions, for brevity. Also, the definitions include a list of channels rather than individual channel names.

We recall the definitions of the processes $PolSe_{CT}$ and $CNOT$ from Section 6.3. The CQP definitions for $PolSe_{CT}$ and $CNOT$ are:

$$PolSe_{CT}(\widetilde{U}) = PolSe(\widetilde{A}) \mid PolSe(\widetilde{B}).$$

$$CNOT(\widetilde{V}) = (\mathsf{new}\ \widetilde{C})(\mathsf{ns}\ y, z)(BS_1(\widetilde{D}, \tfrac{1}{2}) \mid i![y]\,.\,\mathbf{0} \mid BS_2(\widetilde{E}, \tfrac{1}{3}) \mid$$
$$j?[y]\,.\,\mathbf{0} \mid BS_3(\widetilde{F}, \tfrac{1}{3}) \mid n![z]\,.\,\mathbf{0} \mid BS_4(\widetilde{G}, \tfrac{1}{3}) \mid p?[z]\,.\,\mathbf{0} \mid BS_5(\widetilde{H}, \tfrac{1}{2}))$$

Here $\widetilde{V}$ contains the channels c, d, e, f, k, l, q and r. The outputs of $CNOT$ are sent through the channels k, l, q and r, to the process MMT which performs the measurement.

$$MMT(\widetilde{W}) = (\mathsf{new}\ \widetilde{I})(Det_1(\widetilde{J}) \mid Det_2(\widetilde{K}) \mid Counter(\widetilde{L}))$$

Detectors Det_1 and Det_2 are annotated to match Figure 7.7(i) and measure the number states associated with the control and target qubits. The output of a detector are two classical values which represents the measurement outcome, that is the number of photons detected. The outcomes of the detector processes are given as inputs to the process $Counter$.

$$Counter(\widetilde{L}, b : \mathsf{bit}) = u?[c_0 : \mathsf{Int}, c_1 : \mathsf{Int}]\,.\,v?[t_0 : \mathsf{Int}, t_1 : \mathsf{Int}]\,.$$
$$out_1![\mathbf{if}\ (c_0 + c_1 = 1)\ \mathbf{then}\ c_1\ \mathbf{else}\ 0]\,.\,out_2![\mathbf{if}\ (t_0 + t_1 = 1)\ \mathbf{then}\ t_1\ \mathbf{else}\ 0]\,.$$
$$cnt![\mathbf{if}\ (c_0 + c_1 = 1)\ and\ (t_0 + t_1 = 1)\ \mathbf{then}\ b = 1\ \mathbf{else}\ b = 0]\,.\,\mathbf{0}$$

$Counter$ represents the coincidence measurement in optical experiments. Coincidence is observed by detecting two photons, one at channel u and the other at v. It also provides the correct output of the CNOT gate in terms of classical bits through the channels out_1 and out_2. The coincidence count is recorded as 1 at the output of the channel cnt. The unsuccessful outcomes of the CNOT gate are recorded as 0 at the three output channels. This is determined by the **if** ... **else** conditions in the definition. The position of these conditions is an important difference between the two models of the LOQC CNOT gate. In the first model presented in Chapter 4, we had the **if** ... **else** conditions included in the process definitions but in this model we include the conditions in the expression and not in the process. This is a significant change as it helps in proving the correctness of the model which is explained in the later sections of the Chapter.

When we consider the correctness of the system, we will prove that $Model_1$ is equivalent to the following $Specification_1$ process represented in the Figure 7.7 (ii). We use the

same process $PolSe_{CT}$ as the input for $Specification_1$.

$$Specification_1(\widetilde{E}) = (\mathsf{new}\ \widetilde{G})(PolSe_{CT}(\widetilde{U}) \mid OP(\widetilde{C}) \mid Output(\widetilde{D}))$$

OP performs the CNOT operation with a certain probability and is defined by

$$\begin{aligned} OP(\widetilde{C}) = (\mathsf{qbit}\ : q_2)\,.\,c?[s_0]\,.\,d?[s_1]\,.\,e?[s_2]\,.\,f?[s_3]\,.\,\{s_2, s_3 *\!\!= \mathsf{H}\}\,.\\ \{q_2 *\!\!= \mathsf{U}_{\frac{1}{9}}\}\,.\,\{(s_0, s_1), (s_2, s_3) *\!\!= \mathsf{CZ}\}\,.\,\{s_2, s_3 *\!\!= \mathsf{H}\}\,.\\ h![s_0]\,.\,i![s_1]\,.\,j![s_2]\,.\,k![s_3]\,.\,g![\mathsf{measure}\ q_2]\,.\,\mathbf{0} \end{aligned}$$

OP possesses a qubit q_2 (initialised to $|0\rangle$). A random bit is generated with certain probability ($\frac{8}{9}$ for bit 0) by measuring q_2 after the unitary operation with $U_{\frac{1}{9}}$. This is followed by a series of unitary operations namely Hadamard operation (H) which is applied twice on a pair of number states (s_2,s_3) and a controlled Z (CZ) where s_0, s_1 acts as the control pair and s_2, s_3 is the target pair. The number states and the random bit are then communicated to the process $Output$:

$$\begin{aligned} Output(\widetilde{D}) = g?[x : \mathsf{bit}]\,.\,h?[s_0]\,.\,i?[s_1]\,.\,j?[s_2]\,.\,k?[s_3]|\\ Det_1(h, i, l) \mid Det_2(j, k, m) \mid Outcome(l, m, out_1, out_2, cnt) \end{aligned}$$

$Output$ is a process which is a parallel compositions of the processes Det_1, Det_2 and $Outcome$. The first two processes are the detectors which measures the number states and the results are communicated internally to the process $Outcome$.

$$\begin{aligned} Outcome(l, m, out_1, out_2, cnt) = l?[c_0 : \mathsf{Int}, c_1 : \mathsf{Int}]\,.\,m?[t_0 : \mathsf{Int}, t_1 : \mathsf{Int}]\,.\\ out_1![\mathbf{if}\ (x = 1)\ \mathbf{then}\ c_1\ \mathbf{else}\ 0]\,.\,out_2![\mathbf{if}\ (x = 1)\ \mathbf{then}\ t_1\ \mathbf{else}\ 0]\,.\,cnt![x]\,.\,\mathbf{0} \end{aligned}$$

$Outcome$ gives the correct output in the form of classical bits of the CNOT operation when x equals one, which is artificially making the specification work with a certain probability ($\frac{1}{9}$). When x equals zero, the specification does not work and we get zero at all the output channels.

7.3.2 Execution of $Model_1$

Let $t = (\emptyset; \emptyset; \emptyset; Model_1)$ be the initial configuration. The semantics of CQP is non-deterministic and hence the transitions can proceed in different order. In the first few steps, the process $PolSe_{CT}$ receives qubits q_0 and q_1 from the environment, constructing a global quantum state $|\phi\rangle_q = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$. We get the configuration

as:

$$(q_1 : \mathsf{Qbit}, q_2 : \mathsf{Qbit}, q_1q_2 = |\phi\rangle_q; q_0, q_1; (PolSe_{CT}{}' \mid CNOT \mid MMT))$$

After some τ transitions corresponding to $PolSe_{CT}$ operations, the qubits are converted to the respective number states s_0, s_1, s_2 and s_3 by PS operator giving the quantum state $|\phi\rangle_s = \alpha|1010\rangle + \beta|1001\rangle + \gamma|0110\rangle + \delta|0101\rangle$. The configuration is now:

$$(\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi\rangle_s; s_0, s_1, s_2, s_3; (PolSe_{CT}{}'' \mid CNOT \mid MMT))$$

After another set of τ transitions corresponding to the $CNOT$ process, we get the state $|\phi\rangle_{out}$ which is given by Eq. (6.10). The configuration now becomes

$$(\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi\rangle_{out}; s_0, s_1, s_2, s_3; (CNOT' \mid MMT))$$

After the measurement by both detectors, the outcomes are communicated to the *Counter*. This happens internally and hence, we get the mixed configuration:

$$\oplus_{\substack{ij\geq 0\\ kl\geq 0}} g_{ij}h_{ijkl}\ (\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi_{ijkl}\rangle; s_0, s_1, s_2, s_3; \lambda\widetilde{y} \bullet Counter'; i, j, k, l)$$

Here $\widetilde{y}$ is a list of measurement outcomes (c_0, c_1, t_0 and t_1). The output transitions produces the configuration below, which is a mixed state.

$$\oplus_{i,j,k,l,m\in\{0,1\}}\ g_{ijm}h_{ijklm}\ (\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi_{ijkl}\rangle; \widetilde{s}; \lambda\widetilde{z} \bullet \mathbf{0}; i, j, k, l, m)$$

where $\widetilde{z}$ is c_1, t_1, b. The mixture contains both the successful and unsuccessful outcomes of $Model_1$.

7.3.3 Correctness of $Model_1$

We now sketch the proof that $Model_1 \leftrightarrows^c Specification_1$, which by Theorem 7.26 implies that the LOQC CNOT gate works in any context.

Proposition 7.27. *$Model_1 \leftrightarrows^c Specification_1$.*

Proof. First we prove that $Model_1 \leftrightarrows Specification_1$, by defining an equivalence relation $\mathcal{R}$ that contains the pair $((\widetilde{x} : \widetilde{T}; \sigma; \emptyset; Model_1), (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; Specification_1))$ for all σ and is closed under their transitions. $\mathcal{R}$ is defined by taking its equivalence classes to be the $F_i(\sigma)$ defined below, for all states σ, which group configurations according to the

sequences of observable transitions leading to them.

$$\begin{aligned} F_1(\sigma, q_0) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \stackrel{a?[q_0]}{\Longrightarrow} f \text{ and } P \in E\} \\ F_2(\sigma, q_0, q_1) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \stackrel{a?[q_0]b?[q_1]}{\Longrightarrow} f \text{ and } P \in E\} \\ F_3(\sigma, q_1) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \stackrel{a?[q_0]b?[q_1]out_1![c_1]}{\Longrightarrow} f \text{ and } P \in E\} \end{aligned}$$

$$F_4(\sigma) = \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \stackrel{a?[q_0]b?[q_1]out_1![c_1]out_2![c_3]}{\Longrightarrow} f \text{ and } P \in E\}$$

$$F_5(\sigma) = \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \stackrel{a?[q_0]b?[q_1]out_1![c_1]out_2![c_2]cnt![y]}{\Longrightarrow} f \text{ and } P \in E\}$$

Here E is $\{Model_1, Specification_1\}$ and we now prove that $\mathcal{R}$ is a probabilistic branching bisimulation. It suffices to consider transitions between F_i classes, as transitions within classes must be τ and are matched by τ. If $f, g \in F_1(\sigma)$ and $f \stackrel{a?[q_0]}{\longrightarrow} f'$ then $f' \in F_2(\sigma)$ and we find g', g'' such that $g \Longrightarrow g' \stackrel{a?[q_0]}{\longrightarrow} g''$ with $g' \in F_1(\sigma)$ and $g'' \in F_2(\sigma)$, so $(f, g') \in \mathcal{R}$ and $(f', g'') \in \mathcal{R}$ as required. Transitions from $F_2(\sigma)$,$F_3(\sigma)$ and $F_4(\sigma)$ are matched similarly. There are no transitions from $F_5(\sigma)$. There is no need for a probability calculation (case IV of Definition 7.3) because the probabilistic configurations do not arise as the measurement results are communicated internally. Finally, because $Model_1$ and $Specification_1$ have no free variables, their equivalence is trivially preserved by substitutions. □

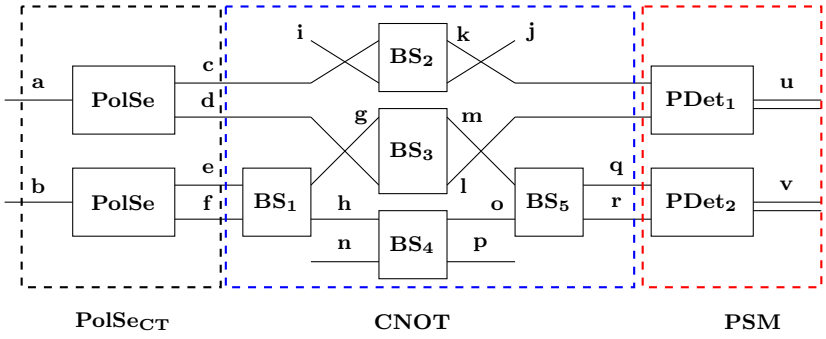
7.4 Post-selective Model

The first model includes an explicit implementation of the *post-selection* procedure, meaning that the specification process has to include the success probability of $\frac{1}{9}$. We now consider a more abstract model shown in Figure 7.8(a), by introducing a new measurement operator which includes *post-selection* and restricts attention to the successful outcomes. This is achieved by replacing the process MMT of our first model by the process PSM which performs *post-selective* measurement and enables a simpler specification to be used. The CQP definition of $Model_2$ is given as:

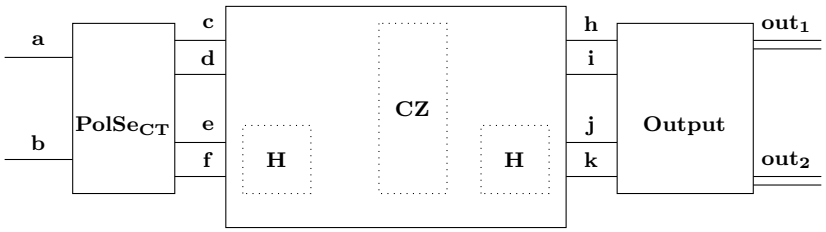
$$Model_2(\widetilde{A}) = (\mathsf{new}\ \widetilde{B})(PolSe_{CT}(\widetilde{C}) \mid CNOT(\widetilde{D}) \mid PSM(\widetilde{E}))$$

Processes $PolSe_{CT}$ and $CNOT$ are defined in the previous model. The difference between the two models lies in the measurement process. The process PSM is defined as

$$PSM(\widetilde{E}) = PDet_1(\widetilde{F}) \mid PDet_2(\widetilde{G}).$$



(a)



(b)

FIGURE 7.8: Model of LOQC CNOT gate: (a) $Model_2$.The dashed lines enclose the subsystems which are defined in the text. (b) $Specification_2$. The dotted lines enclose the unitary operations involved in the system.

$Model_2$ is equivalent to $Specification_2$ shown in Figure 7.8(b). The *post-selective* measurement is an in built operation in the new measurement semantics and this helps us to avoid the process that works on a certain probability to be used in the specification. In a way this demonstrates the flexible approach of process calculus and we define $Specification_2$ as the following:

$$OPCNOT(\widetilde{C}) = c?[s_0]\,.\,d?[s_1]\,.\,e?[s_2]\,.\,f?[s_3]\,.\,\{s_2, s_3 *= \mathsf{H}\}\,.$$
$$\{(s_0, s_1), (s_2, s_3) *= \mathsf{CZ}\}\,.\,\{s_2, s_3 *= \mathsf{H}\}\,.\,h![s_0]\,.\,i![s_1]\,.\,j![s_2]\,.\,k![s_3]\,.\,\mathbf{0}$$

$$Output(\widetilde{D}) = h?[s_0]\,.\,i?[s_1]\,.\,j?[s_2]\,.\,k?[s_3]\,.\,l![\mathsf{measure}\ s_0, s_1]\,.$$
$$m![\mathsf{measure}\ s_2, s_3] \mid Outcome(l, m, out_1, out_2)$$

$$Outcome(l, m, out_1, out_2) = l?[c_0 : \mathsf{Int}, c_1 : \mathsf{Int}]\,.\,m?[t_0 : \mathsf{Int}, t_1 : \mathsf{Int}]\,.\,out_1![c_1]\,.\,out_2![t_1]\,.\,\mathbf{0}$$

$$Specification_2(\widetilde{A}) = (\mathsf{new}\ \widetilde{E})(PolSe_{CT}(\widetilde{B}) \mid OPCNOT(\widetilde{C}) \mid Output(\widetilde{D}))$$

Execution of $Model_2$: Let $t = (\emptyset; \emptyset; \emptyset; Model_2)$ be the initial configuration. Like in previous case after receiving input qubits, we get the configuration as:

$$(q_1 : \mathsf{Qbit}, q_2 : \mathsf{Qbit}, q_1q_2 = |\phi\rangle_q; q_0, q_1; (PolSe_{CT}{}' \mid CNOT \mid PSM))$$

As before the qubits are converted to the number states after some τ operations and the configuration is now:

$$(\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi\rangle_s; s_0, s_1, s_2, s_3; (PolSe_{CT}{}'' \mid CNOT \mid PSM))$$

After another set of τ transitions corresponding to the $CNOT$ process, we get the state $|\phi\rangle_{out}$ which is given by Eq. **??**. The configuration now becomes

$$(\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi\rangle_{out}; s_0, s_1, s_2, s_3; (CNOT' \mid PSM))$$

Measurement by both detectors produces the following the mixed configuration:

$$\oplus_{\substack{ij\in\{0,1\},i\neq j \\ kl\in\{0,1\},k\neq l}} g_{ij}h_{ijkl}\ (\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi_{ijkl}\rangle; s_0, s_1, s_2, s_3; \lambda\widetilde{y} \bullet PSM'; j, l)$$

Here $\widetilde{y}$ is a list of *post-selective* measurement outcomes, which are given as output to the environment, that results in a probabilistic configuration given as:

$$\boxplus_{ij\in\{0,1\},kl\in\{0,1\}} g_{ij}h_{ijkl}(\widetilde{s} : \widetilde{\mathsf{NS}}; \widetilde{s} = |\phi_{ijkl}\rangle; s_0, s_1, s_2, s_3; \lambda\widetilde{y} \bullet \mathbf{0}; j, l)$$

7.4.1 Correctness of $Model_2$

Proposition 7.28. *$Model_2 \leftrightarrows^c Specification_2$.*

Proof. This is similar to the previous case with few differences. We will always get a correct output here since we do not consider any error and the probability of getting one of the outputs is $\frac{1}{4}$. Since, this model involves post-selection, two classical values are given as output to the environment that resulted in a probabilistic configuration, which was not the case for $Model_1$.

$$\begin{aligned}
F_1(\sigma, q_0) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \overset{a?[q_0]}{\Longrightarrow} f \text{ and } P \in E\} \\
F_2(\sigma, q_0, q_1) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \overset{a?[q_0]b?[q_1]}{\Longrightarrow\Longrightarrow} f \text{ and } P \in E\} \\
F_3(\sigma, q_1) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \overset{a?[q_0]b?[q_1]out_1![c_1]}{\Longrightarrow\Longrightarrow\Longrightarrow} f \text{ and } P \in E\} \\
F_4(\sigma) &= \{f \mid (\widetilde{x} : \widetilde{T}; \sigma; \emptyset; P) \overset{a?[q_0]b?[q_1]out_1![c_1]out_2![c_2]}{\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow} f \text{ and } P \in E\}
\end{aligned}$$

Here E is $\{Model_2, Specification_2\}$ and we now prove that $\mathcal{R}$ is a probabilistic branching bisimulation. It suffices to consider transitions between F_i classes, as transitions within classes must be τ and are matched by τ. If $f, g \in F_1(\sigma)$ and $f \overset{a?[q_0]}{\longrightarrow} f'$ then $f' \in F_2(\sigma)$ and we find g', g'' such that $g \Longrightarrow g' \overset{a?[q_0]}{\longrightarrow} g''$ with $g' \in F_1(\sigma)$ and $g'' \in F_2(\sigma)$, so $(f, g') \in \mathcal{R}$ and $(f', g'') \in \mathcal{R}$ as required. Transitions from $F_2(\sigma)$ and $F_3(\sigma)$ are matched similarly.

There are no transitions from $F_4(\sigma)$. There is a need for a probability calculation (case IV of Definition 7.3) because the probabilistic configurations arise as the measurement results are communicated internally. The probability of getting one of the outputs is $\frac{1}{4}$. Finally, because $Model_2$ and $Specification_2$ have no free variables, their equivalence is trivially preserved by substitutions. □

7.5 Discussion

In this section, we discuss the extension of the theory of equivalence of CQP to verify linear optical quantum computing. This is the first work in using quantum process calculus to verify LOQC.

LOQC is considered as one of the physical realisations of quantum computing and the aim of this work is to study the physical understanding of the concept of behavioural equivalence. The syntax and semantics presented in this chapter helps not only to describe or model LOQC but also to verify it. Conditional operations like **if**..**else** are essential in quantum computing. We have seen that the presence of mixed configuration in CQP allows each component in the mixed configuration to have the same process structure. This means that only values can differ between the components. Because of this reason it is complicated to include the conditional operations in processes at present and is part of a study in future. But, it is easier to include the conditional operations in expression configuration. This helps to model and verify LOQC in CQP.

We have defined certain linear optical elements in CQP like the combination of polarising beam splitter (PBS) and phase shifter (PR) to convert polarisation encoding of a qubit to spatial encoding, beam splitter and photon detectors. These elements were considered as they were potentially used in the experimental system that demonstrates the LOQC CNOT gate. Phase shifters like Quarter and half wave plates, which change the polarisation state of a photon, could also be explicitly defined in CQP but are not done in this thesis. We have described the conversion of polarisation encoding to spatial encoding by defining the process *PolSe*, which is a combination of PBS and PR. These elements could also be explicitly defined in CQP but we define the two as a combination, which generates spatial encoded qubits of same polarisation. Although, this is important in the experimental system but is not essential in integrated waveguide circuits exhibiting LOQC CNOT gate.

We have described and analysed two models ($Model_1$ and $Model_2$) of the linear optical experimental system that demonstrates a CNOT gate. Verification is performed by proving that $Model_1$ and $Model_2$ are equivalent to their respective specification process

($Specification_1$ and $Specification_2$). *Post-selection* is an essential property in LOQC and using our second model, we have also described and verified post-selection in CQP. These two models use different measurement semantics in order to work at different levels of abstraction. This shows that the process calculus is flexible enough to support a range of descriptions, from detailed hardware implementations up to more abstract specifications.

The specification processes for the models are defined in a manner to relate closely to the experimental system. For example, $Specification_2$ is given as:

$$Specification_2(\widetilde{A}) = (\mathsf{new}\ \widetilde{E})(PolSe_{CT}(\widetilde{B}) \mid OPCNOT(\widetilde{C}) \mid Output(\widetilde{D}))$$

where the processes $PolSe_{CT}, OPCNOT$ and $Output$ are given by the definitions in section 7.4. We can also define another process $Specification_3$, which expresses the same behaviour:

$$Specification_3(a,b,c,d) = a?[q : \mathsf{Qbit}]\,.\,b?[r : \mathsf{Qbit}]\,.\,\{q, r *= \mathsf{CNot}\}\,.\,c![\mathsf{measure}\ q]\,.\,d![\mathsf{measure}\ r]\,.\,\mathbf{0}$$

It can easily be shown that $Specification_3 \leftrightarrows Specification_2$ which in turn we get $Model_2 \leftrightarrows Specification_3$. This illustrates the flexibility of the process calculus in describing abstract specifications. The essential property that the equivalence is a congruence guarantees that equivalent processes remain equivalent in any context, and supports equational reasoning, which was discussed in Chapter 5. Another task would be to develop the equational theory of CQP that is applicable to number states..

We discuss primarily the application of CQP to LOQC which is concerned with the polarisation and spatial encoding of qubits. There has been other work in LOQC CNOT gate like photonic quantum gates that are applicable only to polarisation qubits [42] and CNOT gate which uses both polarisation and orbital angular momentum of photon [96]. Although these works are not discussed but the language could be easily extended to suit the application. This is an interesting line of future work, as it would increase the compatibility of CQP to be suited for applications of optical quantum computing.

Shor's algorithm operating on four qubits using the basic linear optical elements has been demonstrated in [142]. This is using the same optical elements that are described in this Chapter. Recently it has been demonstrated to experimentally verify the quantum complexity in linear optics [35]. Formally analysing Shor's algorithm in CQP using LOQC would be another interesting part in future work, and would provide a platform to learn about quantum complexity and formally verify it in CQP.

Chapter 8

CQP for higher dimensional protocols

This chapter provides the use of CQP in order to describe higher dimensional quantum protocols. The study encapsulates the theory of CQP that is described in the previous chapter in order to model higher dimensional quantum systems. The quantum process calculi, which have been developed to date, are defined for modelling systems that involves qubits, which are transmitted from process to process along communication channels. Experiments in quantum optics show that the physical systems that represent quantum information processing need not be limited to quantum bits (qubits) but can use higher dimensional systems, i.e. qudits (a quantum system with d-dimensional Hilbert space) [44]. We extend the operational semantics of CQP as described in chapter 4 to model higher dimensional quantum protocols namely qudit teleportation and superdense coding.

8.1 Preliminaries

8.1.1 Qudit

A *qudit* is a physical system which is described by a state vector $|\psi\rangle$. The state vector is an element of a d-dimensional complex Hilbert space, where d corresponds to the number of degrees of freedom of that system. Here we consider only systems with a finite number of degrees of freedom. As seen earlier, the orthonormal basis for qubits is $\{|0\rangle, |1\rangle\}$ and similarly for qudits, we fix each orthonormal basis state of the d-dimensional Hilbert space to correspond to an element of $\mathbb{Z}_d$. This is called the *computational basis* or *standard basis* [17, 133] which is given by $\{|0\rangle, |1\rangle, , |d-1\rangle\}$.

We can write the general state of a qudit as

$$|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$$

where $\alpha_i \in \mathbb{C}$ are complex amplitudes and $\Sigma_{i=0}^{d-1} \mid \alpha_i \mid^2 = 1$.

8.1.2 Quantum operators for qudits

We have discussed the quantum operators for qubits in Chapter 3. Using the same theory, we now introduce the elementary quantum gates or operators for d-dimensional systems.

Let $\mathbb{H}_\mathbb{A}$ and $\mathbb{H}_\mathbb{B}$ be d-dimensional Hilbert spaces, consider the set of $d^2 \times d^2$ unitary transformations $U \in U(d^2)$ that act on the two-qudit quantum system $\mathbb{H}_\mathbb{A} \otimes \mathbb{H}_\mathbb{B}$. The first gate we generalise is the CNOT gate. We have seen that in the context of qubits, the CNOT gate, is basically a mod-2 adder.

For qudits, this operator gives way to a mod-d adder, or a CNOT Right-Shift gate. Let $\mathsf{R_C} \in U(d^2)$ represent the CNOT Right-Shift gate that has control qudit $|\psi\rangle \in \mathbb{H}_\mathbb{A}$ and target qudit $|\phi\rangle \in \mathbb{H}_\mathbb{B}$. The action of $\mathsf{R_C}$ on the set of standard basis states $|m\rangle \otimes |n\rangle$ of $\mathbb{H}_\mathbb{A} \otimes \mathbb{H}_\mathbb{B}$ is given by

$$\mathsf{R_C}|m\rangle \otimes |n\rangle = |m\rangle \otimes |n \oplus m\rangle, \qquad m,n \in \mathbb{Z}_d$$

with $\oplus$ denoting addition modulo d.

Similarly, $\mathsf{L_C} \in \mathcal{U}(d^2)$ denote the generalised CNOT Left-Shift Gate which is defined as:

$$\mathsf{L_C}|m\rangle \otimes |n\rangle \equiv \mathsf{R_C^{-1}}|m\rangle \otimes |n\rangle = |m\rangle \otimes |n \ominus m\rangle$$

$\mathsf{L_C}$ is the inverse of $\mathsf{R_C}$ and also we note that $\mathsf{R_C}^d = \mathsf{I}$.

Generalised Pauli Gates

The next set of operators which are used to perform theoretical investigations of quantum systems are the Pauli operators. We now define the generalised Pauli operators for d-level quantum systems [17].

$$U = \{\mathsf{X^j Z^k} : j,k \in \mathbb{Z}_d\}.$$

where X and Z are defined by their action on the standard basis. This is given by

$$\mathsf{X^j}|m\rangle = |m \oplus j\rangle,$$

$$\mathsf{Z}^{\mathsf{k}}|m\rangle = e^{2\pi \frac{ikm}{d}}|m\rangle = \omega^{km}|m\rangle,$$

where ω is $e^{2\pi \frac{i}{d}}$. The indices j and k refer to shift and phase changes in the standard basis, respectively. Therefore the generalised Pauli operators can be represented in the form

$$U_{jk} = \Sigma_{m\in\mathbb{Z}_d}\omega^{km}|m \oplus j\rangle\langle m|$$

Note that X and Z do not commute; they obey

$$\mathsf{Z}^{\mathsf{k}}\mathsf{X}^{\mathsf{j}} = \omega^{jk}\mathsf{X}^{\mathsf{j}}\mathsf{Z}^{\mathsf{k}}$$

and $\mathsf{X}^{\mathsf{d}} = \mathsf{Z}^{\mathsf{d}} = \mathsf{I}$.

Generalised Hadamard Gate and Bell States

We now define a generalisation of the Hadamard gate which is useful in manipulating qudits for various applications [72].

$$\mathsf{H}|j\rangle = \frac{1}{\sqrt{d}}\sum_{m=0}^{d-1}\omega^{-jm}|m\rangle$$

This operator is also known as the quantum Fourier transform when $d = 2^n$. In that case it acts on n qudits. Here we assume it to be a basic gate on one single qudit, in the same way that the ordinary Hadamard gate is a basic gate on one qubit. This operator is symmetric and unitary, but not Hermitian.

A generalisation of the familiar Bell states for qudits has been introduced in [16]. The entangled state $|\Psi^{nm}\rangle_{AB}$ is called the *generalised Bell state* whereby A and B each possess one qudit of this two qudit state. These are a set of d^2 maximally entangled states and can be explicitly written as:

$$|\Psi^{nm}\rangle_{AB} = \frac{1}{\sqrt{d}}\sum_{j=0}^{d-1}\omega^{-jn}|j\rangle_A \otimes |j \oplus m\rangle_B$$

where m and n run from 0 to $d-1$. These states have the properties:

- $\langle\Psi^{nm}||\Psi^{n'm'}\rangle = \delta_{nn'}\delta_{mm'}$(orthonormality) and
- $tr(|\Psi^{nm}\rangle\langle\Psi^{nm}|) = \frac{1}{d}I$(maximal entanglement).

$$
\begin{array}{rcl}
T & ::= & \mathsf{Int} \mid \mathsf{Qdit} \mid \mathsf{Val} \mid \widehat{\ }[\tilde{T}] \mid \mathsf{Op}(1) \mid \mathsf{Op}(2) \mid \cdots \\
v & ::= & 0 \mid 1 \mid \cdots \mid \mathsf{H} \mid \cdots \\
e & ::= & v \mid x \mid \mathsf{measure}\ \tilde{e} \mid \tilde{e} *= e^{e} \mid e + e \\
P & ::= & \mathbf{0} \mid (P|P) \mid P + P \mid e?[\tilde{x} : \tilde{T}].P \mid e![\tilde{e}].P \mid \{e\}.P \mid [e].P \mid (\mathsf{qdit}\ x)P \mid \\
 & & (\mathsf{new}\ x : \widehat{\ }[T])P
\end{array}
$$

FIGURE 8.1: Syntax of higher dimensional CQP.

To construct the generalised Bell state, we fist apply the Hadamard transform $(\mathsf{H} \otimes \mathsf{I})$ to the qudit A. This acts on basis states $|n\rangle_A|m\rangle_B$ as follows

$$(\mathsf{H} \otimes \mathsf{I})|n\rangle_A|m\rangle_B = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \omega^{-jn}|j\rangle_A \otimes |m\rangle_B \tag{8.1}$$

where ω is a primitive d^{th} root of unity in $\mathbb{C}$ such that $\omega^d = 1$. Then we apply CNOT Right-Shift gate on Eq. (8.1) and we obtain the generalised Bell state given by Eq. (8.2).

$$|\Psi^{nm}\rangle_{AB} = \mathsf{R_C}[(\mathsf{H} \otimes \mathsf{I})|n\rangle_A|m\rangle_B] = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \omega^{-jn}|j\rangle_A \otimes |j \oplus m\rangle_B \tag{8.2}$$

In the later sections, we will use the particular Bell state (represented as Eq. (8.3) which is obtained by substituting m and n as 0.

$$|\Psi^{00}\rangle_{AB} = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} |j\rangle_A \otimes |j\rangle_B \tag{8.3}$$

We will now explain the syntax and semantics of CQP which are needed to describe higher dimensional quantum protocols.

8.2 Syntax and Semantics for higher dimensional CQP

8.2.1 Syntax

The syntax for higher dimensional CQP is defined by the grammar as shown in Figure 8.1. This is similar to the syntax for qubits described in Chapter 4 (shown in Figure 4.1) with a difference in the types T. In the previous syntax, we had qubit as one of the types T but now we have qudits (Qdit) instead of qubits. Another difference is in the data types which includes qudit of type qdit and n-qudit unitary operator types $\mathsf{Op}(n)$. We have a new process qudit declaration $(\mathsf{qdit}\ x)P$. The internal syntax of CQP

$$([q_0,\ldots,q_{n-1} \mapsto \alpha_0|\phi_0\rangle + \cdots + \alpha_{d^n-1}|\phi_{d^n-1}\rangle]; \omega; \mathsf{measure}\ q_0,\ldots,q_{r-1}) \longrightarrow_v \\ \oplus_{0\le m<d^r}\ g_m\ ([q_0,\ldots,q_{n-1} \mapsto \tfrac{\alpha_{l_m}}{\sqrt{g_m}}|\phi_{l_m}\rangle + \cdots + \tfrac{\alpha_{u_m}}{\sqrt{g_m}}|\phi_{u_m}\rangle]; \omega; \lambda x \bullet x; m)$$

(R-Measure)

where $l_m = d^{n-r}m, u_m = d^{n-r}(m+1)-1, g_m = |\alpha_{l_m}|^2 + \cdots + |\alpha_{u_m}|^2$

$$\oplus_{i\in I}\ g_i\ ([\widetilde{q} \mapsto |\psi_i\rangle]; \omega; \lambda\widetilde{x} \bullet (\mathsf{qdit}\ y)P; \widetilde{v}_i) \xrightarrow{\tau} \oplus_{i\in I}\ g_i\ ([\widetilde{q}, q \mapsto |\psi_i\rangle|0\rangle]; \omega, q; \lambda\widetilde{x} \bullet P\{q/y\}; \widetilde{v}_i)$$

where q is fresh (L-Qdit)

Figure 8.2: Modified transition rules for qudits

is the same as shown in Figure 4.2 with the reference to qudits instead of qubits. The values are supplemented with either qudit names q which are generated at run-time and substituted for the variables used in qdit declarations respectively.

8.2.2 Operational Semantics for qudits

The framework of CQP makes it easier to extend the language to describe higher dimensional systems. This is evident in the present section as we present the transition rules of CQP for higher dimensional systems. We modify the operational semantics of CQP using labelled transition system presented in Chapter 4. Most of the transition rules that are applicable to qubits could also be applied to qudits due to the general framework of CQP. This can be clearly seen by the fact that most of the transition rules are the same. Now, we present only the necessary transition rules which has been modified for qudits. The expression transition rules R-Plus, R-Trans and R-Context are virtually the same as that of qubits with the difference that the list of elements $\widetilde{q}$ now represents qudits.

The only change in the expression transition rules is in the measurement rule R-Measure. Since qudits are d-level quantum systems, we need to take into account the the dimension (d) of the system. We generalise the R-Measure rule for qudits and the second modification is in the qudit declaration rule which we present as L-Qdit. Essentially, the modification to the semantics is only to the two rules. The modified rules are presented in Figure 8.2.

As before, we work with *configurations* and one such is given as:

$$([q, r \mapsto \frac{1}{\sqrt{d}}\sum_{j=0}^{d-1}|j\rangle_q \otimes |j\rangle_r]; q; c![q]\,.\,P).$$

Here, the global quantum state consists of two qudits, q and r, in the specified state; that the process term under consideration has access to qudit q but not to qudit r ; and that the process itself is $c![q]\,.\,P$.

Example 8.1. $([q \mapsto \sum_{l=0}^{d-1} \alpha_l|l\rangle]; q; c![\mathsf{measure}\ q].P) \xrightarrow{\tau} \oplus_{i\in\{0,1,..,d-1\}} |\alpha_i|^2\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i)$.

Example 8.1 illustrates a transition that represents the effect of measuring a qudit q. The measurement is within a process which is going to output the result through the channel c. This is very similar to the example of qubit measurement (Example 4.1). The only difference is in the quantum state. In this case, the mixed configuration on the right of the transition is essentially an abbreviation of

$$|\alpha_0|^2([q \mapsto |0\rangle]; q; c![0].P\{0/x\}) \oplus |\alpha_1|^2([q \mapsto |1\rangle]; q; c![1].P\{1/x\} \ldots \\ \oplus|\alpha_{d-1}|^2([q \mapsto |d-1\rangle]; q; c![d-1].P\{d-1/x\})$$

We recall the concept of probabilistic branching that arises when the measurement result is given as output. The system is said to be in one branch or the other and is not a mixture of components. This indicates that the observer would know which of the possible states the system is in.

Example 8.2.

$$\oplus_{i\in\Omega}\ |\alpha|_i^2\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i) \xrightarrow{c![\Omega]} \boxplus_{i\in\Omega}|\alpha_i|^2([q \mapsto |i\rangle]; q; \lambda x \bullet P; i) \\ \overset{|\alpha_0|^2}{\rightsquigarrow}\ ([q \mapsto |0\rangle]; q; \lambda x \bullet P; 0)$$

Example 8.2 shows the effect of the output from the final configuration of Example 8.1. The output transition produces the intermediate configuration, which is a probability distribution over pure configurations. Because it comes from a mixed configuration, the output transition contains a *set* of possible values. From the intermediate configuration there are probabilistic transitions and the number of transitions depends on the dimension d, of which one is shown ($\overset{|\alpha_0|^2}{\rightsquigarrow}$). Here Ω is a set of values given by {0,1,..,d-1}.

Example 8.3.

$$\oplus_{i\in\Omega}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x\bullet(c![x].P \mid c?[y].Q); i) \xrightarrow{\tau} \oplus_{i\in\Omega}\ g_i\ ([q \mapsto |i\rangle]; q; \lambda x\bullet(P \mid Q\{x/y\}); i)$$

Example 8.3 illustrates qudit communication between the processes P and Q, which is similar to that of Example 4.3.

In the next section, we focus our attention to describe the execution of the higher dimensional quantum protocols namely teleportation and superdense coding.

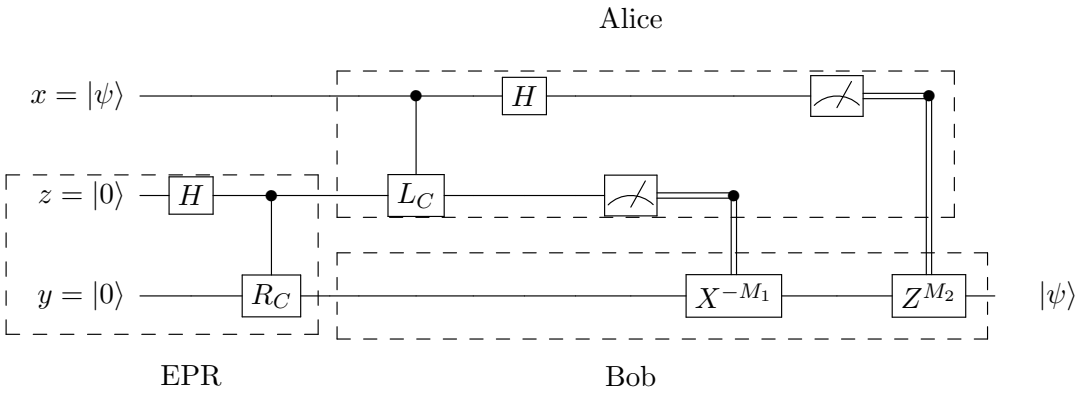


FIGURE 8.3: Qudit Teleportation

8.3 Qudit Protocols

8.3.1 Qudit Teleportation

We have seen in Chapter 3 that quantum teleporation is a protocol where a quantum state can be transferred from one location to another. The protocol explains how a qubit could be communicated from one user to another by using an entangled pair of qubits. In this section, we explain qudit teleportation which is an extension of the qubit teleportation.

Qudit teleportation [16, 72] is a protocol, which allows two users who share an entangled pair of qudits, to exchange an unknown quantum state by communicating only two classical values. The quantum circuit model of the protocol for qudits is shown in Figure 8.3. This circuit model is similar to the quantum teleportation for qubits shown in Figure 3.2. The difference is in the use of generalised quantum gates (CNOT and Hadamard) for qudits that was explained in section 8.1.2.

Although the circuit model represents the teleportation protocol, it defines the operation involved in the protocol, but it does not give a full description of the protocol itself. For example, the circuit model does not explain that the protocol consists of a definition of two users and the way in which they communicate, as well as the definition of the quantum operation involved in the protocol. The benefit of using our CQP model is that it not only provides the definition of the system but gives a clear and formal description of actions of the two users involved in the protocol.

Our model of qudit teleportation protocol consists of two processes: *Alice* and *Bob*, we say the sender is *Alice* and the receiver is *Bob*. *Alice* possesses the qudit labelled x which is in some unknown state $|\psi\rangle$; this is the qudit to be teleported. Qudits y and z are an entangled pair, which is generated by applying a Hadamard and CNOT- Right Shift gate to the qudits. The entangled state $|\Psi^{00}\rangle_{zy}$ is given by equation (14). Then

qudit z is given to *Alice* and qudit y is given to *Bob*. The CQP definition of *Alice* is as follows

$$Alice(c{:}\hat{}[\mathsf{Qdit}], e{:}\hat{}[\mathsf{Val},\mathsf{Val}], z : \mathsf{Qdit}) = c?[x{:}\mathsf{Qdit}] \,.\, \{x, z *{=}\, \mathsf{L}_c\} \,.\, \{x *{=}\, \mathsf{H}\} \,.\, e![\mathsf{measure}\ z, \mathsf{measure}\ x] \,.\, \mathbf{0}$$

Alice is parameterized by two channels, c and e. She receives the qudit on channel c. The type of c is $\hat{}[\mathsf{Qdit}]$. Channel e is where *Alice* sends the classical values resulting from her measurement. Each message on e consists of two classical values, as indicated by the type $\hat{}[\mathsf{Val},\mathsf{Val}]$.

We recall that the right hand side of the definition specifies the behaviour of *Alice*. The first term, $c?[x{:}\mathsf{Qdit}]$ specifies that a qudit is received from channel c and given the local name x. The term $\{x, z *{=}\, \mathsf{L}_c\}$ specifies that the CNOT- Left Shift operation is applied to qudits x and z and next term $\{x *{=}\, \mathsf{H}\}$ specifies that the Hadamard operation is applied to qudit x. The final term $e![\mathsf{measure}\ z, \mathsf{measure}\ x]$ indicates that the qudits x and z are measured which results in two classical values (M_1 and M_2) ranging from 0 to $d-1$ (where d is the dimension of the system). These two values are sent as a message on channel e.

We model the process *Bob*, which receives the two classical values from channel e (connected to *Alice*) and outputs the teleported qudit through channel d.

$$Bob(e{:}\hat{}[\mathsf{Val},\mathsf{Val}], d{:}\hat{}[\mathsf{Qdit}], y : \mathsf{Qdit}) = e?[M_1{:}\mathsf{Val}, M_2{:}\mathsf{Val}] \,.\, \{y *{=}\, \mathsf{X}^{-M_1}\} \,.\, \{y *{=}\, \mathsf{Z}^{M_2}\} \,.\, d![y] \,.\, \mathbf{0}$$

Using the classical values, *Bob* performs the necessary unitary operations on his qudit y as indicated by the terms $\{y *{=}\, \mathsf{X}^{-M_1}\}$ and $\{y *{=}\, \mathsf{Z}^{M_2}\}$. By doing this, *Bob* can recover the original state $|\psi\rangle$. The complete system is defined as follows.

$$Teleport = (\mathsf{qdit}\ y, z)(\{z *{=}\, \mathsf{H}\} \,.\, \{z, y *{=}\, \mathsf{R}_c\} \,.\, (\mathsf{new}\ e)(Alice(c, e, z) \mid Bob(e, d, y)))$$

8.3.2 Execution of Teleportation

Consider a qudit to be teleported is given by the quantum state $|\psi\rangle = \sum_{l=0}^{d-1} \alpha_l |l\rangle$. The initial configuration is $((\tilde{r}x = \sum_{l=0}^{d-1} \alpha_l |l\rangle_x); \emptyset; Teleport)$. In the first few steps, the system executes Qdit terms, the Hadamard operation and the CNOT Right-Shift (R_C), constructing the global quantum state:

$$(\tilde{r}pq_1q_2 = \sum_{l=0}^{d-1} \alpha_l |l\rangle_x \otimes \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} |k\rangle_{q_2} \otimes |k\rangle_{q_1}); q_1, q_2; (\mathsf{new}\ e)(Alice\{q_2/z\} \mid Bob\{q_1/y\}))$$

Alice receives the qudit x, in state ψ, from the environment, through the input transition $\xrightarrow{c?[x]}$, which gives us the 3 qudit state. After some τ transitions corresponding to *Alice's* Hadamard and CNOT Left-Shift (L_C) operations, we have:

$$(\tilde{r}xq_1q_2 = |\Phi_2\rangle); q_1, q_2, p; (\mathsf{new}\ e)(e![\mathsf{measure}\ q_2, \mathsf{measure}\ x]\,.\,\mathbf{0} \mid Bob\{q_1/y\}))$$

where $|\Phi_2\rangle = \frac{1}{d}\sum_{l,j,k=0}^{d-1} \omega^{-lj}\alpha_l|j\rangle_x \otimes |k \ominus l\rangle_{q_2} \otimes |k\rangle_{q_1}$. *Alice* does the measurement of her qudits in the *standard basis* and the results are communicated to *Bob* via channel e. Since the communication is internal within the system, this produces a mixed configuration which is given as:

$$\oplus_{j\in\Omega,s\in\Omega}((\tilde{r}xq_1q_2 = |\Psi_{js}\rangle); q_1, q_2, x; \lambda_{M_1,M_2}\,.\,(\mathsf{new}\ e)(e![M_1, M_2]\,.\,\mathbf{0} \mid Bob\{q_1/y\}); j, s)$$

where $|\Psi_{js}\rangle = \frac{1}{d^2}\sum_{j,s=0}^{d-1} |j\rangle_x|s\rangle_{q_2} \sum_{l=0}^{d-1} \omega^{-lj}\alpha_l|l \oplus s\rangle_{q_1}$. Depending on the classical values (M_1 and M_2) *Bob* does his unitary operations on his qudit q_1 to get the same state of the qudit x which *Alice* possesses. The qudit is then output through channel d.

$$\oplus_{j\in\Omega,s\in\Omega}((\tilde{r}pq_1q_2 = |\Psi'_{js}\rangle); q_2, p; \lambda_{M_1,M_2}\,.\,\mathbf{0}; j, s)$$

where $|\Psi'_{js}\rangle = \frac{1}{d^2}\sum_{l=0}^{d-1} \alpha_l|l\rangle_{q_1}$.

8.3.3 Superdense Coding for qudits

Now, we will describe the superdense coding protocol with respect to qudits. This protocol is considered the opposite of teleportation, where two values of classical information are communicated by exchanging a single qudit. Superdense coding also involves two users sharing a pair of entangled qudits. The quantum circuit for this protocol is given in Figure 8.4. The goal is to transmit some classical information from one user (Alice) to another (Bob). Like the previous protocol, this also begins with the preparation of an entangled pair. Alice is in possession of the first qudit, while Bob has possession of the second qudit. By sending the single qudit in her possession to Bob, it turns out that Alice can communicate two classical values (ranging from 0 to $d-1$) to Bob, where d is the dimension of the system. The CQP definition of the system is given below:

$SDC = (\mathsf{qdit}\ q_1, q_2)(\{q_1 *= \mathsf{H}\}\,.\,\{q_1, q_2 *= \mathsf{R}_c\}\,.\,(\mathsf{new}\ e)(Alice(c, e) \mid Bob(e, d)))$
$Alice(c{:}\hat{}[\mathsf{Val}, \mathsf{Val}], e{:}\hat{}[\mathsf{Qdit}]) = c?[a{:}\mathsf{Val}, b{:}\mathsf{Val}]\,.\,\{q_1 *= \mathsf{X}^b\}\,.\,\{q_1 *= \mathsf{Z}^a\}\,.\,e![q_1]\,.\,\mathbf{0}$
$Bob(e{:}\hat{}[\mathsf{Qdit}], d{:}\hat{}[\mathsf{Val}, \mathsf{Val}]) = e?[q_1{:}\mathsf{Qdit}]\,.\,\{q_1, q_2 *= \mathsf{L}_c\}\,.\,\{q_1 *= \mathsf{H}\}\,.$
$d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2]\,.\,\mathbf{0}$

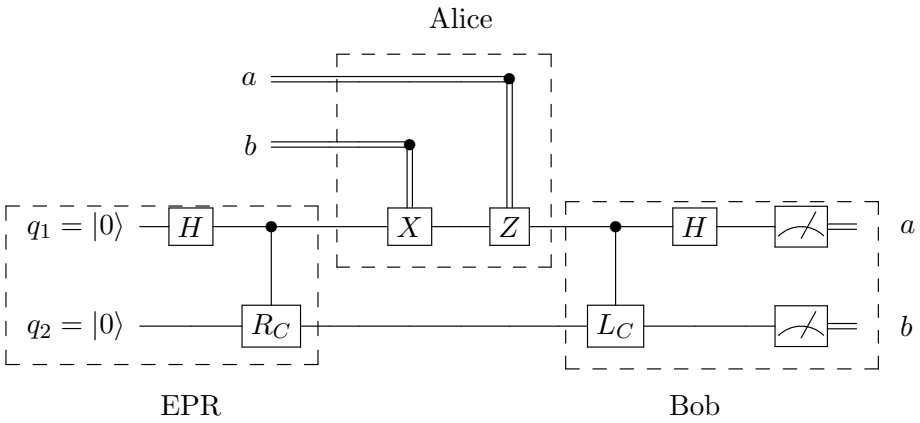


FIGURE 8.4: Superdense Coding Protocol

This CQP model, unlike the circuit model (Figure 8.4), is able to clearly describe the actions of the two users using the processes *Alice* and *Bob*. *Alice* takes one qudit (q_1) of the pair and *Bob* takes the other (q_2). The classical values to be transmitted are labelled a and b. When *Alice* is ready to send, she applies a combination of the X and Z operators to qudit q_1 depending on the values a and b.

After *Alice* has done her encoding, she send her single qudit to *Bob*. Now that *Bob* has both qudits, he can determine which encoding *Alice* used, and therefore the corresponding values a and b. First, he applies a CNOT Left shift operator to qudits q_1 and q_2, followed by the Hadamard operator applied to q_1. He then measures both of these qudits to reveal the respective values. Since the state he measures is not a superposition, the outcome will be certain.

8.3.4 Execution of SDC

In this section, we show the step-by-step execution of the superdense coding protocol, in order to illustrate the operational semantics. Teleportation can also be executed in a similar way according to the transition rules.

Consider an arbitrary quantum state $\tilde{r} = |\psi\rangle$. Let $s = (\tilde{r} = |\psi\rangle; \emptyset; SDC)$, then the execution of superdense coding is as follows.

$$s \xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_1\rangle); q_1, q_2; (\mathsf{new}\ e)(Alice(c, e) \mid Bob(e, d))$$
$$\xrightarrow{c?[a,b]} ((\tilde{r} q_1 q_2 = |\psi_1\rangle); q_1, q_2; (\mathsf{new}\ e)(\{q_1 *= \mathsf{X}^b\} . \{q_1 *= \mathsf{Z}^a\} . e![q_1] . \mathbf{0} \mid Bob(e, d)))$$

$$\xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_2\rangle); q_1, q_2; (\mathsf{new}\ e)(\{q_1 *= \mathsf{Z}^a\} . e![q_1] . \mathbf{0} \mid Bob(e, d)))$$

$$\xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_3\rangle); q_1, q_2; (\mathsf{new}\ e)(e![q_1] . \mathbf{0} \mid Bob(e, d)))$$

$$\xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_3\rangle); q_1, q_2; (\mathsf{new}\ e)(\{q_1, q_2 *= \mathsf{L}_c\} . \{q_1 *= \mathsf{H}\} . d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2] . \mathbf{0}))$$

$$\xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_4\rangle); q_1, q_2; (\mathsf{new}\ e)(\{q_1 *= \mathsf{H}\} . d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2] . \mathbf{0}))$$

$$\xrightarrow{\tau} ((\tilde{r} q_1 q_2 = |\psi_5\rangle); q_1, q_2; (\mathsf{new}\ e)(d![\mathsf{measure}\ q_1, \mathsf{measure}\ q_2] . \mathbf{0}))$$

$$\xrightarrow{d![a,b]} ((\tilde{r} q_1 q_2 = |\psi_6\rangle); q_1, q_2; \mathbf{0})$$

where

$$|\psi_1\rangle = \tfrac{1}{\sqrt{d}} \textstyle\sum_{j=0}^{d-1} |j\rangle_{q_1} \otimes |j\rangle_{q_2}$$

$$|\psi_2\rangle = \tfrac{1}{\sqrt{d}} \textstyle\sum_{j=0}^{d-1} |j \oplus b\rangle_{q_1} \otimes |j\rangle_{q_2}$$

$$|\psi_3\rangle = \tfrac{1}{\sqrt{d}} \textstyle\sum_{j=0}^{d-1} \omega^{a(j\oplus b)} |j \oplus b\rangle_{q_1} \otimes |j\rangle_{q_2}$$

$$|\psi_4\rangle = \tfrac{1}{\sqrt{d}} \textstyle\sum_{j=0}^{d-1} \omega^{a(j\oplus b)} |j \oplus b\rangle_{q_1} \otimes |j \ominus (j \oplus b)\rangle_{q_2} = \tfrac{1}{\sqrt{d}} \textstyle\sum_{k=0}^{d-1} \omega^{ak} |k\rangle_{q_1} \otimes |-b\rangle_{q_2}$$

$$|\psi_5\rangle = |\psi_6\rangle = |a\rangle_{q_1} \otimes |-b\rangle_{q_2}$$

8.4 Orbital Angular Momentum (OAM) of light

In the previous sections, we have provided a theoretical framework which helps us to describe a higher dimensional quantum system using CQP. The higher-dimensionality of the quantum system could be realised with the use of an intrinsic property of light called as the orbital angular momentum (OAM). The present section describes this intrinsic property of light. It has been known that light carries linear momentum [119]. In classical physics, angular momentum $\mathbf{J}$ is an intrinsic property of light and in most

cases it can be separated into two parts: spin angular momentum (SAM) [28] represented as **S** and orbital angular momentum (OAM) represented as **L** [19]. We get:

$$\mathbf{J} = \mathbf{L} + \mathbf{S} \tag{8.4}$$

In optical quantum computing, a qubit is represented by a single photon, which is an elementary particle or the quantum of light. A photon can carry both SAM and OAM and either or both of these properties can be used to represent quantum information. The SAM is due to the rotation of the electric field of light as it propagates which results in the polarisation of light, which is the direction of the electric field amplitude as the electromagnetic wave propagates. The polarisation of light is described completely within a two dimensional Hilbert space and is utilised in linear optical quantum computing (LOQC), which we have seen in Chapters 6.

OAM depends on spatial distribution of the electric field that arises due to the direction of the energy flow around the beam axis [137]. The light field of this form are usually described in the cylindrical coordinate system:

$$E(r, \phi, z) = E_0(r, z)e^{il\phi} \tag{8.5}$$

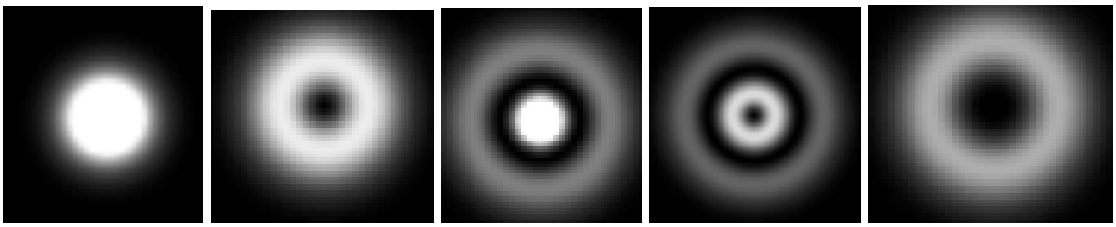
where the OAM is characterised by an azimuthal phase term $e^{il\phi}$. The index l is referred to as the azimuthal index. Unlike SAM (which has two unique modes of rotation), the azimuthal index, l is unbound and can have any value. These values which are the OAM states of light constitute an infinite-dimensional Hilbert space with orthonormal basis states $|l\rangle$, carrying an OAM of $l\hbar$ per photon,

$$|\psi\rangle = \sum_{l=-\infty}^{\infty} a_l |l\rangle$$

Restricting to a finite number of basis states then leads to the implementation of qudits, which carry quantum information in a finite d-dimensional basis.

Examples of light modes which carry OAM are the Laguerre-Gaussian (LG) modes [5]. These modes are light field that has helical wavefronts where the direction of energy flow rotates around the beam axis upon propagation and some of the modes are shown in Figure. 8.5. The field amplitude of such a mode is given by [6]

$$\begin{aligned} LG_p^l(r, \phi, z) = C_{l,p}^{LG} \left(\tfrac{r\sqrt{2}}{w(z)}\right)^{|l|} L_p^{|l|}\left(\tfrac{2r^2}{w^2(z)}\right) exp\left(-\tfrac{r^2}{w^2(z)}\right) exp\left(-\tfrac{ik^2r^2z}{2(z^2+z_R^2)}\right) \\ \times exp(il\phi) exp\left(i(2p+|l|+1)tan^{-1}\left(\tfrac{z}{z_R}\right)\right) \end{aligned} \tag{8.6}$$



LG_0^0 LG_0^1 LG_1^0 LG_1^1 LG_0^2

FIGURE 8.5: Laguerre-Gaussian (LG) modes

where k is the wave-vector magnitude of the field, z_R is the Rayleigh range, $w(z)$ is the radius of the beam at a position z, l is the azimuthal index number and p is the radial index number. $L_p^{|l|}$ is the associated Laguerre polynomial. The indices l and p provide information about the OAM of the beam. It can be shown that each photon in such a beam carries an OAM of $l\hbar$.

In order to provide a platform to describe in CQP the optical experiments with respect to OAM, it is essential to understand the quantum operators of OAM. We present the theory of OAM operators and the theory of manipulation of OAM using a blazed phase grating. The role of OAM operators is important as it leads to the understanding of the diffractive optical elements used in the experiments. This is similar to the approach in LOQC that we have demonstrated in Chapters 6 and 7. Although this is a task that is not yet achieved, we provide an initial attempt to describe the diffractive optical elements in OAM operators in this chapter.

8.4.1 Generation of orbital angular momentum

As it can be seen from Eq. (8.6) that for different values of l gives rise to LG modes of different helical phase structures. The radial distribution of the mode also depends on the index l. Importantly, one LG mode cannot be converted to another by any means but rather the technique used is to generate optical beams carrying OAM, by sending a fundamental Gaussian beam (LG_0^0) through a diffractive optical element. Two of the most commonly used diffractive optical elements for the generation of OAM are the computer-generated phase hologram and the spiral phase plate. Heckenberg *et. al.* [87] demonstrated that OAM beams of any desired order can be generated through diffraction by the incidence of a Gaussian beam on a l-forked hologram. The beam diffracted in the first-order is mostly considered and has a helical phase front that is described by $exp(il\phi)$. When operated in reverse, the diffraction holograms help to detect the OAM of a laser beam.

Blazed phase grating

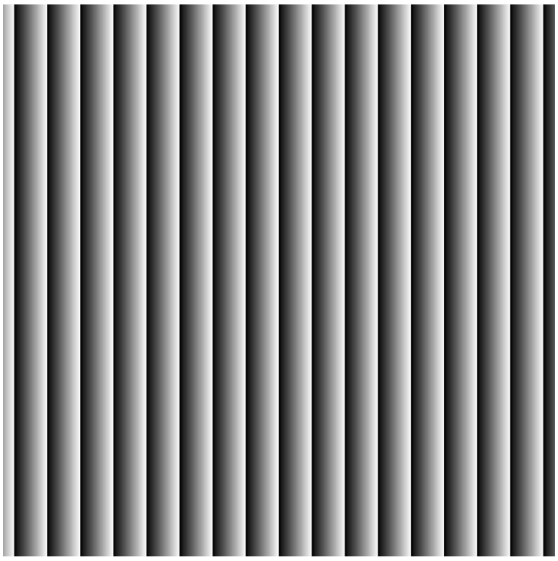


FIGURE 8.6: Blazed grating

Generally, any optical element such as blazed gratings, apertures, prisms and lenses are described by a transmission function [86]. The transmittance function t describes how the optical element changes the amplitude and phase of the propagating wave through the component. For a general case, the transmittance function is defined as:

$$t(x, y) = A(x, y)e^{i\Theta(x,y)} \tag{8.7}$$

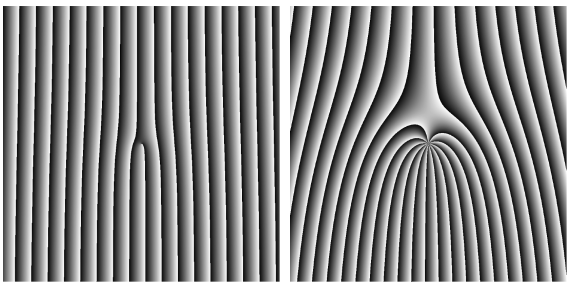
where $A(x, y)$ is the amplitude function and $\Theta(x, y)$ is the phase function of the element.

A one dimensional blazed phase grating is shown in Figure. 8.6. The dark and light shading indicates the variation of phase shifts. This is the simplest diffractive optical element (DOE) that is defined by the transmission function $t(x) = e^{i.k.\Theta(x)}$. Here $\Theta(x) = |\Phi_0.x|_{2\pi}$ is a modulo 2π operation on the phase function of an ideal linear blazed grating. The grating is referred to as blazed as it directs a large fraction of the incident light into one of the diffracted grating orders. The phase grating is assumed to transmit light with no attenuation but imparts a phase variation across the wavefront.

In order to generate a fork hologram, the phase shift $(l.\phi)$ is added to the regular blazed phase grating. The overall phase shift is given as:

$$\Theta(x, y) = \left| l.\phi + \frac{2\pi.x}{\Gamma} \right|_{2\pi} \tag{8.8}$$

where Γ is the grating constant and we consider the variation of Θ along the x direction. The fork holograms for l equals 1 and 11 is shown in Figure. 8.7. In cylindrical



(a) (b)

FIGURE 8.7: Fork Hologram

coordinates, we can rewrite the above equation as:

$$\Theta(r,\phi) = \left| l.\phi + \frac{2\pi.r\cos\phi}{\Gamma} \right|_{2\pi} \tag{8.9}$$

8.4.2 Orbital angular momentum in quantum mechanics

Using the classical mechanics approach, Allen *et. al.* [6] demonstrated that Lagurre-Gaussian modes carry a well defined OAM. The quantum mechanical approach shows that the LG modes are eigenvectors of the OAM operator $\hat{L_z}$ [148]. For a single-photon state, $|\psi\rangle$, say an LG-mode, the x, y, z components of the OAM operators in cartesian coordinates (x, y, z) are given by:

$$\begin{aligned} \hat{L_x} &= -i\hbar\left(\hat{y}\frac{\partial}{\partial\hat{z}} - \hat{z}\frac{\partial}{\partial\hat{y}}\right) \\ \hat{L_y} &= -i\hbar\left(\hat{z}\frac{\partial}{\partial\hat{x}} - \hat{x}\frac{\partial}{\partial\hat{z}}\right) \\ \hat{L_z} &= -i\hbar\left(\hat{x}\frac{\partial}{\partial\hat{y}} - \hat{y}\frac{\partial}{\partial\hat{x}}\right) \end{aligned} \tag{8.10}$$

Transforming the above operators represented in Eq. (8.10) to cylindrical coordinates (r, ϕ, z) to get:

$$\begin{aligned} \hat{L_x} &= i\hbar\cos\phi\frac{\partial}{\partial\phi} \\ \hat{L_y} &= i\hbar\sin\phi\frac{\partial}{\partial\phi} \\ \hat{L_z} &= -i\hbar\frac{\partial}{\partial\phi} \end{aligned} \tag{8.11}$$

Since, the wave is assumed to be propagated along the z direction, we consider the eigen-states of the OAM operator $\hat{L_z}$, with eigenvalues $l\hbar$. That is the eighenvalue equation

is given by $\hat{L_z}\Psi = l\hbar\Psi$ which leads to :

$$\Psi = A.exp(il\phi) \tag{8.12}$$

From Eq. (8.12), it is understood that any mode with a phase factor can be considered as a eigenvector of $\hat{L_z}$ and therefore has a well defined OAM. A photon is represented by a single LG mode that is in a quantum state having a definite value of OAM. It is essential to note that the eigenvalues of $\hat{L_z}$ can assume all integer values both positive or negative and this makes the possibility of having an infinite-dimensional Hilbert space. Another operator $\hat{L^2}$ is given by the sum of $\hat{L_x^2}$, $\hat{L_y^2}$ and $\hat{L_z^2}$, which is:

$$\hat{L^2} = -2\hbar^2 \frac{\partial^2}{\partial\phi^2}$$

The combination of $\hat{L_x}$ and $\hat{L_y}$ gives rise to the *ladder* operators $\hat{L^+}$ and $\hat{L^-}$. The ladder operators are defined by:

$$\hat{L^+} = i\hbar exp(i\phi)\frac{\partial}{\partial\phi} \text{ and } \hat{L^-} = i\hbar exp(-i\phi)\frac{\partial}{\partial\phi} \tag{8.13}$$

The commutation relations between these operators are provided as, $[\hat{L_x}, \hat{L_y}] = i\hbar\hat{L_z}$, $[\hat{L_y}, \hat{L_z}] = i\hbar\hat{L_x}$, $[\hat{L_z}, \hat{L_x}] = i\hbar\hat{L_y}$, $[\hat{L^+}, \hat{L^-}] = 2\hbar\hat{L_z}$, $[\hat{L_z}, \hat{L^+}] = \hbar\hat{L^+}$, $[\hat{L_z}, \hat{L^-}] = -\hbar\hat{L^-}$and $[\hat{X}, \hat{L^2}] = 0$ where $\{\hat{L_x}, \hat{L_y}, \hat{L_z}, \hat{L^+}, \hat{L^-}\} \in X$.

Manipulation of OAM

The Huygens-Fresnel integral is an equation which describes the diffraction due to the diffractive element based on certain assumptions. It is assumed that the source of light is at infinite distance and therefore the aperture is illuminated by a plane wave travelling along the z-axis. Another approximation is that the diffraction makes only small perturbations which is called the paraxial approximation. Based on these approximations, the Huygens-Fresnel integral is formulated as a $2D$ Fourier transform in polar coordinates given by [85]

$$E_p = \frac{i.A}{\lambda R_0}.e^{-ik.R_0}\int_0^b\int_0^{2\pi} t(r,\phi)exp\left[\frac{-i.k.r.\rho}{R_0}\cos(\theta-\phi)\right]r.dr.d\phi \tag{8.14}$$

where E_p is the diffraction field, A is the amplitude of plane wave illuminating the aperture, R_0 is the distance between the observation point and aperture plane. $t(r,\phi)$ is the complex transmission function given by $e^{i.\Theta(r,\phi)}$.

We use Eq. (8.8) in Eq. (8.14) to get:

$$E_p = \frac{i.A}{\lambda R_0}.e^{-ik.R_0}\int_0^{\infty}\int_0^{2\pi} exp\left[i\left(l.\phi + \frac{2\pi.r\cos\phi}{\Gamma}\right)\right] exp\left[\frac{-i.k.r.\rho}{R_0}\cos(\theta-\phi)\right] r.dr.d\phi \tag{8.15}$$

To simplify the equation, we say, $G = \frac{2\pi.r}{\Gamma}$ and $C = \frac{i.A}{\lambda R_0}.e^{-ik.R_0}$ and with a small manipulation, we get:

$$E_p = C.exp(i.l.\theta)\int_0^{\infty} r.dr\int_0^{2\pi} exp(i.l.(\phi-\theta))\times exp(i.G.r\cos\phi)\times exp\left[\frac{-i.k.r.\rho}{R_0}\cos(\theta-\phi)\right] d\phi \tag{8.16}$$

Now, we will try to solve the integral $\int d\phi$ in Eq. (8.16). Let,

$$I = \int_0^{2\pi} exp(i.l.(\phi-\theta)) \times exp(i.G.r\cos\phi) \times exp\left[\frac{-i.k.r.\rho}{R_0}\cos(\theta-\phi)\right] d\phi \tag{8.17}$$

Assuming the integral I is $\int_0^{2\pi} u.dv$ where $\int_0^{2\pi} u.dv = [uv]_0^{2\pi} - v\int_0^{2\pi} du$. Then, if $u = exp(i.G.r\cos\phi)$ and $dv = \int_0^{2\pi} exp(i.l.(\phi-\theta)) \times exp\left[\frac{-i.k.r.\rho}{R_0}\cos(\theta-\phi)\right] d\phi$. The term $[uv]_0^{2\pi}$ vanishes on substitution of u and v. We get

$$I = i.G.r \times \frac{J_l(\frac{k.r.\rho}{R_0})}{(\frac{k.\rho}{R_0})}\int_0^{2\pi} \sin\phi \times exp(i.G.r\cos\phi)d\phi \tag{8.18}$$

where $J_l(\frac{k.r.\rho}{R_0})$ is defined as the Bessel function. We can solve the integral in Eq. (8.18) to be:

$$\int_0^{2\pi} \sin\phi \times exp(i.G.r\cos\phi)d\phi = \frac{2.\pi J_0'(Gr)}{G} \tag{8.19}$$

Using Eq. (8.19) and Eq. (8.18) in Eq. (8.16) to get:

$$E_p = -\frac{2.\pi.C.exp(i.l.\theta)}{(\frac{k.\rho}{R_0})}\int_0^{\infty} r^2.J_l(\frac{k.r.\rho}{R_0}).J_0'(Gr)dr \tag{8.20}$$

Using the Bessel's differential identity that $[x^{-a}J_a(x)]' = -x^{-a}J_{a+1}(x)$, we get $J_0'(Gr) = -J_1(Gr)$ and using it in the above equation to give:

$$E_p = \frac{2.\pi.C.exp(i.l.\theta)}{(\frac{k.\rho}{R_0})}\int_0^{\infty} r^2.J_l(\frac{k.r.\rho}{R_0}).J_1(Gr)dr \tag{8.21}$$

8.5 Discussion

In this section, we consider the motivation of extending quantum process calculus to describe higher dimensional protocols.

Our aim in extending the semantics of CQP is to use the theories and methodologies of quantum process calculus to model optical experiments that exhibits higher dimensionality involving OAM of photon. Experiments have shown that photon pair entangled in their OAM up to a higher quantum number, can be produced with high-fidelity [68, 94]. In relation to quantum computation and communication, the higher dimensional Hilbert space of orbital angular momentum allows the implementation of new quantum protocols, which can offer higher information capacity and greater degree of security [69].

Recent studies have adopted the higher dimensionality encoded in the polarisation and orbital angular momentum for quantum information and cryptographic processing [45]. Boyd *et. al* [31] describe a method to construct a free-space quantum key distribution system that can carry many bits of information per photon, based on the use of LG modes and other field modes that carry OAM.

There has been a significant interest in the use of higher-dimensional systems for quantum information processing and cryptography mainly due to the large state space, which offers the higher rate of data transmission and increased security of cryptographic systems. This provided the motivation to extend the theory of quantum process calculus to model higher dimensional quantum systems.

We have presented only the transition rules that have been extended. Few rules such as R-Measure and L-Qdit are extended and the other rules are the same which demonstrates the compatibility of the general framework of the language. Using the theory of quantum operators in higher dimensions and with the help of the extended semantics, we show that we can model higher dimensional protocols namely qudit teleportation and superdense coding in CQP. We have seen that quantum process calculus provides a systematic methodology for verification of quantum systems. The theory of behavioural equivalence of CQP [51] is defined with respect to qubits and is extended to describe LOQC which is one of the main works of the present thesis. A future work in this regard is to extend the theory of equivalence to qudits, which is believed to be a straightforward task.

We present the theory of OAM operators and the theory of manipulation of OAM using a blazed phase grating. The role of OAM operators is important as it leads to the understanding of the diffractive optical elements used in the experiments. Further work needs to be done in order to describe the diffractive optical elements such as the blazed phase grating and other elements in terms of OAM operators, an approach which is similar to that we have seen in the previous chapters 6 and 7 for linear optical quantum computing. This would then aid us to formally define the diffractive optical elements using CQP and lead to use the mathematical tools of quantum process calculus CQP to model and analyse the quantum optical experiments involving OAM.

Chapter 9

Conclusion

The thesis describes the theory and applications of quantum process calculus, CQP. We have analysed quantum error code correction system, QECC, in CQP and verified the protocol by proving it equivalent to its specification. In addition to the existing axioms that are defined in [51], we have defined a few axioms in this thesis that helps us to reason quantum protocols namely superdense coding, quantum secret sharing, remote CNOT and quantum error code correction.

We have extended CQP to describe and verify the experimental processes associated with linear optical quantum computing (LOQC). In addition, we have extended CQP to model higher dimensional quantum systems. In this chapter, we summarise and discuss the work presented, and outline directions for future work.

9.1 Summary

The following is a detailed summary of the work that are presented in the thesis.

Chapter 1. The first chapter discussed the emergence and significance of the quantum information and quantum computation discipline. We reviewed the characteristics of quantum systems and discussed the key factors that highlighted their peculiarities such as entanglement and non-determinism. Some key results of this field were listed and the motivations for formal analysis of quantum systems were presented.

Chapter 2. A brief overview of formal methods was provided in this chapter and we discussed the recent work in the development of formal modelling and analysis of quantum systems. The chapter provides a short survey of the related work, which includes the

discussion of quantum programming languages, automated verification of quantum systems, semantic techniques for the analysis of quantum systems and quantum computing using linear optics.

Chapter 3. This chapter presented the theoretical background of relevance to this work, including the essential concepts of quantum computation and an introduction to process calculus.

Chapter 4. This chapter reviewed the operational semantics of CQP based on labelled transition systems (LTS). The LTS and its interpretation are essential in order to define the equivalence between processes. We describe the theory of behavioural equivalence provided in [51] and demonstrated that two models of a quantum error correcting code are each congruent to their respective high-level specification processes.

Chapter 5. This chapter focused on the axiomatic approach that is discussed in detail in [51]. By defining the additional axioms, we show that we can reason several other quantum protocols like superdense coding, quantum secret sharing, remote CNOT and quantum error correction code. The new axioms that are introduced are proved to be sound. Finally, the chapter discussed the significance of the role of these axioms in reasoning these protocols.

Chapter 6. The chapter presented an investigation into extending CQP to model linear optical quantum computing. In all previous work on quantum process calculus, qubit was considered as an information encoded within a 2 dimensional Hilbert space describing the internal states of a localised particle. We presented the extension of CQP by allowing multiple particles as information, described by Fock states. We described a physical realisation of quantum computing by defining the linear optical elements in CQP, and have demonstrated a model of an LOQC CNOT gate. Using our model, we have also described post-selection in CQP.

Chapter 7. This chapter provided the extension of theory of equivalence of CQP to verify LOQC. We have addressed the issues concerning the semantics that was discussed in the previous chapter. We described and analysed two models of the linear optical experimental system that demonstrates a CNOT gate. The two models used different measurement semantics in order to work at different levels of abstraction. This demonstrates the flexibility of process calculus to support a range of descriptions.

Chapter 8. This chapter presented the extensions of CQP to model higher dimensional quantum processes. We presented the semantics that are modified to describe the higher dimensional quantum systems. Using the extended semantics, we modelled higher dimensional quantum protocols namely qudit teleportation and superdense coding. The prime motivation of this work was mainly due to optical experiments that exhibit higher

dimensionality using the intrinsic property of light, i.e. OAM of a photon. We presented a brief study on OAM operators and the theory of manipulation of OAM by the diffractive optical element (e.g. blazed phase grating).

9.2 Concluding Remarks

The main focus of this thesis is to further develop the theory of quantum process calculus. In doing so, we extend the applications of the formal techniques from describing abstract models to that of experimental systems associated with quantum information processing.

The previous work on the quantum process calculus, Communicating Quantum Processes (CQP), has provided the foundation for much of this work. The flexibility of the language to adapt to different situations has led to the achievement of these tasks. In Chapters 4 and 5, we reviewed the previous work of CQP. The work defined the theory of behavioural equivalence in CQP and applied the theory to teleportation and superdense coding. We employ the theory (also described in Chapter 4) and extend the application of it by verifying quantum error code correction.

The existing equational theory of CQP based upon the full probabilistic branching bisimilarity provided the motivation of the work presented in Chapter 5. The previous work illustrated the theory in the reasoning of quantum teleportation. We define a few additional rules in Chapter 5 and illustrate its significance by improving the ability to reason equationally. We take a step further in reasoning other protocols namely superdense coding, quantum secret sharing, remote CNOT and quantum error correction code.

The motivation for developing formal methods to quantum systems is to provide an understanding of concurrent, communicating quantum systems, and to use the tools for verifying the correctness of crypto-systems. Using higher-dimensional quantum systems for applications in quantum information and cryptography are becoming of significant interest as it improves the data transmission rate and security of cryptographic interest. Chapter 6 provides the extensions of CQP to model higher dimensional quantum systems, in particular, focussing on the representation and manipulation of the quantum state. The protocols quantum teleportation and superdense coding for higher dimensional quantum systems are studied, as these constitute the building blocks of large and complex systems. In the later part of the work, we attempt to study the optical experimental systems that demonstrate this property of higher-dimensionality.

Optical implementations offer to date the most advanced system for quantum information processing. LOQC is one potential way of implementing small-scale quantum

computing. Chapter 7 provides a deep understanding of the application of quantum process calculus to LOQC. This understanding set out the foundations of Chapter 8, in which different measurement semantics are presented. Another important development is the extension of the theory of equivalence to verify LOQC. This helps to understand and verify a physical realisation of quantum computing.

Another quantum process calculus, qCCS, developed by Feng *et. al.* [66] is similar to our previous work that considers qubit as an abstract information that can be sent or received through channels. qCCS is a quantum extension of the classical value-passing, CCS [124], and proved that the weak bisimilarity is a congruence. Their result is applied to protocols: teleportation, superdense coding and quantum key distribution [108].

9.3 Future Work

In the final section, we provide several directions for further work based around the framework of CQP presented in this thesis.

The extensions of CQP to describe the optical experimental systems that demonstrate the higher-dimensionality is a study which is to be investigated. The study would help us to model the experiments and would also provide an understanding on the decoherence or noise that is involved in the experiments, which is an important aspect of quantum communication devices. Extending the theory of equivalence for higher-dimensional quantum systems is believed to be a straightforward task but needs to be verified.

Ying *et. al.* [169, 170] demonstrated the theory of approximate bisimulation based on strong bisimilarity. It would be interesting to implement this concept of approximate equivalence in CQP as it would provide an understanding on quantum noise that could occur in physical implementations of quantum systems.

Also of interest would be an analysis of integrate waveguide circuit demonstrating Shor's algorithm operating on four qubits [142]. The linear optical circuit uses the basic elements that is defined in this thesis. This helps to formally analyse quantum algorithms in CQP using LOQC and may provide another platform to learn about quantum complexity in LOQC using formal techniques. Another potential task would be to extend the equational theory of CQP to be applied in the setting of LOQC.

The long-term goal is to develop software for automated analysis of CQP models, following the established work in classical process calculus and recent work on automated equivalence checking of concurrent quantum programs [12]. The equivalence checking tool uses the stabilizer formalism for the verification of quantum protocols. It would be

interesting in the possibility of extending this simulation to universal quantum computation. Although this task is not possible to do in an efficient way but in [1] it is shown that stabilizer circuits can be extended to include a limited number of non-clifford gates by not reducing the gain efficiency.

As mentioned earlier, quantum process calculus provides a systematic methodology for verification of quantum systems. This is an important factor as we believe that quantum cryptographic applications will drive the market, and formal methods provide a useful way in analysing the behaviour of these implemented systems. There are many other interesting directions for further study and it is hoped that the present work has provided a good indication for future progress in quantum process calculus.

Bibliography

[1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[2] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science.*, pages 415–425. IEEE, 2004.

[3] S. Abramsky and B. Coecke. Abstract physical traces. *THEORY AND APPLICATIONS OF CATEGORIES*, 14(6):111–124, 2005.

[4] P. Adao and P. Mateus. A process algebra for reasoning about quantum security. *Electronic Notes in Theoretical Computer Science*, 170:3–21, 2007.

[5] L. Allen, S. M. Barnett, and M. J. Padgett. *Optical Angular Momentum*. Optics & Optoelectronics. Taylor & Francis, 2003.

[6] L. Allen, M. W. Beijersbergen, R. J. C. Spreeuw, and J. P. Woerdman. Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes. *Phys. Rev. A*, 45:8185–8189, 1992.

[7] T. Altenkirch and J. Grattage. A functional quantum programming language. In *Proceedings. 20th Annual IEEE Symposium on Logic in Computer Science, 2005.*, pages 249–258. IEEE, 2005.

[8] T. Altenkirch and J. Grattage. Qml: Quantum data and control. 2005.

[9] T. Altenkirch, J. Grattage, J. K. Vizzotto, and A. Sabry. An algebra of pure quantum programming. *Electronic Notes in Theoretical Computer Science*, 170:23–47, 2007.

[10] S. Andova and T. A. C. Willemse. Branching bisimulation for probabilistic systems: Characteristics and decidability. *Theoretical Computer Science*, 356(3):325 – 355, 2006.

[11] E. Ardeshir-Larijani, S. J. Gay, and R. Nagarajan. Equivalence checking of quantum protocols. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *Lecture Notes in Computer Science*, pages 478–492. Springer Berlin Heidelberg, 2013.

[12] E. Ardeshir-Larijani, S. J. Gay, and R. Nagarajan. Verification of concurrent quantum protocols by equivalence checking. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 8413 of *Lecture Notes in Computer Science*, pages 500–514. Springer Berlin Heidelberg, 2014.

[13] B. Atelier. http://www.atelierb.eu/en/.

[14] P. Baltazar, R. Chadha, and P. Mateus. Quantum computation tree logic model checking and complete calculus. *International Journal of Quantum Information*, 6(2):219–236, 2008.

[15] J. Barnes. *High integrity software: the SPARK approach to safety and security.* Addison-Wesley Longman Publishing Co., Inc., 2003.

[16] S. M. Barnett. *Quantum Information.* Oxford University Press, 2009.

[17] S. D. Bartlett, de Guise D, and B. C. Sanders. Quantum encodings in spin systems and harmonic oscillators. *Physical Review Letters A*, 65, 2002.

[18] F. Belardinelli, P. Gonzalez, and A. Lomuscio. Automated verification of quantum protocols using MCMAS. *EPTCS*, 85:48–62, 2012.

[19] F. J. Belinfante. On the current and the density of the electric charge, the energy, the linear momentum and the angular momentum of arbitrary fields. *Physica*, 7(5):449 – 474, 1940.

[20] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.

[21] C. H. Bennett. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters*, 68:3121, 1992.

[22] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *Journal of Cryptology*, 5(1):3–28, 1992.

[23] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179. IEEE, 1984.

[24] 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(13):1895–1899, 1993.

[25] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Phys. Rev. Lett.*, 69:2881–2884, 1992.

[26] J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109 – 137, 1984.

[27] G. Berlin, G. Brassard, F. Brussieres, and N. Godbout. Loss-tolerant quantum coin flipping. *In Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008)*, 3121:1–9, 2008.

[28] R. A. Beth. Mechanical detection and measurement of the angular momentum of light. *Physical Review*, 50(2):115, 1936.

[29] R. F. Blute, I. T. Ivanov, and P. Panangaden. Discrete quantum causal dynamics. *International Journal of Theoretical Physics*, 42(9):2025–2041, 2003.

[30] D. Bohm. *Quantum Theory*. Prentice Hall, 1951.

[31] R. W. Boyd, A. Jha, M. Malik, C. O'Sullivan, B. Rodenburg, and D. J. Gauthier. Quantum key distribution in a high-dimensional state space: exploiting the transverse degree of freedom of the photon. In *SPIE OPTO*, pages 79480L–79480L. International Society for Optics and Photonics, 2011.

[32] G. Brassard and C. Crepeau. Quantum bit commitment and coin tossing protocols. *Advances in Cryptology-CRYPT0 90*, 537:49–61, 1991.

[33] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145, Oct 2001.

[34] L. Cardelli and A. D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures*, pages 140–155. Springer, 1998.

[35] J. Carolan, P. Shadbolt, J. D. A. Meinecke, N. J. Russell, N. Ismail, K. Wörhoff, T. Rudolph, M. G. Thompson, J. L. O'Brien, J. C. F. Matthews, and A. Laing. On the experimental verification of quantum complexity in linear optics. *Nature photonics*, 8:621–626, 2014.

[36] N. J. Cerf, C. Adami, and P. G. Kwiat. Optical simulation of quantum logic. *Physical Review Letters A*, 57:R1477, 1998.

[37] J. Clausen, L. Knll, and D. G. Welsch. Entanglement purification of multimode quantum states. *Journal of Optics B: Quantum and Semiclassical Optics*, 5(6):S561, 2003.

[38] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, 1969.

[39] R. Cleaveland and S. Sims. Concurrency Wworkbench of the New Centry (CWB-NC). 2009.

[40] B. Coecke and R. Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

[41] C. Cohen-Tannoudji, B. Diu, and F. Laloë. *Quantum Mechanics*. Vol 1. Wiley-Interscience, 1977.

[42] A. Crespi, R. Ramponi, R. Osellame, L. Sansoni, I. Bongioanni, F. Sciarrino, G. Vallone, and P. Mataloni. Integrated photonic quantum gates for polarization qubits. *Nature communications*, 2:566, 2011.

[43] D-Wave. http://www.dwavesys.com.

[44] A. C. Dada, J. Leach, G. S. Buller, M. J. Padgett, and E. Andersson. Experimental high-dimensional two-photon entanglement and violations of generalised Bell inequalities. *Nature Physics*, 7:677–680, 2011.

[45] V. D'Ambrosio, E. Nagali, L. Marrucci, and F. Sciarrino. Orbital angular momentum for quantum information processing. *Proceedings of the SPIE*, 8440, 2012.

[46] V. Danos, E. D'Hondt, E. Kashefi, and P. Panangaden. Distributed measurement-based quantum computation. *Electronic Notes in Theoretical Computer Science*, 170:73–94, 2007.

[47] V. Danos and E. Kashefi. Determinism in the one-way model. *Physical Review A*, 74(5):052310, 2006.

[48] V. Danos and E. Kashefi. Pauli measurements are universal. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL)*, volume 170, pages 95 – 100, 2007.

[49] G. M. D'Ariano, C. Macchiavello, and L. Maccone. Quantum computations with polarized photons. *Fortschritte der Physik*, 48(5-7):573–577, 2000.

[50] A. Datta, R. Küsters, J. C. Mitchell, A. Ramanathan, and V. Shmatikov. Unifying equivalence-based definitions of protocol security, 2004.

[51] T. A. S. Davidson. *Formal Verification Techniques using Quantum Process Calculus*. PhD thesis, University of Warwick, 2011.

[52] T. A. S. Davidson, S. J. Gay, R. Nagarajan, and I. V. Puthoor. Analysis of a quantum error correcting code using quantum process calculus. In *Proceedings of the International Workshop on QPL*, volume 95, pages 67–80. EPTCS, 2011.

[53] E. Desurvire. *Classical and Quantum Information Theory: An Introduction for the Telecom Scientist*. Cambridge University Press, 2009.

[54] D. Deutsch. Quantum theory, the Church-Truing principle and the universal quantum computer. In *Proceedings of the Royal Society of London Ser. A*, pages 97–117, 1985.

[55] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London Ser. A*, pages 553–558, 1992.

[56] S. J. Devitt, K. Nemoto, and W. J. Munro. Quantum error correction for beginners. *arXiv preprint arXiv:0905.2794*, 2009.

[57] E. D'hondt and P. Panangaden. Quantum weakest preconditions. *Mathematical Structures in Computer Science*, 16(03):429–451, 2006.

[58] L. Dixon and A. Kissinger. Open-graphs and monoidal theories. *Mathematical Structures in Computer Science*, 23(02):308–359, 2013.

[59] R. Duncan. Believe it or not, bell states are a model of multiplicative linear logic. Technical report, 2004.

[60] P. V. Eijk and M. Diaz, editors. *Formal Description Technique Lotos: Results of the Esprit Sedos Project*. Elsevier Science Inc., 1989.

[61] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, 1935.

[62] A. K. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67:661, 1991.

[63] M. Elboukhari, M. Azizi, and A. Azizi. Analysis of Quantum Cryptography Protocols by Model Checking. *International Journal of Universal Computer Science*, 1(1), 2010.

[64] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schalfer, and H. Yeh. Current status of the DARPA quantum network. *arXiv: quant-ph/0503058v2*, 2005.

[65] Y. Feng, R. Duan, Z. Ji, and M. Ying. Probabilistic bisimilarities between quantum processes. arXiv:cs.LO/0601014, 2006.

[66] Y. Feng, R. Duan, and M. Ying. Bisimulation for quantum processes. In *Proceedings of the 38th Annual ACM Symposium on Principles of Programming Languages*, pages 523–534. ACM, 2011.

[67] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1980.

[68] R. Fickler, M. Krenn, R. Lapkiewicz, S. Ramelow, and A. Zeilinger. Real-time imaging of quantum entanglement. *Scientific reports*, 3, 2013.

[69] S. Franke-Arnold, L. Allen, and M. J. Padgett. Advances in optical angular momentum. *Laser and Photonics Reviews*, 2(4):299–313, 2008.

[70] S. Franke-Arnold, S. J. Gay, and I. V. Puthoor. Quantum process calculus for linear optical quantum computing. In *Proceedings of the 5th Conference on Reversible Computation (RC)*, volume 7948, pages 234–246. LNCS, 2013.

[71] S. Franke-Arnold, S. J. Gay, and I. V. Puthoor. Verification of linear optical quantum computing using quantum process calculus. In *Proceedings of the Combined International Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS)*, volume 160, pages 111–129. EPTCS, 2014.

[72] K. Fujii. Generalized bell states and quantum teleportation. *arXiv: quant-ph/0106018*, 2001.

[73] S. J. Gay. Quantum programming languages: Survey and bibliography. *Mathematical Structures in Computer Science*, 16(04):581–600, 2006.

[74] S. J. Gay and R. Nagarajan. Communicating Quantum Processes. In *Proceedings of the 32nd Annual ACM Symposium on Principles of Programming Languages*, pages 145–157. ACM, 2005.

[75] S. J. Gay and R. Nagarajan. Types and Typechecking for Communicating Quantum Processes. *Mathematical Structures in Computer Science*, 16(3):375–406, 2006.

[76] S. J. Gay, R. Nagarajan, and N. Papanikolaou. Probabilistic Model-Checking of quantum protocols. arxiv:quant-ph/0504007v2. 2005.

[77] S. J. Gay, N. Papanikolaou, and R. Nagarajan. QMC: a model checker for quantum systems. In *CAV 2008: In Proceedings of the 20th International Conference on Computer Aided Verification*, volume LNCS of *Lecture Notes in Computer Science*, pages 543–547. Springer-Verlag, 2008.

[78] S. J. Gay and I. V. Puthoor. Application of Quantum Process Calculus to Higher Dimensional Quantum Protocols. In *Proceedings of the International Workshop on QPL*, volume 158, pages 15–28. EPTCS, 2014.

[79] D. Gottesman and I. L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402:390–393, 1982.

[80] D. Gottesman, A. Kitaev, and J. Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, 64:012310, 2001.

[81] A. S. Green, P. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron. An introduction to quantum programming in quipper. In *Reversible Computation*, volume 7948 of *Lecture Notes in Computer Science*, pages 110–124. Springer Berlin Heidelberg, 2013.

[82] D. M. Greenberger, M. A. Horne, and A. Zeilinger. Going beyond bells theorem. In *Bells theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[83] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219. ACM, 1996.

[84] J. Gruska. *Quantum Computing*. McGaw-Hill International, 1999.

[85] B. Guenther. *Modern Optics*. Wiley, 1990.

[86] E. Hecht. *Optics*. Addison-Wesley, 2002.

[87] N. R. Heckenberg, R. McDuff, C. P. Smith, H. Rubinsztein-Dunlop, and M. J. Wegener. Laser beams with phase singularities. *Optical and Quantum Electronics*, 24(9), 1992.

[88] M. Hillery, V. Bužek, and A. Berthiaume. Quantum secret sharing. *Phys. Rev. A*, 59:1829–1834, 1999.

[89] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[90] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[91] J. C. Howell and J. A. Yeazell. Linear optics simulations of the quantum baker's map. *Phys. Rev. A*, 61:012304, 1999.

[92] G. D. Hutchinson and G. J. Milburn. Nonlinear quantum optical computing via measurement. *Journal of Modern Optics*, 51(8):1211–1222, 2004.

[93] IDQ. http://www.idquantique.com/company/presentation.html.

[94] B. Jack, J. Leach, H. Ritsch, S. M. Barnett, M. J. Padgett, and S. Franke-Arnold. Precise quantum tomography of photon pairs with entangled orbital angular momentum. *New Journal of Physics*, 11, 2009.

[95] P. Jorrand and M. Lalire. Toward a quantum process algebra. In *CF '04: Proceedings of the 1st Conference on Computing Frontiers*, pages 111–119, New York, NY, USA, 2004. ACM Press.

[96] K. Kagalwala, G. Di Giuseppe, A. F. Abouraddy, and B. Saleh. Cnot gate with polarization and orbital angular momentum of single photons. In *Frontiers in Optics*, pages FTh1C–2. Optical Society of America, 2013.

[97] O. Kahramanogullari, F. Jordan, and C. Priami. Composability: Perspectives in ecological modeling. In *Algebraic and Numeric Biology*, volume 6479 of *Lecture Notes in Computer Science*, pages 136–148. Springer Berlin Heidelberg, 2012.

[98] R. Kaivola, R. Ghughal, N. Narasimhan, A. Telfer, J. Whittemore, S. Pandav, A. Slobodov, C. Taylor, V. Frolov, E. Reeber, and A. Naik. Replacing testing with formal verification in Intel Core TM i7 processor execution engine validation. In *Computer Aided Verification*, volume 5643 of *Lecture Notes in Computer Science*, pages 414–429. Springer Berlin Heidelberg, 2009.

[99] R. W. Keyes. Minitaurization of electronics and its limits. *IBM J. Res. Dev.*, 32(1):24–28, 1988.

[100] D. Kielpinski, C. Monroe, and D. J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417:709–711, 2002.

[101] A. Kissinger. *Pictures of Processes: Automated Graph Rewriting for Monoidal Categories and Applications to Quantum Computing.* PhD thesis, University of Oxford, 2012.

[102] A. Kissinger, A. Merry, and M. Soloviev. Pattern graph rewrite systems. arxiv:1204.6695. 2012.

[103] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. Te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit rsa modulus. In *Proceedings of the 30th Annual Conference on Advances in Cryptology*, CRYPTO'10, pages 333–350. Springer-Verlag, 2010.

[104] C. Kloeffel and D. Loss. Prospects for Spin-Based Quantum Computing in Quantum Dots. *Annual Review of Condensed Matter Physics*, 4(1):51–81, 2013.

[105] E. Knill. Conventions for quantum pseudocode. *Technical Report LAUR-96-2724*, 1996.

[106] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409:46, 2001.

[107] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.*, 79:135–174, 2007.

[108] T. Kubota, Y. Kakutani, G. Kato, Y. Kawano, and H. Sakurada. Application of a process calculus to security proofs of quantum protocols. In *Proceedings of WORLDCOMP/FCS2012*, 2012.

[109] M. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer Berlin Heidelberg, 2011.

[110] M. Lalire. Relations among quantum processes: bisimilarity and congruence. *Mathematical Structures in Computer Science*, 16(3):407–428, 2006.

[111] T. Lanting, A. J. Przybysz, A. Y. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, and G. Rose. Entanglement in a quantum annealing processor. *Phys. Rev. X*, 4:021041, 2014.

[112] K. G. Larsen and L. Xinxin. Compositionality through an operational semantics of contexts. *Journal of Logic and Computation*, 1(6):761–795, 1991.

[113] S. Lloyd and S. L. Braunstein. Quantum computation over continuous variables. 82(8):1784–1787, 1999.

[114] A. Lomuscio, H. Qu, and F. Raimondi. Mcmas: A model checker for the verification of multi-agent systems. In *Computer Aided Verification*, volume 5643 of *Lecture Notes in Computer Science*, pages 682–688. Springer Berlin Heidelberg, 2009.

[115] G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer Berlin Heidelberg, 1996.

[116] MagiQ. http://www.magiqtech.com/magiq/home.html.

[117] Y. I. Manin. Computable and uncomputable (in Russian). *Sovetskoye Radio*, 1980.

[118] D. Markham and B. C. Sanders. Graph states for quantum secret sharing. *Phys. Rev. A*, 78:042309, 2008.

[119] J. C. Maxwell. A dynamical theory of the electromagnetic field. *Proceedings of the Royal Society of London*, 13:531–536, 1863.

[120] D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.

[121] F. Miatto, S. M. Barnett, A. M. Yao, M. Padgett, B. Jack, and M. J. Romero. Investigating the entanglement structure of down-converted photon pairs. In *International Conference on Quantum Information*, page QTuF6. Optical Society of America, 2011.

[122] R. Miller, T. E. Northup, K. M. Birnbaum, A. Boca, A. D. Boozer, and H. J. Kimble. Trapped atoms in cavity QED: coupling quantized light and matter. *Journal of Physics B Atomic Molecular Physics*, 38:551, 2005.

[123] R. Milner. *A Calculus of Communicating Systems.* Springer-Verlag New York, Inc., 1982.

[124] R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.

[125] R. Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.

[126] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. *Information and Computation*, 100:1–40, 1992.

[127] G. Moore. Cramming more components on integrated circuits. *Electronics*, 38(8):82–85, 1965.

[128] C. R. Myers and R. Laflamme. Linear optics quantum computation: an overview. *arXiv preprint quant-ph/0512104*, 2005.

[129] R. Nagarajan and S. J. Gay. Formal verification of quantum protocols. arxiv:quant-ph/0203086v1. 2002.

[130] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-Cooper-pair box. *Nature*, 398:786–788, 1999.

[131] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, 1978.

[132] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[133] G. M. Nikolopoulos and G. Alber. Security bound of two-basis quantum-key-distribution protocols using qudits. *Physical Review Letters A*, 72, 2005.

[134] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning. Demonstration of an all-optical quantum controlled-not gate. *Nature*, 426:264, 2003.

[135] B. Ömer. A procedural formalism for quantum computing. Master's thesis, University of Vienna, 1998.

[136] B. Ömer. Quantum programming in qcl. Master's thesis, Technical University of Vienna, 2000.

[137] M. J. Padgett and L. Allen. The Poynting vector in Laguerre-Gaussian laser modes. *Optics communications*, 121(1):36–40, 1995.

[138] D. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer Berlin Heidelberg, 1981.

[139] S. Perdrix. Quantum patterns and types for entanglement and separability. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL)*, volume 170, pages 125 – 138, 2007.

[140] G. D. Plotkin. A structural approach to operational semantics. 1981.

[141] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O'Brien. Silica-on-silicon waveguide quantum circuits. *Science*, 320:646, 2008.

[142] A. Politi, J. C. F. Matthews, and J. L. O'Brien. Shor's quantum factoring algorithm on a photonic chip. *Science*, 325:1221, 2009.

[143] A. Poppe, A. Fedrizzi, R. Ursin, H. Böhm, T. Lörunser, O. Maurhardt, M. Peev, M. Suda, C. Kurtsiefer, H. Weinfurter, T. Jennewein, and A. Zeilinger. Practical quantum key distribution with polarisation entangled photons. *Optics Express*, 12(16):3865–3871, 2004.

[144] F. Prost and C. Zerrari. A logical analysis of entanglement and separability in quantum higher-order functions. arxiv:0801.0649. 2008.

[145] T. C. Ralph, A. Gilchrist, G. J. Milburn, W. J. Munro, and S. Glancy. Quantum computation with optical coherent states. *Phys. Rev. A*, 68:042319, 2003.

[146] T. C. Ralph, N. K. Langford, T. B. Bell, and A. G. White. Linear optical controlled-NOT gate in the coincidence basis. *Physical Review A*, 65(6):062324, 2002.

[147] A. W. Roscoe. *Model-checking CSP*. Prentice Hall, 1994.

[148] J. Sakurai. *Modern Quantum Mechanics*. Addison-Wasley, 1994.

[149] J. W. Sanders and P. Zuliani. Quantum programming. In *In Mathematics of Program Construction*, pages 80–99. Springer-Verlag, 1999.

[150] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[151] P. Selinger. A brief survey of quantum programming languages. In *Functional and Logic Programming*, pages 1–6. Springer, 2004.

[152] P. Selinger. Towards a quantum programming language. *Mathematical. Structures in Comp. Sci.*, 14(4):527–586, 2004.

[153] P. Selinger. Towards a semantics for higher-order quantum computation. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages, TUCS General Publication.*, volume 33, pages 127–143, 2004.

[154] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. IEEE Press, 1994.

[155] D. A. Sofge. A survey of quantum programming languages: History, methods, and tools. In *Quantum, Nano and Micro Technologies, 2008 Second International Conference on*, pages 66–71. IEEE, 2008.

[156] T. P. Spiller, W. J. Munro, S. D. Barrett, and P. Kok. An introduction to quantum information processing: applications and realizations. *Contemporary Physics*, 46(6):407–436, 2005.

[157] A. M. Steane. A tutorial on quantum error correction. In *PROCEEDINGS-INTERNATIONAL SCHOOL OF PHYSICS ENRICO FERMI*, volume 162, page 1, 2007.

[158] N. Trčka and S. Georgievska. Branching bisimulation congruence for probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 220(3):129 – 143, 2008.

[159] L. Vaidman. Teleportation of quantum states. *Phys. Rev. A*, 49:1473–1476, 1994.

[160] R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3):555–600, 1996.

[161] A. van Tonder. A lambda calculus for quantum computation. *SIAM Journal on Computing*, 33(5):1109–1135, 2004.

[162] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001.

[163] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

[164] C. P. Williams and S. H. Clearwater. *Ultimate Zero and One: Computing at the Quantum Frontier*. Springer, 2000.

[165] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald. Formal methods: Practice and experience. *ACM Comput. Surv.*, 41(4):19:1–19:36, 2009.

[166] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 1982.

[167] A. K. Wright and M. Felleisen. A syntactic approach to type soundness. *Information and Computation*, 115(1):38 – 94, 1994.

[168] T. Yamamoto, M. Koashi, Ş. K. Özdemir, and N. Imoto. Experimental extraction of an entangled photon pair from two identically decohered pairs. *Nature*, 421(6921):343–346, 2003.

[169] M. Ying, Y. Feng, and R. Duan. An algebra of quantum processes. http://arxiv.org/abs/0707.0330v1, Jul 2007.

[170] M. Ying, Y. Feng, R. Duan, and Z. Ji. An algebra of quantum processes. *ACM Transactions on Computational Logic*, 10(3):1–36, 2009.

[171] M. Ying, Y. Feng, and N. Yu. Quantum information-flow security: Noninterference and access control. In *Proceedings of the 26th IEEE Symposium on Computer Security Foundations (CSF)*, pages 130–144, 2013.

[172] X. Zhou, D. W. Leung, and I. L. Chuang. Methodology for quantum logic gate construction. *Physical Review A*, 62(5):052316, 2000.