APPLICATION OF FORMAL METHODS TO QUANTUM CRYPTOGRAPHY
（形式的手法の量子暗号への応用）

by

Takahiro Kubota
久保田 貴大

A Doctor Thesis
博士論文

Submitted to
the Graduate School of the University of Tokyo
on December 13, 2013
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Masami Hagiya　萩谷 昌己
Professor of Computer Science

ABSTRACT

In general, it is difficult to verify security of cryptographic protocols. Indeed, flaws of designs and security proofs of some protocols were found after they had been presented. In deductive verification using formal methods, protocols and security properties are described in formal languages, and correctness of designs and security proofs are deduced by inference rules. While a number of formal frameworks and verification tools have been developed and applied to classical protocols, few formal methods have been applied to security proofs of quantum protocols. The contributions of this thesis consist of the following three results.

First, we developed a software tool to verify bisimilarity of configurations (pairs of processes and quantum states) of qCCS, a quantum process calculus presented by Feng et al. Bisimilar configurations behave indistinguishably from the outside. We designed a formal framework for the verifier, which we call nondeterministic qCCS, on the basis of qCCS by Feng et al. While the transition system of qCCS is both nondeterministic and probabilistic, we presented a nondeterministic and non-probabilistic transition system for configurations, extending the definition of them. This allows the verifier to verify bisimilarity efficiently. Next, we designed the verifier to handle security parameters and quantum states symbolically. A purpose is to apply it to quantum cryptographic protocols, where the dimensions of quantum states depend on security parameters. When the verifier checks bisimulation relation of configurations, it uses user-defined equations on the symbolic representations to check the quantum states that the outsider can access are always equal. Besides, the verifier is sound with respect to qCCS, that is, when it runs with two configurations as input and returns *true*, a symbol representing success of the verification, the configurations are in bisimulation relation in the qCCS's definition.

Second, we defined a notion of approximate bisimulation relation on configurations of nondeterministic qCCS, and extended the verifier to check the approximate bisimilarity. Approximately bisimilar configurations reveal quantum states with close trace distance to the outsider after transitions. This property is useful in security proofs of quantum key distribution protocols. We then proved the approximate bisimulation relation is closed under application of evaluation contexts of processes.

Third, we formally verified Shor and Preskill's security proof of BB84 quantum key distribution protocol. They first considered another protocol (the EDP-based protocol) and proved the security of BB84 and the EDP-based protocol is equivalent. They next proved the latter is secure. For the first step of their proof, we formalized the two protocols as configurations and formally verified that they are bisimilar. For the second step, we defined a completely secure protocol (EDPideal) and formally verified that the configurations of it and the EDP-based protocol are approximately bisimilar. This is the first work where a security proof of a quantum cryptographic protocol is mechanically verified using a software tool.

論文要旨

暗号プロトコルの安全性の検証は一般に難しく, 実際, いくつかのプロトコルの設計や安全性証明の誤りが, 提案後に指摘されるということが起こってきた. 形式的手法を用いた導出的な検証では, プロトコルや安全性証明を形式言語で記述し, 設計や証明の正しさを推論規則にしたがって導出する. 古典暗号プロトコルに対しては, 検証のために多くの形式体系や検証ツールが開発され適用されているが, 量子暗号プロトコルの安全性証明に対して形式的手法はほとんど適用されていない. 本研究の貢献は以下の 3 種類の結果からなる.

一つ目の結果は, Feng らの量子プロセス計算 qCCS のコンフィグレーション (プロセスと量子状態の組) たちの双模倣性の検証器を実装したことである. 双模倣なコンフィグレーションたちは, 外から見て同じように振る舞う. まず我々は, 検証器のための形式体系である非決定的 qCCS を, Feng らの qCCS に基づいて設計した. qCCS の状態遷移系は確率的かつ非決定的であったが, 定義を拡張したコンフィグレーションに対する確率的でない状態遷移系を提案し, 採用した. このことにより, 検証器は双模倣性を効率よく検証する. 次に, 我々は検証器を, セキュリティパラメタや量子状態を記号的に扱うように設計した. 量子暗号プロトコルでは, 量子状態の次元がセキュリティパラメタに依存することがあるが, そのようなプロトコルに適用するためである. 双模倣関係の検証において, 検証器は, 外部者がアクセス可能な量子状態が常に等しいことを確認するために, 量子状態の記号表現に対するユーザ定義の等式を用いる. さらに, 検証器は qCCS に対して健全である. すなわち, 検証器にふたつのコンフィグレーションを入力したとき, 検証成功を表す $true$ を出力したならば, それらは qCCS の定義での双模倣関係にある.

二つ目の結果として, 我々は, 非決定的 qCCS のコンフィグレーションたちに対して近似双模倣関係を定義し, それを検証するよう検証器を拡張した. 近似双模倣関係にあるふたつのコンフィグレーションの遷移後には, 外部から見た量子状態のトレース距離が近いことが保証される. この性質は, 量子鍵配送プロトコルの安全性証明において有用である. さらに, 近似双模倣関係がプロセスの評価文脈の適用に対して閉じていることを示した.

三つ目の結果は, Shor と Preskill による BB84 量子鍵配送プロトコルの安全性証明を, 検証器を用いて形式的に検証したことである. 彼らの証明では, 第一に安全性が解析しやすい別のプロトコル (EDP に基づくプロトコル) が考えられ, BB84 と EDP に基づくプロトコルの安全性が等価であることが示された. 第二に, EDP に基づくプロトコルが安全であることが示された. 我々は, 第一のステップに対しては, この二つのプロトコルを形式化したコンフィグレーションが双模倣であることを検証した. 第二のステップに対しては, まず, 完全に安全な鍵配送プロトコル (EDPideal) を定義し, それと EDP に基づいたプロトコルのコンフィグレーションが近似双模倣であることを検証した. 量子鍵配送プロトコルの数学的な安全性を, ソフトウェアを用いて機械的に検証したのは本研究が初めてである.

Acknowledgements

I would like to express my gratitude to my supervisor, Masami Hagiya for his advice, encouragement, and offering for me a number of valuable opportunities to study. I am sincerely grateful to my mentor, Yoshihiko Kakutani for his guidance, decisive comments, enlightening, and encouragement during the research. He kindly welcomed any of my questions and took a lot of time for discussions with me. Ichiro Hasuo gave me detailed advice as well as supported me mentally. He kindly allowed me several chances for discussions. I am indebted to Naoki Kobayashi for constructive advice and essential comments. I would like to thank Masahito Hasegawa, François Le Gall, and Mingsheng Ying, for valuable comments and advice.

I express my thanks to members in NTT Communication Science Laboratories, Go Kato, Yasuhito Kawano, Hideki Sakurada, and Yasuyuki Tsukada for advice and comments on the basis of their expertise of physics and formal methods.

I am grateful to Gergei Bana, Taku Onodera, and Ben Smyth for fruitful discussions. I was supported by a grant from Graduate School of Information Science and Technology, the University of Tokyo.

I thank members and ex-members of Hagiya laboratory especially Kentaro Honda, Yusuke Kawamoto, Yoichi Hirai, Masahiro Hamano, Daisuke Kimura, Tatsuya Abe, and Yukinao Kano for useful comments, kind advice, and help. Discussing with them, I could make vague ideas concrete and improve presentations.

Contents

List of Figures

Chapter 1

Introduction

1.1 Formal Verification of Cryptographic Protocols

1.1.1 Background

Cryptographic protocols are essential elements of the infrastructure to ensure secure communication and information processing. However, security proofs of such protocols tend to be complex and difficult to verify, which is recognized by researchers [67, 37, 19]. Indeed, flaws of designs [54, 16, 20] and security proofs [66, 32] of cryptographic protocols were found years after they had been presented. The difficulty of verification of cryptographic protocols is considered to come from the following points [36].

- To define and formulate security properties, which depend on the functionality of each protocol, is difficult. Indeed, the definitions are often reconsidered [34, 28, 62, 8, 40].
- Execution models of cryptographic protocols are complex. In general, principals in a cryptographic protocol, including adversaries, run in parallel. Since each principal runs nondeterministically and probabilistically, execution models and security properties must be defined on the basis of parallelism, nondeterminacy, and probability.
- Methods to prove that cryptographic protocols satisfy defined security properties are not self-evident. Security proofs performed on the basis of such execution models tend to be quite complex.

Formal methods have been applied to model, analyze, and verify cryptographic protocols [13, 4, 11, 7, 5]. They are based on formal frameworks, including formal languages and systems to prove security properties such as inference rules. The languages are used to formalize cryptographic protocols and security properties, and the inference rules are used to perform formal proofs.

Advantages of formal methods are as follows.

- The use of formal languages precisely prevents ambiguity. Although mathematical proofs in natural languages are rigorous, ones in formal languages can be more in terms of both description and interpretation, because the syntax and semantics of them are defined mathematically.
- It is made to be explicit that all inferences in a proof obey pre-defined inference rules. On the other hand in ordinary proofs, it is not always the case. This is possibly because some inferences are apparently too obvious to write down explicitly. However, writing such inferences is still valuable when we put priority on rigor.

- Verification and proof can be partially automated. It reduces human costs as well as prevents errors. Parts that are automated depend on software tools. We introduce some of the tools in the next subsection.

1.1.2 Existing Verification Tools

CertiCrypt [7] is a framework where we interactively construct game-based cryptographic proofs in a proof assistant Coq [26]. It has been successfully applied to verify the security of prominent cryptographic protocols such as FDH [72] and OAEP [6]. The effort one needs to build the machine-checked proofs in CertiCrypt is thought to be more than moderate [5], contrasted to the high guarantees of security. In spite of the effort to construct the proofs, ones who read them have only to understand the correctness of the formalization of the target cryptographic protocol and the statement of the security to achieve. EasyCrypt [5] further reduces users' effort. It generates a whole machine-checked proof from *proof sketches*, which are representation of essence of a security proof. It was applied to verify the security of Cramer-Shoup public key encryption scheme [21], to which CertiCrypt had never been applied. Those approaches are computer-aided verification of cryptographic *proofs*, ideas of which are in general considered by men.

AVISPA [4] and ProVerif [13] are frameworks to analyze security protocols in abstract models, where cryptographic primitives are idealized. For example, ciphertexts encrypted using a public key are decrypted only using the secret key which corresponds to the public key. In AVISPA framework, a protocol designer describe a protocol paired with expected security properties in a formal language HLPSL. The scripts are translated into the intermediate format and passed to the backends including model checkers. One of them for example tries to find attacks by exploring the transition system specified by the intermediate format. Those approaches use a computer to exhaust execution states of protocols, whose number is possibly too large to do by hand, rather than obtaining cryptographic proofs.

1.2 Formal Methods for Quantum Cryptography

We may call *classical* cryptography, formal framework, process calculus, and so on for *non-quantum* ones.

1.2.1 Security of Quantum Key Distribution Protocols

Security proofs are also complex in quantum cryptography, where we must also consider attacks using entanglements. BB84 [10] is a prominent quantum key distribution (QKD) protocol, which allows two remote principals to share a secret key. Let us call the two principals and an adversary Alice, Bob, and Eve respectively. A key of the security of BB84 and other QKD protocols [53, 68] is that Alice and Bob can estimate the amount of information leakage to Eve. If Eve tries to obtain information from quantum systems passing through a quantum channel, she must measure some physical values of them. It possibly disturbs the states of quantum systems. To check the disturbance, Alice sends to Bob additional quantum systems other than those which are sources of the shared secret key. If the information leakage is judged to be too large, they abort the protocol. BB84 provides *unconditional security* that the mutual information of Alice's and Eve's key is negligible with respect to the number of quantum bits

if Alice and Bob have not aborted the protocol. The advantage of QKD protocols is that the security does not depend on adversary's computational resources, while the security of classical key exchange protocols are ensured on the basis of conjectured difficulty of computing certain functions [27].

The first security proof of BB84 is presented by Mayers [55]. It is about 50 pages long and complex. Lo and Chau proposed another QKD protocol [53] whose security proof is simple. It is based on an entanglement distillation protocol (EDP). It has the drawback that it requires quantum computers, while BB84 only requires devices for state preparation and quantum measurement. Shor and Preskill presented a simple proof of BB84 [65]. They showed that the security of BB84 is equivalent to that of a modified version of Lo and Chau's protocol (modified Lo-Chau protocol, the EDP-based protocol) [53], whose security proof is also simple. Our target of formal verification in this thesis is Shor and Preskill's proof.

1.2.2 Motivation to Apply Formal Methods to Quantum Cryptographic Protocols

QKD protocols have advantage not to depend on conjectured difficulty of computing certain functions. They are ones of the closest application to practice in the quantum information field. Actually, several companies such as Id Quantique, MagiQtechnologies, Toshiba, and NEC are developing commercial quantum cryptographic systems. It is also possible that more complex quantum protocols be presented in the future. Therefore, it is important to develop formal frameworks to verify quantum protocols' security and also make the security proofs machine-checkable.

1.2.3 Process Calculi for Quantum Protocols

Process calculi [58, 1, 11] are formal frameworks that are suitable to verify properties of parallel systems. They have successfully applied to verification of a number of classical cryptographic protocols such as Diffie-Hellman key agreement [1], Needham-Shoroeder shared and public key protocols [11], FDH [15], and Kerberos [14]. To clone the success in quantum information fields, several quantum process calculi have been proposed [59, 49, 3, 31]. Feng et al. defined a process calculus qCCS [30, 70, 31, 24, 29]. In qCCS, a quantum protocol is formalized as a configuration $(\!|P, \rho|\!)$[1]. which is a pair of a process P and a quantum mixed state ρ that is referred by the variables in P. Gay et al. defined a quantum process calculus CQP [59, 22], which is based on pi-calculus. In the later version of CQP [22], a configuration is defined as a triple $(\sigma; \tilde{q}; P)$ consisting of a collective quantum pure state σ, an indicator of P's ownership of variables $\tilde{q}$, and a process P.

An important notion in process calculi is *weak bisimulation* relation on processes [58, 1]. Processes in weak bisimulation relation behave equivalently: they perform identical actions that are visible from the outside up to invisible ones. For example, visible actions are communications of processes via public channels, and invisible ones are communications via private channels. Usage of the relation is for example as follows. If we formalize some protocol and its specification as processes and prove that they are in bisimulation relation, then we have proved the protocol satisfies the specification.

[1] In [30, 70, 31, 24, 29], a configuration is written of the form $\langle P, \rho \rangle$ using angle brackets. In this thesis, we write $(\!|P, \rho|\!)$ since we frequently write density operators using bra-ket notation.

In qCCS, weak bisimulation relation $\approx$ on configurations is defined. Their definition is successful in that the relation is closed by application of parallel composition of processes: if $(\!|P, \rho|\!) \approx (\!|Q, \sigma|\!)$ holds, then $(\!|P||R, \rho|\!) \approx (\!|Q||R, \sigma|\!)$ holds for all process R with which $P||R$ and $Q||R$ are defined. $P||R$ means that P and R run in parallel. This property of $\approx$ is called congruence. Similarly in CQP, weak bisimulation relation is defined and proven to be congruent. The congruence property is significant when we account on compositional behavioural equivalence.

Application of Quantum Process Calculi

In qCCS, quantum teleportation, super dense coding protocols [31], and simplified version of BB84 [24] are formalized. That they satisfy their specifications is also verified using weak bisimulation relation. In CQP, quantum coin-flipping game [57], quantum teleportation protocol, and quantum bit-commitment protocol are formalized and their execution are modeled [59]. Three fold repetition quantum error correcting code and its specification are formalized, and they are proven to be weakly bisimilar [22].

To extend application of process calculi to security proofs, we applied qCCS to Shor and Preskill's security proof of BB84 [65]. We previously formalized BB84 and the modified Lo and Chau's protocol as configurations of qCCS and proved their bisimilarity by hand [46].

1.3 Contributions

We addressed the following three limitations of previous work.

- Automated verification techniques have never been applied to verify that a quantum cryptographic protocol satisfies certain security criteria that are accepted in the field of quantum cryptography. In previous work [59, 61] security of BB84 was analyzed automatically for several fixed strategies of the adversary Eve. One of the strategies is *the intercept and resend attack*: she only intercepts each qubit through the quantum channel, chooses either $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$ basis randomly, measures the qubit in the basis, and resends it through the quantum channel. She thus cannot make her qubits entangled with Bob's. On the other hand, Eve is assumed to perform arbitrary quantum operations in quantum cryptographic proofs [55, 65].

- Verification of weak bisimilarity in any quantum process calculi has not been automated although by-hand verification of it is often hard when objective configurations have many long branches in their transition trees.

- Methodology to prove quantum cryptographic security using process calculi has not been established yet while considering weak bisimilarity is useful to verify equivalence of protocols. As for Shor and Preskill's security proof, equivalence of BB84 and modified Lo and Chau's protocol is stated as weak bisimilarity of configurations formalizing them. However, the way to state the security of the latter has not been obvious.

Specifically, the contributions of this thesis are as follows.

1.3.1 A Software Tool to Verify Weak Bisimilarity of qCCS Configurations

We implemented a software tool, which we call *Verifier1*, that formally verifies weak bisimilarity of qCCS configurations without recursive structures. This work, which was presented in [47], is described in Chapter 3. The overview is as follows.

- Verifier1 adopts a simplified formal framework based on qCCS.

  1. In the original syntax, there are constructors of a quantum measurement $M[\tilde{q}; x].P$ and application of a quantum operator $op[\tilde{q}].P$. Since we can also formalize a measurement as a special quantum operation, we always have two ways to formalize one. We considered criteria to select one way from the two, which is reported in [46]. We simplified the syntax reflecting the criteria so that a user who verifies equivalence of quantum cryptographic protocols using weak bisimulation can select the feasible way to formalize a quantum measurement. Let $\mathcal{P}$ be the set of processes defined by the simplified syntax. The simplification of the syntax is described in Section 3.2.
  2. qCCS's transition system is probabilistic and nondeterministic. A configuration is of the form $(\!|P, \rho|\!)$ and $\mathrm{tr}(\rho) = 1$. Suppose a probability-weighted configuration $\frac{1}{2} \bullet (\!|P, \rho|\!)$, which is interpreted to that we have the configuration $(\!|P, \rho|\!)$ with probability $\frac{1}{2}$. Instead of considering a probability-weighted configuration, we allow ρ satisfying $0 < \mathrm{tr}(\rho) \leq 1$ and interpret $\mathrm{tr}(\rho)$ as the probability to reach the configuration. For instance, we consider $(\!|P, \frac{1}{2}\rho|\!)$ in the simplified transition system instead of $\frac{1}{2} \bullet (\!|P, \rho|\!)$ in the original one. The simplified one is only nondeterministic. To verify behavioural equivalence of configurations has become easier whether it is done by hand or tool. The simplification of the semantics is described in Section 3.3. Precisely, the simplified operational semantics is defined as transition rules for elements in $\mathcal{C} := \mathcal{P} \times \mathcal{S}$, where $\mathcal{S}$ is the set of the *symbolic representations* of probability-weighted quantum states. For $\rho \in \mathcal{S}$, we write $[\![\rho]\!]$ as its interpretation as a probability-weighted quantum state. For an element in $\mathcal{C}$, we use the notation $\{\!|$ and $|\!\}$ for pairing.

- Verifier1 handles quantum states symbolically and it can be applied to security proofs. In security proofs, the dimensions of quantum states are generally unfixed, because they depend on security parameters such as the number of qubits which Alice sends to Bob. Therefore, the way to represent states in a software tool is not self-evident. In our verifier, security parameters and quantum states are represented as symbols. A user is supposed to define in Verifier1 symbolic representations of quantum states and equations on them. To compare symbolic representations, Verifier1 applies such user-defined equations to them and simplifies them. We call the simplified formal framework for the verifier *nondeterministic qCCS*. The algorithms of Verifier1 are described in Section 3.4.

- Although Verifier1 adopts the simplified syntax and operational semantics, it is sound with respect to qCCS. If Verifier1 returns *true* with two configurations and a set of valid user-defined equations as input, then the two converted configurations in qCCS are weakly bisimilar. Soundness of Verifier1 is described in Section 3.5.

1.3.2 Approximate Bisimulation for Quantum Processes

In formal verification using process calculi, notions of *approximate bisimulation* are useful: configurations in the relation behave equivalently up to negligible probability. This is possible usage of the notions. To evaluate security of a system, we first consider an ideal system that is always secure. We next prove the system and the ideal one are approximately bisimilar. This proves that the system is secure except for negligible probability.

We defined two kinds of approximate bisimulation relations in the formal framework for the verifier (nondeterministic qCCS), and studied properties of them. As described in the next subsection, we applied the second notion to the last step of Shor and Preskill's security proof. The definitions, properties, application, and limitations of the approximate bisimulation relations are described in Chapter 4.

Originally, $(\!|P, \rho|\!) \approx (\!|Q, \sigma|\!)$ means

- $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$, and
- whenever one of $(\!|P, \rho|\!)$ or $(\!|Q, \sigma|\!)$ can perform an action, the other can perform the same action up to invisible ones.

The set of quantum variables occurring in P is denoted by $\mathrm{qv}(P)$. A state space corresponds to each quantum variable. When the quantum state of all quantum variables is ρ, $\mathrm{tr}_{\mathrm{qv}(P)}(\rho)$ is the quantum state that one who does not have variables in $\mathrm{qv}(P)$ can access.

We relaxed the conditions up to gaps of probabilities of configurations' performing actions and *trace distance* $d(\cdot, \cdot)$. When we measure an arbitrary observable (a physical value) of quantum states with small trace distance, we obtain identical results with close probability.

1. The first relation $\sim_{\zeta,\eta}$, which is described in Section 4.2, is parametrized with ζ, η satisfying $0 \leq \eta, \zeta \leq 1$. Roughly speaking, $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ means

   - $d(\mathrm{tr}_{\mathrm{qv}(P)}([\![\rho]\!]), \mathrm{tr}_{\mathrm{qv}(Q)}([\![\sigma]\!])) \leq \zeta$, and
   - whenever one of $\{\!|P, \rho|\!\}$ or $\{\!|Q, \sigma|\!\}$ can perform an action with probability greater than η, the other can perform the same action up to invisible ones.

   The relation is not transitive but if $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ and $\{\!|Q, \sigma|\!\} \sim_{\zeta',\eta'} \{\!|R, \theta|\!\}$ hold, then $\{\!|P, \rho|\!\} \sim_{\zeta+\zeta', \max\{\eta,\eta'\}+2(\zeta+\zeta')} \{\!|Q, \sigma|\!\}$ holds. We proved that if $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ and $\eta > 2\zeta$ hold, then $\{\!|P||R, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q||R, \sigma|\!\}$ holds for an arbitrary process R. The condition $\eta > 2\zeta$ is reasonable, because the difference of outsider's quantum states infects her behavior.

2. The second relation $\sim$, which is described in Section 4.3, is defined when quantum states are functions of security parameters, with which the notion of *negligibility* makes sense. Roughly speaking, $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ means

   - $d(\mathrm{tr}_{\mathrm{qv}(P)}([\![\rho]\!]), \mathrm{tr}_{\mathrm{qv}(Q)}([\![\sigma]\!]))$ is negligible, and
   - whenever one of $\{\!|P, \rho|\!\}$ or $\{\!|Q, [\![\sigma]\!]|\!\}$ can perform an action with non-negligible probability, the other can perform the same action up to invisible ones.

The relation is transitive. We proved that if $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ holds, then $\{\!|P||R, \rho|\!\} \sim \{\!|Q||R, \sigma|\!\}$ holds for an arbitrary process R. The relation $\sim$ is an equivalence relation and closed under parallel composition, and thus we say it is *congruent*. This property is useful especially when we consider multiple sessions of a protocol or its behavior employed as a primitive of another protocol.

We next extended Verifier1 to verify a subset of the second approximate bisimulation relation $\sim$. The extended verifier is called *Verifier2*, which is described in Section 4.4. It uses user-defined rewriting rules of the form (ρ, σ, n) for symbolic representations ρ, σ of quantum states, and a security parameter n. Each rule is expected to satisfy that $d([\![\rho]\!], [\![\sigma]\!])$ is negligible with respect to n.

In Section 4.5, application of the relation $\sim$ to security proofs of quantum key distribution protocols is described. Although we could apply the relation to security proofs in some specific cases, whether it guarantees approximate *observational equivalence* [23, 2, 25, 11, 69] is not clear. We have considered probability-weighted quantum states instead of the original distributive system for convenience of implementation, but this causes a problem. An idea of its solution is described in Section 4.5.3. The relation $\sim_{\zeta,\eta}$ also has similar limitations.

1.3.3 Application of the Verifiers to Shor and Preskill's Security Proof of BB84

We applied Verifier1 and Verifier2 to Shor and Preskill's security proof of BB84 [65], which is described in Chapter 5. Our formal verification consists of the following two steps.

- In the first step of Shor and Preskill's security proof, equivalence of BB84 and an EDP-based protocol is proven. The latter protocol is a modification of Lo and Chau's protocol [53]. We first verified the equivalence using Verifier1. We formalized them as configurations based on our previous work [46]. We then defined equations to verify equivalence of the two protocols. The rewriting rules are obtained from properties of error-correcting codes discussed in the original proof [65] and basic facts about measurement of halves of EPR pairs. The input is the equations and configurations of BB84 and of the EDP-based protocol. Verifier1 returns *true* with the input. This work was presented in [47].

- Second, we verified security of the EDP-based protocol. We defined an ideal protocol called *EDP-ideal*, where Alice and Bob can create a shared key leaking no information to Eve. We formalized it as a configuration in Verifier2. We then defined rewriting rules to verify approximate equivalence of the two protocols. They are obtained from the second step of the original proof [65] to show the security of the EDP-based protocol. The input is the rewriting rules and configurations of the EDP-based protocol and of the ideal protocol. Verifier2 returns *true* with the input.

Formalization techniques, scripts, and experimental results are described in Chapter 5. The package of Verifier1 and Verifier2 is available from `http://hagi.is.s.u-tokyo.ac.jp/~tk/qccsverifier.tar.gz`. It includes a user manual and example scripts in the directories `doc` and `scripts`.

1.4 Related Work

1.4.1 Formal Approaches to Security of BB84

Automated Analysis using Probabilistic Model Checking

Model checking methods have been applied to analyze security of QKD protocols. Nagarajan et al. applied the probabilistic model checker PRISM 2.0 [48] to analyze BB84 [59] by calculating the probability of eavesdropping detection. They assumed restricted adversaries through noiseless quantum channels, and left a full security proof in a formal framework for future work. In contrast, we target formalization of security proofs such as Shor and Preskill's [65], where the quantum channel is assumed to be noisy and Eve performs arbitrary quantum operations.

Verification Using a Sequential Quantum Programming Language

In our previous work [45], we applied program transformation methods and Hoare logic to Shor and Preskill's security proof of BB84. We formalized BB84 and the EDP-based protocol using a Selinger's QPL [64]. We then formalized their inferences as rewriting rules of programs. Soundness of each rule was proved on the basis of the semantics of QPL. BB84 is transformed to the EDP-based protocol by the rewriting by the rules. We finally verified the security of the EDP-based protocol formally using Kakutani's quantum Hoare logic [41].

When we formally verify cryptographic protocols, an advantage of process calculi to sequential programs is that communications and nondeterminacy are explicitly written.

1.4.2 Quantum Process Calculi

CQP

In CQP's transition system, a state is a triple $(\sigma, \tilde{q}, P)$ called a configuration that consists of a map σ from a quantum variable to a quantum pure state, a set $\tilde{q}$ of quantum variables, and a process P. A configuration transits its state interacting with the outsider similarly to qCCS. An example of the configuration is as follows.

$$A \stackrel{\text{def}}{\equiv} ([q, r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; r; d![\mathsf{measure}\ r].Q)$$

There are quantum variables q and r in this configuration. The first component means the state of 2-qubit system indicated by the variables q and r is $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The second component r means that the process $d![\mathsf{measure}\ r].Q$ has access to r, but not to q. Instead, the outsider has access to q. The third component $d![\mathsf{measure}\ r].Q$ is a process that sends the measurement result (i.e. classical data) through the channel d, and executes Q.

The first state transition of the above configuration is as follows. The right-hand side is called a *mixed configuration*.

$$A \xrightarrow{\tau} \frac{1}{2}([q, r \mapsto |00\rangle]; r; d![0].Q) \oplus \frac{1}{2}([q, r \mapsto |11\rangle]; r; d![1].Q) \stackrel{\text{def}}{\equiv} B$$

For a mixed configuration, the *density matrix of the environment qubits* is defined. For the above configuration B, the density matrix is

$$\mathrm{tr}_r(\frac{1}{2}|00\rangle\langle 00|_{q,r} + \frac{1}{2}|11\rangle\langle 11|_{q,r}) = \frac{1}{2}|0\rangle\langle 0|_q + \frac{1}{2}|1\rangle\langle 1|_q.$$

For example, it is useful to identify the density matrix that B and the configuration B' below reveal to the outside.

$$B' \stackrel{\text{def}}{\equiv} \frac{1}{2}([q, r \mapsto |+0\rangle]; r; d![0].Q) \oplus \frac{1}{2}([q, r \mapsto |-1\rangle]; r; d![1].Q)$$

Next, the configuration B sends the value 0 or 1, which means that it reveals the measurement result to the outsider. When a mixed configuration is ready to send the value to the outside like B above, it transits to an *intermediate configuration*, which is a probability distribution on configurations.

$$B \xrightarrow{\tau} \frac{1}{2}([q, r \mapsto |00\rangle]; r; d![0].Q) \boxplus \frac{1}{2}([q, r \mapsto |11\rangle]; r; d![1].Q) \stackrel{\text{def}}{\equiv} C$$

The intermediate configuration probabilistically performs one of the following transitions to become a configuration.

$$C \overset{\frac{1}{2}}{\rightsquigarrow} ([q, r \mapsto |00\rangle]; r; d![0].Q) \qquad C \overset{\frac{1}{2}}{\rightsquigarrow} ([q, r \mapsto |11\rangle]; r; d![1].Q)$$

CQP is a successful formal framework with the definition of congruent bisimulation but there is no verification tool for bisimilarity of CQP configurations. Besides, there is no notion of approximate bisimulation.

Symbolic Bisimulation in qCCS

The authors of qCCS presented the notion of symbolic bisimulation for quantum processes [29]. A purpose is to verify bisimilarity algorithmically. They proved the strong symbolic bisimilarity (internal actions must be simulated) is equivalent to the strong open bisimilarity, and actually presented an algorithm to verify symbolic ground bisimilarity (outsiders do not perform quantum operations adaptively). Since our purpose is to apply a process calculus to security proofs where adversarial interference must be taken into account, we implemented a tool that verifies weak open bisimilarity on the basis of the previous version of qCCS [24].

1.4.3 Approximate Bisimulation

Approximate Bisimulation in qCCS

The authors of qCCS also presented the notion of approximate strong bisimulation in an earlier version of qCCS [70]. The notion is different from ours in this thesis. In the framework, the transitions of TPCP map application are defined to be *labeled*, namely, not τ transitions. Approximate strong bisimulation identifies transitions of TPCP maps whose *diamond distance* is not grater than some parameter. While this identification is significant, we did not directly apply this framework to our verification targets. The first reason was only *strong* bisimulation was proposed, while the protocols that we attempted to verify formally were thought to be weakly bisimilar but not to be strongly bisimilar. The second reason was that there was no conditional branch in the syntax, while aborting of an execution of a QKD protocol must be formalized.

Approximate Bisimulation in Classical Process Calculi

Ying et al. introduced a notion of approximate bisimulation in classical process calculi [71] with labeled transition systems. In their framework, the set of *actions* is a metric space. They applied the notion to verify formally approximate correctness of real time systems such as real time ACP. Our notion of approximate bisimulation is independent of theirs: distance of quantum states is considered but that of actions is not.

Approximate Bisimulation in Labeled Transition Systems with Observations

Girard et al. defined a notion of approximate bisimulation in *labeled transition systems with observations* [33]. In a labeled transition system with observations, there is an *observation map* that carries a state q to an *observation* $\langle\langle q \rangle\rangle$, and the set of observations Π is a metric space. The notion of approximate bisimulation is defined based on the distance $d_\Pi(\langle\langle q \rangle\rangle, \langle\langle q' \rangle\rangle)$. Our notion of approximate bisimulation appears to be similar to theirs when we substitute trace distance for d_Π and partial trace for $\langle\langle \cdot \rangle\rangle$. There are three different points between our work and theirs. First, we considered *weak* bisimulation relation and proved that it is closed by parallel composition of processes, which are important peculiarly in process calculi. Second, since our formal framework involves probability, transitions with probability less than some threshold η (respectively, transitions with negligible probability) is ignored in our notion. Third, since the relation $\sim$ incorporates with the notion of negligibility, it is transitive.

1.4.4 Automated Verification Tool for Classical Protocols

CryptoVerif [11] is a software tool to verify security of *classical* protocols. It has been applied to both high-level protocols [14, 12] that employ cryptographic primitives and to cryptographic schemes [15] that are possibly be used as primitives. There are two features of CryptoVerif that are related to our verifiers: it is designed on the basis of a probabilistic process calculus, and it incorporates with negligibility. Of course, it cannot be directly applied to verify security of QKD protocols. In CryptoVerif's framework, all data are classical and a process, which may be an adversary, is bounded in polynomial time, while an adversary against a QKD protocol is not.

As a proof technique, CryptoVerif applies *observational equivalence* of processes. Let $\Pr(P \rightsquigarrow a)$ be the probability that the process P transits to a process that is ready to send some data through the channel a. Two processes P and Q are observationally equivalent, written $P \cong Q$ here, if

$$|\Pr(C[P] \rightsquigarrow a) - \Pr(C[Q] \rightsquigarrow a)|$$

is negligible for all evaluation context[2] $C[\_]$ that runs in polynomial time and channel a that is not restricted. The relation $\cong$ is *congruent* by the definition, namely, if $P \cong Q$ holds, then $C[P] \cong C[Q]$ holds for all evaluation context $C[\_]$ running in polynomial time. When we consider $C[\_]$ as a polynomial time adversary that runs in parallel and interacts with the protocol P or Q, the observational equivalence of them is intuitively interpreted as indistinguishability of the protocols from an adversary.

In cryptographic proofs, security of a high-level protocol is reduced to security of the employed cryptographic primitives. Similarly, security of a cryptographic scheme is reduced to assumed difficulty of computing certain functions. In CryptoVerif, a user formalizes such assumptions as observational equivalence of processes. CryptoVerif uses such user-defined equivalences as rewriting rules: if a target process P is of the form $C[X]$ and there is a user defined equivalence $X \cong Y$, then it is rewritten to $C[Y]$. By congruence, $P \cong C[Y]$ holds. Given a process formalizing a target protocol and user-defined observational equivalences, CryptoVerif rewrites the process repeatedly until it becomes a process that is obviously secure.

[2] An evaluation context is composed by a hole, channel restrictions, and parallel compositions.

On the other hand, our tools verify bisimilarity by tracing execution paths of configurations, not by rewriting processes. Fortunately, the bisimulation in qCCS is congruent, and it is thus possible that a verification tool is designed to verify bisimilarity by rewriting. With such a verifier, bisimilarity of big-sized configurations is derived from that of some small-sized ones. Especially in proofs of security of QKD protocols, difficulty of computing certain functions is not assumed. Therefore, even if a verifier conducts rewriting, we possibly need to prove bisimilarity of such small-sized configurations unlike verification using CryptoVerif.

Chapter 2

Preliminaries

2.1 Notations

We use the following notations in this thesis.

- $\mathbb{N}$, $\mathbb{N}_+$, $\mathbb{R}$, and $\mathbb{C}$ are the set of natural numbers, the set of positive natural numbers, real numbers, and complex numbers, respectively.
- e is the base of the natural logarithm.
- For $a \in \mathbb{C}$, a^* is the complex conjugate of a, and $|a| = \sqrt{a^* a}$.
- For a linear operator A, $A^\dagger$ is the adjoint of A. I and O are the identity operator and the zero operator on a vector space with the appropriate dimension in a context. $A > 0$ means that A is positive.
- $[1..n]$, $(0, 1]$, and $[0, 1]$ are $\{1, 2, ..., n\}$, $\{x \in \mathbb{R} \mid 0 < x \leq 1\}$, and $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$, respectively.
- $\mathcal{R}^*$ is the reflexive and transitive closure of a binary relation $\mathcal{R}$. $\mathcal{R}^{-1}$ is the inverse relation of $\mathcal{R}$.
- $\Pr(A)$ is the probability that an event A happens.
- $I(X; Y)$ is the mutual information of discrete classical random variables X and Y.

2.2 Basic Quantum Information

We consulted the textbooks by Nielsen and Chuang [60, Part 1] and by Ishizaka et al. [39] in writing this section.

Quantum States and Operators

A quantum bit (qubit) is a physical system whose *pure state* is described as a unit vector in a 2-dimensional complex Hilbert space. The space is called the state space of the qubit. For a 2-dimensional complex Hilbert space $\mathcal{H}$, we can fix an orthonormal basis of $\mathcal{H}$ and write one as $|0\rangle$ and the other as $|1\rangle$. For $|\phi\rangle \in \mathcal{H}$, the adjoint $|\phi\rangle^\dagger$ of a qubit string $|\phi\rangle$ is denoted by $\langle\phi|$. For $|\phi\rangle, |\psi\rangle \in \mathcal{H}$, their inner product is written $\langle\phi|\psi\rangle$. $\mathbb{C}^2$ with the canonical inner product is an instance of 2-dimensional Hilbert space. For all $|\psi\rangle \in \mathbb{C}^2$, there exist $\alpha, \beta \in \mathbb{C}$ satisfying

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \text{ and } |\alpha|^2 + |\beta|^2 = 1, \text{ where } |0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and for all $|\phi\rangle, |\xi\rangle \in \mathbb{C}^2$, if $|\phi\rangle = a|0\rangle + b|1\rangle$ and $|\xi\rangle = c|0\rangle + d|1\rangle$, then

$$\langle\phi|\psi\rangle = a^*c + b^*d.$$

The two pure states $|0\rangle$ and $|1\rangle$ correspond to classical bit values 0 and 1. Quantum states $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ are written $|+\rangle$ and $|-\rangle$.

A discrete time evolution of a qubit is a unitary operator on its state space. For example, the following operators on $\mathbb{C}^2$ are unitary.

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

X, Y, and Z are called Pauli matrices. H is called an Hadamard transformation. The following equations hold for the matrices.

$$X|0\rangle = |1\rangle,\ X|1\rangle = |0\rangle,\ Z|+\rangle = |-\rangle,\ Z|-\rangle = |+\rangle,\ Y = iXZ,$$
$$H|0\rangle = |+\rangle,\ H|1\rangle = |-\rangle.$$

X is said to give a bit flip, and Z is said to give a phase flip.

Let $\mathcal{H}_1$,...,$\mathcal{H}_n$ be 2-dimensional complex Hilbert spaces. The pure state of a qubit string with bit length n is described as a unit vector in $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$. We write $|\psi_1...\psi_n\rangle$ as $|\psi_1\rangle \otimes ... \otimes |\psi_n\rangle$. The set $\{|x_1...x_n\rangle\}_{x_1,...,x_n \in \{0,1\}}$ is an orthonormal basis and called the *computational basis state*. In this thesis, we may convert $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ and $|\phi\rangle \otimes |\psi\rangle \in \mathcal{H}_2 \otimes \mathcal{H}_1$ each other, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces. This conversion is written as $\simeq$.

Let $|\phi_1\rangle, ..., |\phi_m\rangle$ be states of qubit strings with the same bit length and $p_1, ..., p_m$ satisfy $\sum_{i=1}^m p_i = 1$ and $0 \leq p_i \leq 1$ for all i. The quantum *mixed state* where the state is $|\phi_i\rangle$ with probability p_i for all i is denoted by the *density operator* $\sum_{i=1}^m p_i|\phi_i\rangle\langle\phi_i|$. The set of density operators on a Hilbert space $\mathcal{H}$ is written $\mathcal{D}(\mathcal{H})$. We may omit scalar multiplication when it is trivial. A local quantum operation acting on a mixed state is represented by a *trace preserving and completely-positive (TPCP) map*. A map $\mathcal{E}$ is positive if it maps a positive operator to a positive operator. A map $\mathcal{E}$ is CP if $\mathcal{E} \otimes I$ is positive for all $n \in \mathbb{N}$, which is the dimension of the domain of I. For each TPCP map $\mathcal{E}$, there exist $V_1, ..., V_k$ that satisfy $\mathcal{E}(\rho) = \sum_{i=1}^k V_i \rho V_i^\dagger$ and $\sum_{i=1}^k V_i^\dagger V_i = I$. For each CP map $\mathcal{F}$, there exist $W_1, ..., W_l$ that satisfy $\mathcal{F}(\rho) = \sum_{j=1}^l W_j \rho W_j^\dagger$.

Quantum Measurement

To obtain classical information from a quantum system in a certain state, we have to *measure* some physical value of the system in the state. A quantum measurement may change the state of the target system. A physical value that can be measured is called an *observable*. An observable of a system $|\psi\rangle \in \mathcal{H}$ is denoted by an Hermitian operator on $\mathcal{H}$, namely, an operator A satisfying $A = A^\dagger$. An Hermitian operator A has an eigenvalue decomposition $A = \sum_{i \in I} \lambda_i |i\rangle\langle i|$, where λ_i is an eigenvalue and $|i\rangle$ is the eigenvector corresponding to λ_i. A has the unique spectral decomposition $A = \sum_{j \in J} \lambda_j P_j$, where $\lambda_j = \lambda_{j'}$ implies $j = j'$ for all $j, j' \in J$. P_j is called the projector to the eigenspace of λ_j.

When we measure an observable $A = \sum_i \lambda_j P_j$ of a system in a pure state $|\psi\rangle$, we obtain the result λ_j with probability $\langle\psi|P_j|\psi\rangle$, and the post-measurement state is $\frac{P_j|\psi\rangle}{\sqrt{\langle\psi|P_j|\psi\rangle}}$. For example, if we measure an observable $Z = 1|0\rangle\langle 0| + (-1)|1\rangle\langle 1|$ of a state $\alpha|0\rangle + \beta|1\rangle$, we obtain the result 1 and the post-measurement state $|0\rangle$ with probability $|\alpha|^2$, and the result -1 and the post-measurement state $|1\rangle$ with

probability $|\beta|^2$. We can calculate the probability of obtaining each measurement result from a mixed state. Let the objective mixed state is $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. When we measure an observable $A = \sum_i \lambda_i P_i$, the probability that we obtain λ_j is

$$\sum_i p_i \langle\psi_i|P_j|\psi_i\rangle = \mathrm{tr}(P_j \rho P_j) = \mathrm{tr}(P_j \rho),$$

and the post-measurement state is

$$\sum_i \frac{p_i \langle\psi_i|P_j|\psi_i\rangle}{\mathrm{tr}(P_j\rho)} \frac{P_j|\psi_i\rangle\langle\psi_i|P_j}{\langle\psi_i|P_j|\psi_i\rangle} = \frac{P_j \rho P_j}{\mathrm{tr}(P_j\rho)}.$$

Some abbreviations about measurements are often used. We say "measure $|\psi\rangle$ in the $\{|0\rangle, |1\rangle\}$ basis" for "measure an observable $0|0\rangle\langle 0| + 1|1\rangle\langle 1|$ of $|\psi\rangle$". Similarly, we may say "measure $|\psi\rangle$ in the $\{|+\rangle, |-\rangle\}$ basis" for "measure an observable $0|+\rangle\langle +| + 1|-\rangle\langle -|$ of $|\psi\rangle$".

For a density operator ρ and an observable $\sum_j \lambda_j P_j$, $\frac{P_j \rho P_j}{\mathrm{tr}(P_j\rho)}$ is the density operator denoting the conditional probability distribution given the result λ_j of the measurement. When the pre-measurement state is ρ, the probability distribution obtained after the measurement of an observable $\sum_j \lambda_j P_j$ is

$$\sum_j \mathrm{tr}(P_j\rho) \frac{P_j \rho P_j}{\mathrm{tr}(P_j\rho)} = \sum_j P_j \rho P_j.$$

The map $\mathcal{E}_{projmeas}(\rho) = \sum_j P_j \rho P_j$ is a TPCP map. For all j, the map $\mathcal{E}^j_{projmeas}(\rho) = P_j \rho P_j$ is a CP map.

Partial Trace

Let $\rho \in \mathcal{D}(\mathcal{H}_1 \otimes \mathcal{H}_2)$. ρ can be written as $\sum_i C_i \otimes D_i$, where C_i is a linear operator on $\mathcal{H}_1$ and D_i is on $\mathcal{H}_2$ for all i. The *partial trace* of ρ by $\mathcal{H}_1$, denoted by $\mathrm{tr}_{\mathcal{H}_1}(\rho)$, is defined as $\sum_i \mathrm{tr}(C_i) D_i$. When one measures an observable $I \otimes (\sum_j \lambda_j P_j)$ on $\mathcal{H}_1 \otimes \mathcal{H}_2$, he obtains the result λ_j and the post-measurement state is $\frac{(I\otimes P_j)\rho(I\otimes P_j)}{\mathrm{tr}((I\otimes P_j)\rho)}$ with probability $\mathrm{tr}((I \otimes P_j)\rho)$ for a pre-measurement state ρ. Let us write ρ_2 for $\mathrm{tr}_{\mathcal{H}_1}(\rho)$. We have

$$\mathrm{tr}((I \otimes P_j)\rho) = \mathrm{tr}(P_j \rho_2)$$

and

$$\mathrm{tr}_{\mathcal{H}_1}\left(\frac{(I \otimes P_j)\rho(I \otimes P_j)}{\mathrm{tr}((I \otimes P_j)\rho)}\right) = \frac{P_j \rho_2 P_j}{\mathrm{tr}(P_j \rho_2)}.$$

Hence, the above measurement can be considered as the measurement of the observable $\sum_j \lambda_j P_j$ for the pre-measerment state ρ_2. Assume that Bob can measure an observable only on the partial quantum system $\mathcal{H}_2$. For the above reason, we may say that $\mathrm{tr}_{\mathcal{H}_1}(\rho)$ is the quantum state that he has access or his *view*, in the following chapters.

2.3 Quantum Error Correcting Code

The quantum key distribution (QKD) protocols that are our target of formal verification in this thesis include an error correction step after quantum communication. The error correction step is based on Calderbank-Shor-Steane (CSS) [18] quantum error correcting code (QECC), and it is described in the stabilizer formalism. In this section, we introduce necessary definitions and properties of the stabilizer formalism and CSS QECC.

We rely on the textbook by Nielsen and Chuang [60, Chapter 10].

2.3.1 Stabilizer Formalism

Stabilizers

A quantum state $|\psi\rangle$ is *stabilized* by a unitary operator U if $U|\psi\rangle = |\psi\rangle$. Let the Pauli group G_n on 2^n-dimensional space be defined as

$$G_n := \{\pm g, \pm ig \,|\, g = A_1 \otimes A_2 \cdots \otimes A_n, A_j \in \{I, X, Y, Z\} \text{ for all } j\}.$$

For $g, g' \in G_n$, g and g' is said to be *commutative* if $gg' = g'g$ and *anticommutative* if $gg' = -g'g$. For all $g, g' \in G_n$, g and g' are either commutative or anticommutative.

Let S be a subgroup of G_n and a set V_S be defined as

$$V_S := \{|\psi\rangle \,|\, \text{ For all } g \in S, g \text{ stabilizes } |\psi\rangle\}.$$

S is said to be commutative if g and g' are commutative for all $g, g' \in S$. V_S is a vector space and S is called the stabilizer of V_S. V_S is said to be non-trivial if $V_S \neq \{0\}$. The condition that V_S is non-trivial is characterized as follows.

Proposition 2.3.1. *V_S is non-trivial if and only if S is commutative and $-I \notin S$ holds.*

Let $\{g_1, g_2, ..., g_l\} \subseteq S$. If for all $a \in S$, a can be written as a product of the elements in $\{g_1, g_2, ..., g_l\}$, $\{g_1, g_2, ..., g_l\}$ is said to be a generator of S, and we write $S = \langle g_1, g_2, ..., g_l\rangle$. A generator $\{g_1, g_2, ..., g_l\}$ is said to be *independent* if for all i, g_i cannot be written as a product of the elements in $\{g_1, g_2, ..., g_l\}\backslash\{g_i\}$. The following proposition gives the dimension of the space of quantum codewords in the later discussion.

Proposition 2.3.2. *Let $S = \langle g_1, g_2, ..., g_{n-k}\rangle$. V_S is a 2^k-dimensional vector space if $\{g_1, g_2, ..., g_{n-k}\}$ is independent and commutative, and $-I \notin S$ holds.*

We have the following way to check independence and commutativity of generators $g_1, g_2, ..., g_l$ using bit vectors and matrices. For $g \in G_n$, let $r(g)$ be the bit vector with the length $2n$ defined as

$$\begin{aligned}
&r(g) := \begin{bmatrix} b_1 & b_2 & ... & b_n & b_{n+1} & ... & b_{2n} \end{bmatrix} \text{ where} \\
&g = cA_1 \otimes A_2 \otimes \cdots A_n \text{ and for all } j \text{ and } c \in \{\pm 1, \pm i\}, \\
&\text{if } A_j = I \text{ then } b_j = 0 \text{ and } b_{n+j} = 0 \\
&\text{if } A_j = X \text{ then } b_j = 1 \text{ and } b_{n+j} = 0 \\
&\text{if } A_j = Z \text{ then } b_j = 0 \text{ and } b_{n+j} = 1 \\
&\text{otherwise then } b_j = 1 \text{ and } b_{n+j} = 1
\end{aligned}$$

Next, let Λ be

$$\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}.$$

We then have that $r(g)\Lambda r(g') = 0$ if and only if g and g' are commutative. We can describe a generator $\{g_1, ..., g_l\}$ as a matrix

$$\begin{bmatrix} r(g_1) \\ r(g_2) \\ ... \\ r(g_l) \end{bmatrix}$$

though it does not keep the information of scalar multiplication ± 1, $\pm i$.

In the following discussions, we assume $\{g_1, g_2, ..., g_l\}$ are independent and commutative, and $-I \notin S$ holds for each stabilizer $S = \langle g_1, g_2, ..., g_l\rangle$.

Unitary Operations in Stabilizer Formalism

Let V_S be stabilized by a subgroup $S = \langle g_1, g_2, ..., g_l \rangle$. Let $|\psi\rangle$ be an arbitrary element in V_S. For all unitary operator U and $g \in S$, we have

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle.$$

Therefore, $UV_S := \{U|\psi\rangle \,|\, |\psi\rangle \in V_S\}$ is stabilized by $USU^\dagger := \{UgU^\dagger \,|\, g \in S\}$.

Measurement in Computational Bases in Stabilizer Formalism

Let us consider the measurement of an observable

$$g \in \{A_1 \otimes A_2 \otimes \cdots \otimes A_n, A_i \in \{I, X, Y, Z\} \text{ for all } i\} \subseteq G_n.$$

Assume the system is in a state $|\psi\rangle$ that is stabilized by $S = \langle g_1, g_2, ..., g_l \rangle$. There are two possibilities.

1. g is commutative with all $g_1, g_2, ..., g_l$.
2. g is anticommutative with some of $g_1, g_2, ..., g_l$. In this case, we can assume g is anticommutative with g_1 and commutative with $g_2, ..., g_l$ without loss of generality. When g is anticommutative with g_i, we can relabel g_i to g_1 and g_1 to g_i. We then have that g is commutative with $g_1 g_j$ if g_j is not commutative with g. We can replace g_j with $g_1 g_j$.

In fact, the result of the measurement is as follows in each case.

1. Either g or $-g$ is in S. If $g \in S$ holds, then the measurement result is 1 with probability 1. If $-g \in S$ holds, then the measurement result is -1 with probability 1. In both cases, the measurement does not change the state $|\psi\rangle$.
2. Neither g nor $-g$ is in S. We have the measurement result 1 or -1 with probability $\frac{1}{2}$. The state after the measurement is stabilized by $\langle g, g_2, ..., g_l \rangle$ if the result is 1 or by $\langle -g, g_2, ..., g_l \rangle$ if -1.

2.3.2 Stabilizer Codes

A vector space that is stabilized by $S = \langle g_1, g_2, ..., g_{n-k} \rangle$ is called $[n, k]$-*stabilizer code* and written $C(S)$. The elements in $C(S)$ are called codewords. We define the *logical* computational basis states as follows. Let $\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_k \in G_n$ make the set $\{g_1, g_2, ..., g_{n-k}, \bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_k\}$ independent and commutative. The state that is stabilized by

$$\langle g_1, g_2, ..., g_{n-k}, (-1)^{x_1}\bar{Z}_1, (-1)^{x_2}\bar{Z}_2, ..., (-1)^{x_k}\bar{Z}_k \rangle$$

is defined to be a logical computational basis state $|x_1 x_2 \cdots x_k\rangle_L$. For $j \in [1..k]$, $\bar{Z}_j$ is the logical Pauli operator Z that acts on the j-th logical qubit. Let $\bar{X}_j \in G_n$ be an operator that satisfies $\bar{X}_j \bar{Z}_j \bar{X}_j^\dagger = -\bar{Z}_j$ and $\bar{X}_i \bar{Z}_j \bar{X}_i^\dagger = \bar{Z}_i$ for all i with $i \neq j$. $\bar{X}_j$ is a logical X operator that acts on the j-th logical qubit.

In fact, it is sufficient to consider bit flip and phase flip errors to consider correction of general errors. Let us take an arbitrary element $E \in G_n$ acting on $C(S)$, where $S = \langle g_1, g_2, ..., g_{n-k} \rangle$. There are the following 3 cases.

1. E is anticommutative with some g_i for $i \in [1..n-k]$. By the error E, the stabilizer becomes $\langle g_1, g_2, ..., -g_i, ..., g_{n-k} \rangle$. When the observables $g_1, ..., g_i, ..., g_{n-k}$ are measured, the results are $1, ..., -1, ..., 1$. Therefore, we can identify the position i by the measurement.

2. $E \in S$. The error E does not change the objective state.

3. E is commutative with g_i for all $i \in [1..n-k]$ and $E \notin S$. Such errors maps a codeword in $C(S)$ to a codeword in $C(S)$. In fact, some of the errors cannot be corrected.

Let the *centralizer* $Z(S)$ of S be defined as

$$Z(S) := \{E \mid Eg = gE \text{ for all } g \in S\}.$$

We have the following theorem.

Theorem 2.3.3. *Let $C(S)$ be a stabilizer code and $\{E_j\}_{j \in J}$ be a set of operators in G_n. If $E_j^\dagger E_k \notin Z(S) - S$ for all $j, k \in J$, then $\{E_j\}_{j \in J}$ is the set of errors that can be corrected.*

We describe the way to correct errors. Let $S = \langle g_1, g_2, ..., g_{n-k} \rangle$ and $\{E_j\}_{j \in J}$ be a set of errors satisfying the condition of Theorem 2.3.3. Assume that an arbitrary error E_j has performed to an arbitrary codeword in $C(S)$. The error correction goes as follows.

- We measure the observables $g_1, g_2, ..., g_{n-k}$, and let measurement results, namely the *syndrome*, be $\beta_1, \beta_2, ..., \beta_{n-k} \in \{1, -1\}$. $E_j g_l E_j^\dagger = \beta_l g_l$ holds for all $l \in [1..n-k]$.

- If E_j is the only error that has the syndrome $\beta_1, \beta_2, ..., \beta_{n-k}$, then it is sufficient to correct the error to apply E_j to the objective system, because $E_j \beta_l g_l E_j^\dagger = \beta_l^2 g_l = g_l$ holds.

- If there is an error $E_{j'}$ in $\{E_j\}_{j \in J}$ that has the same syndrome as E_j, then it is sufficient to correct the error to apply $E_{j'}^\dagger$ to the objective system. The reason is as follows. Let P be a projector to $C(S)$. Since E_j and $E_{j'}$ have the same error syndrome, $E_j P E_j^\dagger = E_{j'} P E_{j'}^\dagger$ holds. This implies $E_{j'}^\dagger E_j P E_j^\dagger E_{j'} = P$. Because of the assumption that $\{E_j\}_{j \in J}$ satisfies the condition of Theorem 2.3.3, $E_{j'}^\dagger E_j \in S$ holds. Namely, $E_{j'}^\dagger E_j$ stabilizes the objective state.

Distance of Quantum Codes

Similarly to classical codes, the notion of distance of quantum codes is defined. The weight of $E \in G_n$ is defined as the number of the factors of E that are not equal to I. The *distance* of a stabilizer code $C(S)$ is defined as the minimum weight of $Z(S) - S$. When $C(S)$ is an $[n, k]$-stabilizer code with the distance d, $C(S)$ is said to be a $[n, k, d]$-stabilizer code. By Theorem 2.3.3, a stabilizer code with the distance $2t + 1$ can correct arbitrary errors in t qubits.

2.3.3 CSS Quantum Error Correcting Code

Stabilizer Form

CSS quantum error correcting code (QECC) [18] is described in the stabilizer formalism. It employs a classical $[n, k_1]$ code C_1 and a $[n, k_2]$ code C_2 satisfying $C_2 \subseteq C_1$. It is also assumed that both C_1 and $C_2^{\perp}$ correct t errors. Let $\mathrm{CSS}(C_1, C_2)$ be a $[n, k_1 - k_2]$ stabilizer code that is stabilized by the set whose generator is described by the following matrix

$$\begin{bmatrix} H(C_2^{\perp}) & 0 \\ 0 & H(C_1) \end{bmatrix},$$

where $H(C_2^{\perp})$ and $H(C_1)$ are parity check matrices of $C_2^{\perp}$ and C_1, whose types are $(n-k_2) \times n$ and $k_1 \times n$. The generators $g_1, g_2, ..., g_{n-(k_1-k_2)}$ are commutative and independent since $C_2 \subseteq C_1$. The distance of $\mathrm{CSS}(C_1, C_2)$ is at least $2t+1$.

Construction of CSS code

Let $w \in \{0,1\}^k$ and assume $|w\rangle$ be the state to be coded. Let $x = G_1 w \in C_1$, where G_1 is a generator matrix of C_1. The codeword $|x + C_2\rangle$ for w is defined as

$$|x + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle.$$

A parametrized $\mathrm{CSS}_{u,v}(C_1, C_2)$ code defined as follows is equivalent to $\mathrm{CSS}(C_1, C_2)$ for $u \in C_2$ and $v \in \{0,1\}^n - C_1$.

$$|x + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{u \cdot y} |x + y + v\rangle.$$

A parametrized $\mathrm{CSS}_{u,v}(C_1, C_2)$ is considered in the discussion of equivalence of BB84 and the EDP-based protocol.

2.3.4 Entanglement Distillation based on an Error Correcting Code

Let $|\beta_{00}\rangle := \frac{|00\rangle + |11\rangle}{\sqrt{2}}$. Two qubits in the state $|\beta_{00}\rangle$ are called an EPR pair. Let us consider the following scenario. First, Alice prepares $|\beta_{00}\rangle^{\otimes n} \simeq \sum_{i \in \{0,1\}^n} |i\rangle|i\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$. Second, Alice sends the qubit string whose state is in $\mathcal{H}_B$ to Bob through a noisy quantum channel. Alice and Bob want to share the halves of a smaller number $k(\leq n)$ of EPR pairs from the given state that may be influenced by noise. In fact, this is possible by quantum error correction if the number of errors is small enough to be corrected.

Let $\{g_1, ..., g_{n-k}\}$ be commutative, independent, and do not produce $-I$. Then, $C(S)$ with $S := \langle g_1 \otimes I, ..., g_{n-k} \otimes I, I \otimes g_1, ..., I \otimes g_{n-k} \rangle$ is a $[2n, 2k]$-stabilizer code. When the observables $g_1 \otimes I, ..., g_{n-k} \otimes I, I \otimes g_1, ..., I \otimes g_{n-k}$ of the state $\sum_{i \in \{0,1\}^n} |i\rangle|i\rangle$ are measured, the resulting state is stabilized by $\langle (-1)^{b_1} g_1 \otimes I, ..., (-1)^{b_{n-k}} g_{n-k} \otimes I, I(-1)^{b'_1} \otimes g_1, ..., (-1)^{b'_{n-k}} I \otimes g_{n-k} \rangle$, where $b_1, ..., b_{n-k}$ and $b'_1, ..., b'_{n-k}$ are measurement results obtained by Alice and Bob. If there is no error, $b_i = b'_i$ holds for all i because of the entanglement. In fact, the resulting state can be regarded as $|\beta_{00}\rangle^{\otimes k}$. If there are some errors, $b_i \neq b'_i$ possibly holds for some positions i. In such a case, if $b_i = 0$ and $b'_i = 1$ for example, then the state after the measurement is stabilized by $\langle ..., g_i \otimes I, ..., -I \otimes g_i, ... \rangle$. Alice can inform b_i to Bob so that they can modify the state to be stabilized by

$\langle ..., g_i \otimes I, ..., I \otimes g_i, ... \rangle$. Similarly, by Alice's informing her measurement result to Bob, they can correct the difference. Let $C(\langle g_1, ..., g_{n-k} \rangle)$ corrects t errors. In fact, $|\beta_{00}\rangle^{\otimes k}$ can be obtained even if the second halves contain at most t errors.

2.4 Quantum Key Distribution Protocols

The word BB84 does not identify one unique protocol, because there are several possible methods for *error correction* and *privacy amplification* after the quantum communication. In this paper, since we formalize Shor and Preskill's proof, the implementation of BB84 follows their paper [65]. It employs two classical linear codes C_1, C_2 that satisfy $C_2 \subseteq C_1$. In this paper, the protocol is slightly modified for simplicity: Alice only generates $2n$ qubits. This modification causes Bob to store qubits in his side, but does not affect the security at all.

2.4.1 BB84 (slightly modified)

Assumptions

- The length of codeword $n \in \mathbb{N}$ and the error threshold $h \in [0..n]$ are defined and known to Alice, Bob, and Eve.
- Classical linear codes C_1 and C_2 with length n are defined and known to Alice, Bob, and Eve. C_1 and C_2 satisfies $\{0^n\} \subseteq C_2 \subseteq C_1 \subseteq \{0,1\}^n$.
- They use quantum and public classical channels. Eve can interpolate qubits passing through the quantum channel, and listen data passing through the public classical channel.

Protocol
We denote the following protocol as $\mathsf{BB84}^{n,h}_{C_1,C_2}$

1. Alice generates two random $2n$-bit strings $d_1, ..., d_{2n}$ and $b_1, ..., b_{2n}$.
2. Alice prepares a $2n$-qubit string $q_1, ..., q_{2n}$ according to the randomness: for each $q_i \, (1 \leq i \leq 2n)$, Alice prepares the state $|0\rangle$ if $d_i = 0, b_i = 0$, $|1\rangle$ if $d_i = 1, b_i = 0$, $|+\rangle$ if $d_i = 0, b_i = 1$, $|-\rangle$ if $d_i = 1, b_i = 1$.
3. Alice sends $q_1, ..., q_{2n}$ to Bob through the quantum channel.
4. Bob receives them and announces Alice that fact.
5. Alice announces $b_1, ..., b_{2n}$ using the classical channel.
6. For each i, Bob measures q_i in $\{|0\rangle, |1\rangle\}$ basis if $b_i = 0$; in $\{|+\rangle, |-\rangle\}$ basis if $b_i = 1$. Let the results, which are either 0 or 1, of the measurement be $c_1, ..., c_{2n}$. (If no error occurs, $d_i = c_i$ for all i.) Alice randomly chooses n bits from them as check bits. Let the indices of the check bits be $k_1, ..., k_n$. Alice tells Bob $k_1, ..., k_n \, (k_1 < ... < k_n)$.
7. Bob tells Alice $c_{k_1}, ..., c_{k_n}$ using the classical channel. Alice counts the number of j's with $d_{k_j} \neq c_{k_j}$. If the number is greater than the threshold h, they abort the protocol.
8. Let x be the bitstring with the length n obtained by eliminating $d_{k_1}, ..., d_{k_n}$ from $d_1, ...d_{2n}$. Alice chooses a codeword $u \in C_1$ at random, and announces $u + x$.

9. (Error correction) Bob lets y be the bitstring with the length n obtained by eliminating $c_{k_1}, ..., c_{k_n}$ from $c_1, ...c_{2n}$, and $\tilde{w}$ be $u + x + y$. (Ideally, the condition $x + y = 0$ is expected to hold.) Bob performs error correction of $\tilde{w}$ to obtain w. If he succeeds to correct errors, $w = u$ holds.

10. (Privacy amplification) Alice lets her secret key k_A be $u + C_2$ and Bob lets his secret key k_B be $w + C_2$, where $u + C_2 := u + \sum_{y \in C_2} y$

BB84 is transformed into the following EDP-based protocol, which is a modification of the Lo and Chau's protocol [53].

2.4.2 The EDP-based Protocol

Assumptions

- The length of codeword $n \in \mathbb{N}$ and the error threshold $h \in [0..n]$ are defined and known to Alice, Bob, and Eve.
- Classical linear codes C_1 and C_2 with length n are defined and known to Alice, Bob, and Eve. C_1 and C_2 satisfies $\{0^n\} \subseteq C_2 \subseteq C_1 \subseteq \{0,1\}^n$. Alice and Bob use the CSS code constructed from C_1 and C_2.
- They use quantum, public classical, and a private classical channels. Eve can interpolate qubits passing through the quantum channel, and listen data passing through the public classical channel.

Protocol
We denote the following protocol as $\mathsf{EDP}^{n,h}_{C_1,C_2}$

1. Alice prepares $2n$ EPR pairs $(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^{\otimes 2n}$ and a random bitstring $b_1, ..., b_{2n}$.
2. For each i, Alice performs Hadamard transformation on the second half of i-th pair of $(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^{\otimes 2n}$ if $b_i = 1$. She then sends the second halves of the pairs to Bob.
3. Bob receives the halves and announces Alice that fact.
4. Alice announces $b_1, ..., b_{2n}$ through the public classical channel. For each i, Bob performs Hadamard transformations to i-th half if $b_i = 1$.
5. Alice randomly chooses n pairs from the pairs for error check. Let $k_1, ..., k_n$ be the positions. Alice tells Bob $k_1, ..., k_n$.
6. For each $j \in [1..n]$, Alice and Bob measure their halves of k_j-th pair in $\{|0\rangle, |1\rangle\}$ basis, and share the measurement results. (If no error occurs, they have the same values as the results.) If the number of errors is greater than the threshold h, they abort the protocol.
7. (Entanglement Distillation) Let $H(C_1)$ and $H(C_2^\perp)$ be the parity check matrices of C_1 and $C_2^\perp$. Alice and Bob measures the observables which are the generators described by the matrix

$$\begin{bmatrix} H(C_2^\perp) & 0 \\ 0 & H(C_1) \end{bmatrix}.$$

   Alice informs the her measurement results to Bob, and Bob corrects errors using them. The measurement results corresponding to $H(C_2^\perp)$ and $H(C_1)$ are sent through the private and public channel respectively. If the error correction succeeds, they share logical $|\beta_{00}\rangle^{\otimes(k_1-k_2)}$.

8. Alice and Bob measure their qubits in $\{|0\rangle, |1\rangle\}$ basis to obtain shared secret keys k_A and k_B.

2.4.3 Security of Quantum Key Distribution

We describe here the security criteria introduced in Nielsen and Chuaung's book [60, Chapter 12]. First, we introduce the notions of *negligible* and *overwhelming* functions.

Definition 2.4.1. *A function $f : \mathbb{N}_+ \to [0,1]$ is negligible if for all polynomial $p(\cdot)$, there exists a natural number N such that for all $n \geq N$, $f(n) \leq \frac{1}{p(n)}$ holds. A function f is non-negligible if f is not negligible.*

Definition 2.4.2. *A function $f : \mathbb{N}_+ \to [0,1]$ is overwhelming if $1-f$ is negligible, where $(1-f)(n) = 1 - f(n)$.*

The security criteria is defined as follows.

Definition 2.4.3. *Let k_A, k_B, and k_E are random variables of Alice's, Bob's, and Eve's keys under the probability distribution after the execution of a QKD protocol. The protocol is secure with respect to security parameters $s > 0$ and $l > 0$ if Alice and Bob have aborted the protocol or $\Pr(k_A = k_B)$ is overwhelming with respect to s and $I(k_A; k_E)$ is negligible with respect to l.*

In the definition above, confidentiality of the secret key is stated as "$I(k_A; k_E)$ is negligible" and correctness of the keys is stated as "$\Pr(k_A = k_B)$ is overwhelming". However, Eve can block the protocols by jamming the quantum channel [56]. If she intercepts all qubits sent from Alice and performs some operations to change their states and resends them to Bob, then the errors in check bits will be large and the protocol will be aborted.

2.5 Shor and Preskill's Security Proof

The flow of Shor and Preskill's security proof [65] is as follows. First, BB84 is shown to be equivalent for Eve to the EDP-based protocol. Concretely, equivalence means that the information obtained by Eve who adopts an arbitrary strategy is equal in the both protocols. We call this step *the transformation step*. Next, the security of the EDP-based protocol is proven. This implies the security of BB84. We call this step *the analysis step*. We explain the discussions of the two steps briefly.

2.5.1 Transformation step

The transformation starts at $\mathsf{EDP}^{n,h}_{C_1,C_2}$. The first observation is that it does not matter even if Alice measures her check bits before she sends the other halves of EPR pairs to Bob. It is the same as her choosing $|0\rangle$ or $|1\rangle$ at random. Moreover, it does not matter, even if she first measures the observables for entanglement distillation for her code bits. In fact, this is equivalent to sending $k_1 - k_2$ halves of EPR pairs encoded by the $\mathrm{CSS}_{u,v}(C_1, C_2)$ code for two random parameters $u, v \in \{0,1\}^n$. u and v are determined by the measurement results of the observables corresponding to $H(C_2^{\perp})$ and $H(C_1)$. Eventually, instead of measuring Alice's halves, she can encode a random $k_1 - k_2$ bit string using $\mathrm{CSS}_{u,v}(C_1, C_2)$ with randomly chosen u and v. The following $\mathsf{CSS}^{n,h}_{C_1,C_2}$ QKD protocol is then obtained, which is an intermediate one in the transformation.

CSS Codes Protocol

Assumptions
The same assumptions as the EDP-based protocol are used.
Protocol

1. Alice prepares $k_1 - k_2$ code bits, u, and v at random. Alice then encodes the code bits using $\mathrm{CSS}_{u,v}(C_1, C_2)$ code. Alice next prepares n random check bits and a random bitstring $b_1, ..., b_{2n}$. The string of code bits is Alice's secret key.

2. Alice randomly chooses n out of $2n$ positions, put check bits in the positions, and put code bits in the remaining positions. Let $k_1, ..., k_n$ be the check positions.

3. For each i, Alice performs Hadamard transformation to the qubits in the positions with $b_i = 1$. She then sends the qubits to Bob.

4. Bob receives the halves and announces Alice that fact.

5. Alice announces $b_1, ..., b_{2n}$ through the public classical channel. For each i, Bob performs Hadamard transformations to i-th half if $b_i = 1$.

6. Alice tells Bob the positions of check bits $k_1, ..., k_n$.

7. For each $j \in [1..n]$, Bobs measure qubits in the position of k_j in $\{|0\rangle, |1\rangle\}$ basis, and share the measurement results. If the number of errors is greater than the threshold h, they abort the protocol.

8. Alice tells Bob u through the public classical channel and v through the secret classical channel.

9. Bob decodes his qubits using u and v, and obtains his secret key.

Next, $\mathsf{CSS}^{n,h}_{C_1,C_2}$ is transformed into $\mathsf{BB84}^{n,h}_{C_1,C_2}$. When the coded secret key in C_1 prepared by Alice is $k'_A \in C_1$, the CSS codeword is

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{u \cdot y} |k'_A + y + v\rangle.$$

Since Bob only wants to have k'_A, the value of u is not necessary. Indeed, he can measure the state in $\{|0\rangle, |1\rangle\}$ basis to have the bitstring $k'_A + y_0 + v$ for some $y_0 \in C_2$. He subtracts v from it to have $k'_A + y_0$. As the secret key is the coset $k'_A + C_2$, the value of y_0 does not matter. We then assume Alice does not send u to Bob. In Bob's view, the state of the given qubit is the mixed state

$$\sum_u (\sum_y (-1)^{u \cdot y} |k'_A + y + v\rangle)(\sum_y (-1)^{u \cdot y} |k'_A + y + v\rangle)^\dagger = \sum_y |k'_A + y + v\rangle\langle k'_A + y + v|$$

The state of the right-hand side can be prepared taking $y \in C_2$ at random. Let us focus on Bob's view before obtaining v. Recall that the value of k'_A is also taken uniformly. The state is the mixed state

$$\sum_{k'_A \in C_1} \sum_{v \in \{0,1\}^n - C_1} \sum_{y \in C_2} |k'_A + y + v\rangle\langle k'_A + y + v| = \sum_{v \in \{0,1\}^n - C_1} \sum_{k'' \in C_1} |k'' + v\rangle\langle k'' + v|.$$

We observe that $k'' + v$ is uniform random in $\{0, 1\}^n$. Therefore, the related part of the protocol can be modified as follows.

- Alice chooses $k'' \in C_1$ and $v \in \{0,1\}^n - C_1$ at random, and sends $|k'' + v\rangle$ to Bob, performing Hadamard transformation randomly.
- After Hadamard transformation, Bob measures it and obtains classical bit-string $k'' + v + \epsilon$, where ϵ is the error.
- Alice tells v to Bob. Bob obtains $k'' + v + e + v = k'' + e$. As $k'' \in C_1$, Bob performs error correction. If he succeeds, he obtains k''. The shared key is the coset of k'' in C_2.

We eventually obtain $\mathsf{BB84}^{n,h}_{C_1,C_2}$ by the transformation.

2.5.2 Analysis step

A key point is that Alice and Bob can accurately judge from the error rate obtained at the step 6 whether the error correction will succeed. Since check bits are randomly chosen, the numbers of errors contained in the code bits and check bits are close ($\sharp$). The numbers of bit and phase flip errors are estimated from the check bits with $b_i = 0$ and $b_i = 1$ in the step 2. Shor and Preskill uses a lemma given by Lo and Chau [53]. If Alice and Bob share a state having fidelity $F = 1 - 2^{-s}$ with $|\beta_{00}\rangle^{\otimes k_1 - k_2}$, $I(k_A; k_E) \leq 2^{-s + \log_2(2(k_1-k_2)+s+1/\log_e 2)} + 2^{O(-2s)}$ holds. The fidelity is actually estimated from the following fact.

$F := \langle\beta_{00}|^{\otimes m} \rho' |\beta_{00}\rangle^{\otimes m} \geq \mathrm{tr}(\Pi\rho)$ holds, where
ρ and ρ' are the states of the pairs before and after the error correction, and
Π is the projector to the space in which errors in code bits are correctable.

By ($\sharp$), $\mathrm{tr}(\Pi\rho)$ is overwhelming if they have decided not to abort the protocol.

Formally, the statement of security of BB84 is as follows.

Theorem 2.5.1 (Shor-Preskill [65])**.** *Let $[n, k_1]$-code C_1 and $[n, k_2]$-code C_2 satisfies $C_2 \subseteq C_1$, and C_1 and $C_2^\perp$ correct t errors. $\mathsf{BB84}^{n,h}_{C_1,C_2}$ is secure with respect to n. Concretely,*

$\Pr(k_A = k_B) \geq 1 - e^{-\frac{1}{4}\epsilon^2 n/(\delta - \delta^2)}$, *where* $\delta = \dfrac{t}{n}, \epsilon = \delta - \dfrac{h}{n}$, *and*
$I(k_A; k_E) \leq 2^{-s + \log_2(2(k_1-k_2)+s+1/\log_e 2)} + 2^{O(-2s)}$ *hold, where s satisfies*
$s \geq \dfrac{1}{4}\epsilon^2 n/(\delta - \delta^2)$

Chapter 3

Automated Verification of Bisimilarity of qCCS configurations

3.1 qCCS

We introduce the qCCS formal framework presented by Deng and Feng [24]. Three data types $Bool, Real$, and Qbt are used for booleans, real numbers, and qubits, respectively. Let $cVar$ be a countably infinite set for classical variables, and $qVar$ be a finite set[1] $qVar$ for quantum variables. $cVar$ and $qVar$ are ranged over by $x, y, z, ...$ and $q, r, ...$ For each $q \in qVar$, its qubit-length $|q|$ is defined. A finite sequence of quantum variables is written $\tilde{q}$. When $\tilde{q} = q_1, q_2, ..., q_n$, $|\tilde{q}|$ represents $|q_1| + |q_2| + \cdots + |q_n|$. A sequence $\tilde{q} = q_1, q_2, ..., q_n$ may be regarded as a set $\{q_1, q_2, ..., q_n\}$ implicitly when there is no fear of confusion. Let Exp be a set of real expressions, and $BExp$ be a set of boolean expressions. Exp is ranged over by $e, e', ...$ $BExp$, ranged over by $b, b', ...$, is composed of constants $\texttt{true}, \texttt{false}$, atomic expressions $e \,\mathrm{rel}\, e'$, and logical connectives $\neg, \wedge, \vee$, and $\rightarrow$, where $\mathrm{rel} \in \{>, <, \geq, \leq, =\}$.

Let $cChan$ be a set of classical channels, and $qChan$ be a set of quantum channels. $cChan$ is ranged over by $c, d, ...$, and $qChan$ is ranged over by $\mathtt{c}, \mathtt{d}, ...$

For a Hilbert space $\mathcal{H}$, $\dim(\mathcal{H})$ denotes the dimension of $\mathcal{H}$. For a linear operator $A : \mathcal{H} \rightarrow \mathcal{H}$, $\dim(A)$ denotes $\dim(\mathcal{H})$. For a TPCP map $\mathcal{E} : \mathcal{D}(\mathcal{H}_A) \rightarrow \mathcal{D}(\mathcal{H}_B)$, $\mathrm{dom}(\mathcal{E})$ and $\mathrm{cod}(\mathcal{E})$ denote its domain $\mathcal{D}(\mathcal{H}_A)$ and codomain $\mathcal{D}(\mathcal{H}_B)$. For $e \in Exp$ and $b \in BExp$, $[\![e]\!]$ and $[\![b]\!]$ denote their evaluations.

Let Op, ranged over by $op, op_1, ...$, be a set of identifiers of TPCP maps. For each $op \in Op$, a corresponding TPCP map $\mathcal{E}^{op}$ satisfying $\mathrm{dom}(\mathcal{E}^{op}) = \mathrm{cod}(\mathcal{E}^{op})$ is defined.

3.1.1 Syntax

While the original syntax of qCCS allows recursive definitions of processes, we restricted them for simplicity. The sub-language is still expressive to describe protocols including our target QKD protocols. We also eliminated the constructors of choice $+$, tau $\tau.P$ and relabeling $P[f]$ because we do not use them.

Definition 3.1.1. *The syntax of qCCS process is given as follows.*

$$\begin{aligned} Proc \ni P, Q ::= {} & \texttt{nil} \mid c?x.P \mid c!e.P \mid \mathtt{c}?q.P \mid \mathtt{c}!q.P \\ & \mid \texttt{if } b \texttt{ then } P \texttt{ fi} \mid op[\tilde{q}].P \mid M[\tilde{q}; x].P \mid P||Q \mid P\backslash L \end{aligned}$$

where M is an Hermitian operator and L is a set of channels.

[1] The set of quantum variables is countably infinite in the original qCCS and each element represents a qubit, not a qubit string.

The set of quantum free variables in a process P, denoted by $\mathrm{qv}(P)$*, is inductively defined as follows.*

$$\begin{array}{ll}
\mathrm{qv}(\mathtt{nil}) = \emptyset & \mathrm{qv}(c!e.P) = \mathrm{qv}(P) \\
\mathrm{qv}(c?x.P) = \mathrm{qv}(P) & \mathrm{qv}(\mathtt{c}!q.P) = \{q\} \cup \mathrm{qv}(P) \\
\mathrm{qv}(\mathtt{c}?q.P) = \mathrm{qv}(P) - \{q\} & \mathrm{qv}(\mathtt{if}\ b\ \mathtt{then}\ P\ \mathtt{fi}) = \mathrm{qv}(P) \\
\mathrm{qv}(op[\tilde{q}].P) = \tilde{q} \cup \mathrm{qv}(P) & \mathrm{qv}(M[\tilde{q}; x].P) = \tilde{q} \cup \mathrm{qv}(P) \\
\mathrm{qv}(P||Q) = \mathrm{qv}(P) \cup \mathrm{qv}(Q) & \mathrm{qv}(P\backslash L) = \mathrm{qv}(P)
\end{array}$$

The constructors $c?x, M[\tilde{q}; x]$, and $\mathtt{c}?q$ bind a classical variable x and a quantum variable q. Bound quantum variables in P is denoted $\mathrm{qbv}(P)$*. We consider processes whose classical variables are bounded.*

For a process to be legal, the following conditions are required.

1. $\mathtt{c}!q.P \in$ *Proc only if* $q \notin \mathrm{qv}(P)$*,*
2. $P||Q \in$ *Proc only if* $\mathrm{qv}(P) \cap \mathrm{qv}(Q) = \emptyset$*.*

We explain intuitive meanings of the constructors. The process `nil` does nothing. The process $c?x.P$ receives a value of the type *Real* through the channel c, binds it to the variable x, and executes P. The process $c!e.P$ sends a value that is obtained evaluating the expression e through the channel c, and executes P. The process $\mathtt{c}?q.P$ receives a qubit through the channel `c`, and executes P. The process $\mathtt{c}!q.P$ sends a qubit indicated by the quantum variable q through the channel `c`, and executes P. The requirement *1* says that a qubit string, which is a physical object, becomes inaccessible after one sends it. The process $\mathtt{if}\ b\ \mathtt{then}\ P\ \mathtt{fi}$ executes P iff the evaluation of the condition b is *true*. The process $op[\tilde{q}].P$ performs the corresponding TPCP map $\mathcal{E}^{op}$ to the Hilbert space indicated by $\tilde{q}$, and executes P. The process $M[\tilde{q}; x].P$ measures an observable M of the quantum state indicated by $\tilde{q}$, stores the result of the measurement into a classical variable x, and executes P. The process $P||Q$ executes the process P and Q in parallel. The requirement *2* means that P and Q do not share quantum systems. The process $P\backslash L$ executes the process P with private channels in L.

For a classical variable x and a value v of the type *Real*, $P\{v/x\}$ is the process obtained replacing x with v. For quantum variables q and r, $P\{r/q\}$ is the process obtained replacing q with r.

Example 3.1.2. *Examples of the processes are as follows,*

$\mathtt{c}!r.M_1[q; x].\mathtt{nil}$

$\mathtt{measure}[r].\mathtt{c}!r.\mathtt{nil}$

$M_1[q; x].M_2[r, s; y].\mathtt{if}\ x + y \leq 4\ \mathtt{then}\ (c!(x + y).\mathtt{c}!r.\mathtt{nil}||c?z.d!z.\mathtt{d}?t.\mathtt{nil})\ \mathtt{fi}\backslash\{c\}$

where $x, y \in cVar$, $q, r, s, t \in qVar$, $c, d \in cChan$, $\mathtt{c}, \mathtt{d} \in qChan$. $M_1 = |1\rangle\langle 1|$ and $M_2 = |001\rangle\langle 001| + 2(|010\rangle\langle 010| + |011\rangle\langle 011|) + 6|110\rangle\langle 110|$ with $|q| = |r| = |t| = 1$ and $|s| = 2$. $\mathtt{measure} \in Op$ corresponds a TPCP map $\mathcal{E}^{\mathtt{measure}}(\rho) = |0\rangle\langle 0|\rho|0\rangle\langle 0| + |1\rangle\langle 1|\rho|1\rangle\langle 1|$.

3.1.2 Semantics

For each $q \in qVar$, there assumed to be a corresponding $2^{|q|}$ dimensional Hilbert space $\mathcal{H}_q$. For $\tilde{q} = q_1, q_2, ..., q_l$, let $\mathcal{H}_{\tilde{q}}$ be $\mathcal{H}_{q_1} \otimes \mathcal{H}_{q_2} \otimes \cdots \otimes \mathcal{H}_{q_l}$. Let $\mathcal{H}_S = \bigotimes_{q \in S} \mathcal{H}_q$

for $S \subseteq qVar$ and let $\mathcal{H} = \mathcal{H}_{qVar}$ [2]. Let $\mathcal{D}(\mathcal{H})$, ranged over by $\rho, \sigma, ...$, be the set of all density operators on $\mathcal{H}$. For a process to be legal with respect to the semantics, the following conditions are additionally required.

3. $op[\tilde{q}].P \in Proc$ only if $\mathrm{dom}(\mathcal{E}^{op}) = \mathcal{D}(\mathcal{H}_{\tilde{q}})$,

4. $M[\tilde{q}; x].P \in Proc$ only if $\mathrm{dom}(M) = \mathcal{H}_{\tilde{q}}$,

For $\mathcal{E}^{op}$ with $\mathrm{dom}(\mathcal{E}^{op}) = \mathcal{H}_{\tilde{q}}$, let $\mathcal{E}^{op}_{\tilde{q}} : \mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{H})$ be $I_{\mathcal{D}(\mathcal{H}_S)} \otimes \mathcal{E}^{op} \otimes I_{\mathcal{D}(\mathcal{H}_T)}$ for $S \cup T = qVar - \tilde{q}$, where $I_{\mathcal{D}(\mathcal{H}_S)}$ and $I_{\mathcal{D}(\mathcal{H}_T)}$ are identity operators on $\mathcal{D}(\mathcal{H}_S)$ and $\mathcal{D}(\mathcal{H}_T)$. Similarly, for an Hermitian operator $M : \mathcal{H}_{\tilde{q}} \to \mathcal{H}_{\tilde{q}}$ with spectrum decomposition $M = \sum_i \lambda_i E^i$, $E^i_{\tilde{q}} : \mathcal{H} \to \mathcal{H}$ is defined as $I_{\mathcal{H}_S} \otimes E^i_{\tilde{q}} \otimes I_{\mathcal{H}_T}$.

Let $Con = Proc \times \mathcal{D}(\mathcal{H})$. An element of Con is called a configuration. A configuration consisting of $P \in Proc$ and $\rho \in \mathcal{D}(\mathcal{H})$ is written $(\!|P, \rho|\!)$[3].

Example 3.1.3. *Examples of the configurations are as follows,*

$(\!|\mathtt{c}!r.M_1[q; x].\mathtt{nil}, EPR_{q,r} \otimes |01\rangle\langle 01|_s \otimes |-\rangle\langle -|_t|\!)$
$(\!|\mathtt{measure}[r].\mathtt{c}!r.\mathtt{nil}, EPR_{q,r} \otimes |00\rangle\langle 00|_s \otimes |+\rangle\langle +|_t|\!)$
$(\!|M_1[q; x].M_2[r, s; y].\mathtt{if}\ x + y \leq 4\ \mathtt{then}\ (c!(x + y).\mathtt{c}!r.\mathtt{nil} || c?z.d!z.\mathtt{d}?t.\mathtt{nil})\ \mathtt{fi} \backslash \{c\},$
$|+\rangle\langle +|_q \otimes |+\rangle\langle +|_r \otimes |10\rangle\langle 10|_s \otimes |0\rangle\langle 0|_t|\!)$

where the sets $cVar$, $qVar$, Op, and the Hermitian operators M_1 and M_2 are defined in Example 3.1.2.

qCCS has a nondeterministic and finite-support probabilistic transition system. The set of all finite-support probability distribution on Con is denoted $D(Con)$, which is ranged over by $\mu, \nu, ...$ Namely,

$$D(Con) = \{\mu \mid \sum_{(\!|P,\rho|\!) \in Con} \mu((\!|P, \rho|\!)) = 1, \text{ and for only finitely many } (\!|P, \rho|\!), \text{ we have } \mu((\!|P, \rho|\!)) > 0\}.$$

For $\mu \in D(Con)$, we write $\mu = \boxplus_{i \in I} p_i \bullet (\!|P_i, \rho_i|\!)$ if $\mu((\!|P_i, \rho_i|\!)) = p_i$ and $\sum_{i \in I} p_i = 1$ hold. For a point distribution, we may simply write $(\!|P, \rho|\!)$ instead of $1 \bullet (\!|P, \rho|\!)$. We also write $\mu = \sum_{i \in I} p_i \mu_i$ if $\mu((\!|P, \rho|\!)) = \sum_{i \in I} p_i \mu_i((\!|P, \rho|\!))$ for all $(\!|P, \rho|\!) \in Con$ and $\mu_i \in D(Con)$.

Let the set of actions Act_τ, ranged over by $\alpha, ...$, be $\{c?v, c!v, \mathtt{c}!q, \mathtt{c}?q \mid c \in cChan, \mathtt{c} \in qChan, v \text{ is of the type } Real, q \in qVar\} \cup \tau$. Channel name $\mathrm{cn}(\alpha)$ in α is defined as $\mathrm{cn}(c?v) = \mathrm{cn}(c!v) = \{c\}$, $\mathrm{cn}(\mathtt{c}!q) = \mathrm{cn}(\mathtt{c}?q) = \{\mathtt{c}\}$, and $\mathrm{cn}(\tau) = \emptyset$. Quantum bound variable $\mathrm{qbv}(\alpha)$ in α is defined as $\mathrm{qbv}(c!v) = \mathrm{qbv}(c?v) = \mathrm{qbv}(\mathtt{c}!q) = \mathrm{qbv}(\tau) = \emptyset$, and $\mathrm{qbv}(\mathtt{c}?q) = \{q\}$.

Definition 3.1.4. *The relation of transitions $\to \subseteq Con \times Act_\tau \times D(Con)$ is defined by the rules in Figure 3.1. We regard $\xrightarrow{\alpha}$ as the subset of $Con \times D(Con)$ for fixed α. The relation $\xrightarrow{\hat{\alpha}} \subseteq Con \times D(Con)$ is defined as follows.*

$$\xrightarrow{\hat{\alpha}} := \begin{cases} \xrightarrow{\tau} \cup \{((\!|P, \rho|\!), 1 \bullet (\!|P, \rho|\!))\} & (\alpha \text{ is } \tau) \\ \xrightarrow{\alpha} & (otherwise) \end{cases}$$

Example 3.1.5. *Examples of the transitions are described in Figure 3.2 and 3.3.*

[2] As assumed in Chapter 2, we identify $H_1 \otimes H_2$ with $H_2 \otimes H_1$ for Hilbert spaces H_1 and H_2. Therefore, the order of $\mathcal{H}_q$ with respect to $\otimes$ for $q \in S$ is not significant here.

[3] In [30, 70, 31, 24, 29], a configuration is written $\langle P, \rho \rangle$ using angle brackets. In this thesis, we write $(\!|P, \rho|\!)$ since we frequently write density operators using bra-ket notation.

$$\frac{v \text{ is of the type } \mathit{Real}}{(\!|c?x.P,\rho|\!) \xrightarrow{c?v} (\!|P\{v/x\},\rho|\!)}(\text{C-Inp}) \qquad \frac{[\![e]\!] = v}{(\!|c!e.P,\rho|\!) \xrightarrow{c!v} (\!|P,\rho|\!)}(\text{C-Outp})$$

$$\frac{(\!|P_1,\rho|\!) \xrightarrow{c!v} (\!|P_1',\rho|\!) \quad (\!|P_2,\rho|\!) \xrightarrow{c?v} (\!|P_2',\rho|\!)}{(\!|P_1||P_2,\rho|\!) \xrightarrow{\tau} (\!|P_1'||P_2',\rho|\!)}(\text{C-Com})$$

$$\frac{r \notin \mathrm{qv}(P)\backslash\{q\}}{(\!|\mathsf{c}?q.P,\rho|\!) \xrightarrow{\mathsf{c}?r} (\!|P\{r/q\},\rho|\!)}(\text{Q-Inp}) \qquad \frac{}{(\!|\mathsf{c}!q.P,\rho|\!) \xrightarrow{\mathsf{c}!q} (\!|P,\rho|\!)}(\text{Q-Outp})$$

$$\frac{}{(\!|op[\tilde{q}].P,\rho|\!) \xrightarrow{\tau} (\!|P,\mathcal{E}_{\tilde{q}}^{op}(\rho)|\!)}(\text{Oper}) \quad \frac{(\!|P_1,\rho|\!) \xrightarrow{\mathsf{c}?r} (\!|P_1',\rho|\!) \; (\!|P_2,\rho|\!) \xrightarrow{\mathsf{c}!r} (\!|P_2',\rho|\!)}{(\!|P_1||P_2,\rho|\!) \xrightarrow{\tau} (\!|P_1'||P_2',\rho|\!)}(\text{Q-Com})$$

$$\frac{(\!|P,\rho|\!) \xrightarrow{\alpha} \mu, [\![b]\!] = \mathit{true}}{(\!|\texttt{if } b \texttt{ then } P \texttt{ fi},\rho|\!) \xrightarrow{\alpha} \mu}(\text{Cho}) \qquad \frac{(\!|P,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|P_i,\rho_i|\!) \quad \mathrm{cn}(\alpha) \cap L = \emptyset}{(\!|P\backslash L,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|P_i\backslash L,\rho_i|\!)}(\text{Res})$$

$$\frac{(\!|P,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|P_i',\rho_i|\!) \; \mathrm{qbv}(\alpha) \cap \mathrm{qv}(Q) = \emptyset}{(\!|P||Q,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|P_i'||Q,\rho_i|\!)}(\text{IntL})$$

$$\frac{(\!|P,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|P_i',\rho_i|\!) \; \mathrm{qbv}(\alpha) \cap \mathrm{qv}(Q) = \emptyset}{(\!|Q||P,\rho|\!) \xrightarrow{\alpha} \uplus_i p_i \bullet (\!|Q||P_i',\rho_i|\!)}(\text{IntR})$$

$$\frac{}{(\!|M[\tilde{r};x].P,\rho|\!) \xrightarrow{\tau} \sum_i p_i \bullet (\!|P\{\lambda_i/x\}, E_{\tilde{r}}^i \rho E_{\tilde{r}}^i/p_i|\!)}(\text{Meas})$$

where M has the spectrum decomposition

$$M = \sum_i \lambda_i E^i, \text{ and } p_i = \mathrm{tr}(E_{\tilde{r}}^i \rho)$$

Figure 3.1: Labelled Transition Rule

3.1.3 Lifting Relations

To define weak bisimilarity, the relations of transitions $\xrightarrow{\alpha}$, $\xrightarrow{\hat{\alpha}} \subseteq \mathit{Con} \times D(\mathit{Con})$ are lifted to subsets of $D(\mathit{Con}) \times D(\mathit{Con})$. We introduce the definitions by Deng and Feng [24] here with some of the useful properties. Some definitions are rephrased in equivalent forms.

Definition 3.1.6. *For $\mathcal{R} \subseteq \mathit{Con} \times D(\mathit{Con})$, its lifted relation $\mathcal{R}^\dagger \subseteq D(\mathit{Con}) \times D(\mathit{Con})$ is defined as the smallest relation that satisfies*

- *$(\!|P,\rho|\!)\mathcal{R}\mu$ implies $1 \bullet (\!|P,\rho|\!)\mathcal{R}^\dagger\mu$, and*
- *(Linearity) $\mu_i\mathcal{R}^\dagger\nu_i$ for any $i \in I$ implies $\sum_{i\in I} p_i\mu_i\mathcal{R}^\dagger \sum_{i\in I} p_i\nu_i$ for any $p_i \in [0,1]$ with $\sum_{i\in I} p_i = 1$, where I is a finite index set.*

Proposition 3.1.7. *$\mu\mathcal{R}^\dagger\nu$ if and only if there is a finite set I such that*

$$(\!|\mathsf{c}!r.M_1[q;x].\mathsf{nil}, EPR_{q,r} \otimes |01\rangle\langle 01|_s \otimes |-\rangle\langle -|_t|\!)$$
$$\downarrow \mathsf{c}!r$$
$$(\!|M_1[q;x].\mathsf{nil}, EPR_{q,r} \otimes |01\rangle\langle 01|_s \otimes |-\rangle\langle -|_t|\!)$$
$$\downarrow \tau$$
$$\frac{1}{2}(\!|\mathsf{nil}, |00\rangle\langle 00|_{q,r} \otimes |01\rangle\langle 01|_s \otimes |-\rangle\langle -|_t|\!) \boxplus \frac{1}{2}(\!|\mathsf{nil}, |11\rangle\langle 11|_{q,r} \otimes |01\rangle\langle 01|_s \otimes |-\rangle\langle -|_t|\!)$$

$$(\!|\mathtt{measure}[r].\mathsf{c}!r.\mathsf{nil}, EPR_{q,r} \otimes |00\rangle\langle 00|_s \otimes |+\rangle\langle +|_t|\!)$$
$$\downarrow \tau$$
$$(\!|\mathsf{c}!r.\mathsf{nil}, (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|_{q,r}) \otimes |00\rangle\langle 00|_s \otimes |+\rangle\langle +|_t|\!)$$
$$\downarrow \mathsf{c}!r$$
$$(\!|\mathsf{nil}, (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|_{q,r}) \otimes |00\rangle\langle 00|_s \otimes |+\rangle\langle +|_t|\!)$$

Figure 3.2: Examples of the Transitions (1)

$$(\!|M_1[q;x].M_2[r,s;y].\mathtt{if}\ x+y \leq 4\ \mathtt{then}\ (c!(x+y).\mathsf{c}!r.\mathsf{nil}||c?z.d!z.\mathsf{d}?t.\mathsf{nil})\ \mathtt{fi}\backslash\{c\},$$
$$|+\rangle\langle +|_q \otimes |+\rangle\langle +|_r \otimes |10\rangle\langle 10|_s \otimes |0\rangle\langle 0|_t|\!)$$
$$\downarrow \tau$$
$$\frac{1}{2}(\!|M_2[r,s;y].\mathtt{if}\ 0+y \leq 4\ \mathtt{then}\ \ldots\ \mathtt{fi}, |0\rangle\langle 0|_q \otimes |+\rangle\langle +|_r \otimes |100\rangle\langle 100|_{s,t}|\!)$$
$$\boxplus\frac{1}{2}(\!|M_2[r,s;y].\mathtt{if}\ 1+y \leq 4\ \mathtt{then}\ \ldots\ \mathtt{fi}, |1\rangle\langle 1|_q \otimes |+\rangle\langle +|_r \otimes |100\rangle\langle 100|_{s,t}|\!)$$
$$\downarrow \tau$$
$$\ldots$$

$$(\!|\mathtt{if}\ 0+2 \leq 4\ \mathtt{then}\ (c!(0+2).\mathsf{c}!r.\mathsf{nil}||c?z.d!z.\mathsf{d}?t.\mathsf{nil})\ \mathtt{fi}\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!)$$
$$\downarrow \tau$$
$$(\!|\mathsf{c}!r.\mathsf{nil}||d!3.\mathsf{d}?t.\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!)$$
$$\downarrow \mathsf{c}!r \qquad\qquad \downarrow d!3$$
$$(\!|\mathsf{nil}||d!3.\mathsf{d}?t.\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!) \qquad (\!|\mathsf{c}!r.\mathsf{nil}||\mathsf{d}?t.\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!)$$
$$\downarrow d!3 \qquad\qquad \downarrow$$
$$(\!|\mathsf{nil}||\mathsf{d}?t.\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!) \qquad \ldots$$
$$\downarrow \mathsf{d}?t \qquad\qquad \downarrow \mathsf{d}?q$$
$$(\!|\mathsf{nil}||\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!) \qquad (\!|\mathsf{nil}||\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!)$$
$$\downarrow \mathsf{d}?r \qquad\qquad \downarrow \mathsf{d}?s$$
$$(\!|\mathsf{nil}||\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!) \qquad (\!|\mathsf{nil}||\mathsf{nil})\backslash\{c\}, |00100\rangle\langle 00100|_{q,r,s,t}|\!)$$

Figure 3.3: Examples of the Transitions (2)

- $\mu = \sum_{i \in I} p_i (\!| P_i, \rho_i |\!)$,
- $\nu = \sum_{i \in I} p_i \nu_i$,
- $(\!| P_i, \rho_i |\!) \mathcal{R} \nu_i$ *for all* $i \in I$.

$\mathcal{R}^\dagger$ may be simply written $\mathcal{R}$. We have that $(\!| P, \rho |\!) \xrightarrow{\alpha} \mu$ implies $(\!| P, \rho |\!) (\xrightarrow{\alpha})^\dagger \mu$. The converse is not true. Indeed, $(\!| c!1.\mathtt{nil} || c!1.\mathtt{nil}, \rho |\!) (\xrightarrow{c!1})^\dagger \frac{1}{2} (\!| \mathtt{nil} || c!1.\mathtt{nil}, \rho |\!) \boxplus \frac{1}{2} (\!| c!1.\mathtt{nil} || \mathtt{nil}, \rho |\!)$ holds but the statement does not hold that is obtained replacing $(\xrightarrow{c!1})^\dagger$ to $\xrightarrow{c!1}$.

The internal action is then defined. It represents actions that are not observed by the outsider and is important to define weak bisimilarity.

Definition 3.1.8. *The internal action* $\Rightarrow \subseteq D(Con) \times D(Con)$ *is defined as* $((\xrightarrow{\hat{\tau}})^\dagger)^*$.

Proposition 3.1.9. *The relation* $\Rightarrow \xrightarrow{\hat{\tau}} \Rightarrow$ *is equal to* $\Rightarrow$. *The relation* $\Rightarrow \xrightarrow{\hat{\alpha}} \Rightarrow$ *is linear for all* α.

Relations on Con is also lifted to those on $D(Con)$.

Definition 3.1.10. *For* $\mathcal{R} \subseteq Con \times Con$, $\mathcal{R}^\dagger \subseteq D(Con) \times D(Con)$ *is defined as*

$$\{(\mu, \nu) \mid \exists I : \textit{finite index set}. \mu = \sum_{i \in I} p_i (\!| P_i, \rho_i |\!), \nu = \sum_{i \in I} p_i (\!| Q_i, \sigma_i |\!), \\ \forall i \in I. (\!| P_i, \rho_i |\!) \mathcal{R} (\!| Q_i, \sigma_i |\!)\}.$$

Proposition 3.1.11. *For* $\mathcal{R} \subseteq Con \times Con$, $\mathcal{R}^\dagger \subseteq D(Con) \times D(Con)$ *is linear.*

3.1.4 Bisimulation

The strong and weak open bisimulation relation of qCCS configurations defined by Deng and Feng [24] is introduced. Let $\mathcal{H}_{\overline{\mathrm{qv}(P)}}$ be $\bigotimes_{q \in qVar - \mathrm{qv}(P)} \mathcal{H}_q$.

Definition 3.1.12. *A relation* $\mathcal{R} \subseteq Con \times Con$ *is a strong simulation if* $(\!| P, \rho |\!) \mathcal{R} (\!| Q, \sigma |\!)$ *implies* $\mathrm{qv}(P) = \mathrm{qv}(Q)$, $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$ *and for all TPCP map* $\mathcal{E}$ *that acts on* $\mathcal{H}_{\overline{\mathrm{qv}(P)}}$,

- *whenever* $(\!| P, \mathcal{E}(\rho) |\!) \xrightarrow{\alpha} \mu$, *there exists* ν *such that* $(\!| Q, \mathcal{E}(\sigma) |\!) \xrightarrow{\alpha} \nu$ *and* $\mu \mathcal{R}^\dagger \nu$.

$\mathcal{R}$ *is a strong bisimulation if* $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are strong simulations. The relation* $\dot{\approx}$ *is defined as the largest strong bisimulation. If* $(\!| P, \rho |\!) \dot{\approx} (\!| Q, \sigma |\!)$, *we say they are strongly bisimilar.*

Definition 3.1.13. *A relation* $\mathcal{R} \subseteq Con \times Con$ *is a weak simulation if* $(\!| P, \rho |\!) \mathcal{R} (\!| Q, \sigma |\!)$ *implies* $\mathrm{qv}(P) = \mathrm{qv}(Q)$, $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$ *and for all TPCP map* $\mathcal{E}$ *that acts on* $\mathcal{H}_{\overline{\mathrm{qv}(P)}}$.

- *whenever* $(\!| P, \mathcal{E}(\rho) |\!) \xrightarrow{\alpha} \mu$, *there exists* ν *such that* $(\!| Q, \mathcal{E}(\sigma) |\!) \Rightarrow \xrightarrow{\hat{\alpha}} \Rightarrow \nu$ *and* $\mu \mathcal{R}^\dagger \nu$

$\mathcal{R}$ *is a weak bisimulation if* $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are weak simulations. The relation* $\approx$ *is defined as the largest weak bisimulation. If* $(\!| P, \rho |\!) \approx (\!| Q, \sigma |\!)$, *we may simply say they are bisimilar instead of weakly bisimilar.*

For the bisimulation relations of qCCS configurations, ownership of quantum variables, which represent physical objects, is significant. The first condition $\mathrm{qv}(P) = \mathrm{qv}(Q)$ implies $qVar - \mathrm{qv}(P) = qVar - \mathrm{qv}(Q)$, which means the equality of quantum variables that the outsider possesses. The second condition $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$ means the equality of quantum states that the outsider can access. In the next condition, an arbitrary TPCP map $\mathcal{E}$ that acts on $qVar - \mathrm{qv}(P)$ is taken. This allows the outsider to perform an arbitrary operation to quantum systems that she can access.

Remark 3.1.14. *There is another way to define probabilistic bisimulation based on equivalence classes [50, 35, 22]. When we define by this way, an equivalent notion is in fact defined. Concretely, if $(\!|P, \rho|\!)\mathcal{R}(\!|Q, \sigma|\!)$ holds for some strong bisimulation relation $\mathcal{R}$, then*

$$\begin{aligned}
&\mathrm{qv}(P) = \mathrm{qv}(Q), \mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma), \ \textit{and} \\
&\forall \mathcal{E}_{\tilde{\tau}} : \mathcal{D}(\mathcal{H}_{qVar-\mathrm{qv}(P)}) \to \mathcal{D}(\mathcal{H}_{qVar-\mathrm{qv}(P)}).\ \forall S \in Con/\mathcal{R}. \\
&(\exists \mu.\ (\!|P, \mathcal{E}_{\tilde{\tau}}(\rho)|\!) \xrightarrow{\alpha} \mu \ \textit{and} \sum_{S\mathcal{R}X_i} \mu(X_i) = p) \\
\Leftrightarrow &\exists \nu.\ (\!|Q, \mathcal{E}_{\tilde{\tau}}(\sigma)|\!) \xrightarrow{\alpha} \nu \ \textit{and} \sum_{S\mathcal{R}Y_i} \nu(Y_i) = p))
\end{aligned}$$

hold, and conversely.

Although we consider a little different formal framework, $\approx$ has the following properties. Proposition 3.1.15 and Theorem 3.1.17 are proven similarly to the original [24]. Theorem 3.1.16 is proven similarly to the previous version [31].

Proposition 3.1.15. *$\approx$ is an equivalence relation.*

Theorem 3.1.16. *$(\!|P, \rho|\!) \approx (\!|Q, \sigma|\!)$ if and only if $\mathrm{qv}(P) = \mathrm{qv}(Q)$, $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$ and for all TPCP map $\mathcal{E}$ that acts on $\mathcal{H}_{\overline{\mathrm{qv}(P)}}$,*

1. *whenever $(\!|P, \mathcal{E}(\rho)|\!) \xrightarrow{\alpha} \mu$, there exists ν such that $(\!|Q, \mathcal{E}(\sigma)|\!) \Rightarrow \xrightarrow{\hat{\alpha}} \Rightarrow \nu$ and $\mu \approx^{\dagger} \nu$,*
2. *whenever $(\!|Q, \mathcal{E}(\sigma)|\!) \xrightarrow{\alpha} \nu$, there exists μ such that $(\!|P, \mathcal{E}(\rho)|\!) \Rightarrow \xrightarrow{\hat{\alpha}} \Rightarrow \mu$ and $\mu \approx^{\dagger} \nu$.*

Especially, the next theorem is useful to examine equivalence of protocols under the existence of other protocols.

Theorem 3.1.17. *If $(\!|P, \rho|\!) \approx (\!|Q, \sigma|\!)$,*

- *$(\!|P \backslash L, \rho|\!) \approx (\!|Q \backslash L, \sigma|\!)$, and*
- *$(\!|P || R, \rho|\!) \approx (\!|Q || R, \sigma|\!)$*

hold for all set of channels L and process R with $\mathrm{qv}(P) \cap \mathrm{qv}(Q) = \emptyset$.

We also use the following properties of *strong* bisimulation to prove the soundness of our verifier. The properties are proven similarly to those of weak bisimulation [24].

Proposition 3.1.18. *$\dot{\approx}$ is an equivalence relation.*

Proposition 3.1.19. *If $(\!|P, \rho|\!) \dot{\approx} (\!|Q, \sigma|\!)$, then $(\!|P, \mathcal{E}(\rho)|\!) \dot{\approx} (\!|Q, \mathcal{E}(\sigma)|\!)$ for all TPCP map acting on $\mathcal{H}_{\overline{\mathrm{qv}(P)}}$.*

Theorem 3.1.20. *If* $(\!|P, \rho|\!) \dot{\approx} (\!|Q, \sigma|\!)$*, then*

- $(\!|P\backslash L, \rho|\!) \dot{\approx} (\!|Q\backslash L, \sigma|\!)$*, and*
- $(\!|P||R, \rho|\!) \dot{\approx} (\!|Q||R, \sigma|\!)$

hold for all set of channels L *and process* R *with* $\mathrm{qv}(P) \cap \mathrm{qv}(Q) = \emptyset$*.*

We call the properties of $\approx$ and $\dot{\approx}$ *congruence* that are stated by Theorem 3.1.17 and Theorem 3.1.20, although the relations are *not* closed under application of *all* constructors.

Finally, we introduce some example and counter-example of bisimulation. In the following examples, let EPR be $(\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger$.

Example 3.1.21. *The following two configurations are bisimilar for an arbitrary process* $P(q^A)$ *satisfying* $q^A \in \mathrm{qv}(P(q^A))$ *and quantum state* $\rho^E \in \mathcal{D}(\mathcal{H}_{\overline{\{q^A, q^B\}}})$*.*

1. $X \stackrel{\mathrm{def}}{\equiv} (\!|\mathsf{c}!q^B.\mathtt{measure}[q^A].P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$
2. $Y \stackrel{\mathrm{def}}{\equiv} (\!|\mathtt{measure}[q^A].\mathsf{c}!q^B.P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$

A proof of the bisimilarity using Theorem 3.1.16 is as follows. For $X = (\!|P, \rho|\!) \in \mathit{Con}$, let $\mathcal{E}(X)$ be $(\!|P, \mathcal{E}(\rho)|\!)$ for a TPCP map $\mathcal{E}$. For X and Y, the conditions of quantum variables and partial traces are easily checked. Without loss of generality, we can take $I \otimes \mathcal{E}_1$ as an arbitrary TPCP map acting on $\mathcal{H}_{\overline{\{q_A, q_B\}}}$. Let ρ' be $\mathcal{E}_1(\rho^E)$. For the transition

$$(I \otimes \mathcal{E}_1)(X) \xrightarrow{\mathsf{c}!q^B} (\!|\mathtt{measure}[q^A].P(q^A), EPR_{q^A,q^B} \otimes \rho'|\!),$$

we have

$$(I \otimes \mathcal{E}_1)(Y) \Rightarrow \xrightarrow{\mathsf{c}!q^B} (\!|P(q^A), (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho'|\!).$$

For the transition

$$(I \otimes \mathcal{E}_1)(Y) \xrightarrow{\tau} (\!|\mathsf{c}!q^B.P(q^A), (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho'|\!),$$

we have $(I \otimes \mathcal{E}_1)(X) \Rightarrow (I \otimes \mathcal{E}_1)(X)$. To prove $X \approx Y$, it is sufficient to show

$$(\!|\mathtt{measure}[q^A].P(q^A), EPR_{q^A,q^B} \otimes \rho'|\!) \approx (\!|P(q^A), (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho'|\!) \ (\sharp)$$
$$\text{and } (I \otimes \mathcal{E}_1)(X) \approx (\!|\mathsf{c}!q^B.P(q^A), EPR_{q^A,q^B} \otimes \rho'|\!) \ (\flat).$$

For $(\sharp)$, the condition of partial trace holds because

$$\mathrm{tr}_{q^A}(EPR_{q^A,q^B}) = \mathrm{tr}_{q^A}((\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B})$$

holds. Let $\mathcal{E}_2$ be an arbitrary TPCP map acting on $\mathcal{H}_{\overline{\{q^A\}}}$. For the transition

$$(\!|\mathtt{measure}[q^A].P(q^A), \mathcal{E}_2(EPR_{q^A,q^B} \otimes \rho')|\!)$$
$$\xrightarrow{\tau} (\!|P(q^A), \mathcal{E}_2(\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho')|\!) \stackrel{\mathrm{def}}{\equiv} Z,$$

we have

$$(\!|P(q^A), \mathcal{E}_2(\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho')|\!) \Rightarrow Z$$

and $Z \approx^\dagger Z$. Next, for an arbitrary TPCP map $\mathcal{E}_3$ acting on $\mathcal{H}_{\overline{\{q^A\}}}$ and transition,

$$(\!|P(q^A), \mathcal{E}_3(\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{q^A,q^B} \otimes \rho^E)|\!) \xrightarrow{\alpha} \mu,$$

we have the transition

$$(\!|\texttt{measure}[q^A].P(q^A), \mathcal{E}_3(EPR_{q^A,q^B} \otimes \rho^E)|\!) \Rightarrow (\xrightarrow{\alpha})^\dagger \mu$$

and $\mu \approx^\dagger \mu$ holds. The case when $P(q^A)$ does not perform any transition is easily checked.

The condition ($\flat$) can be similarly checked.

Example 3.1.22. *The following two configurations are not bisimilar in general.*

1. $X \stackrel{\text{def}}{\equiv} (\!|\mathsf{c}!q^B.|1\rangle\langle 1|[q^A; x].P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$
2. $Y \stackrel{\text{def}}{\equiv} (\!||1\rangle\langle 1|[q^A; x].\mathsf{c}!q^B.P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$

We prove one of the necessary conditions of bisimulation cannot be satisfied. Let $P(q^A)$ do not perform any transition. For the transition

$$X \xrightarrow{\mathsf{c}!q} (\!||1\rangle\langle 1|[q^A; x].P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!),$$

the only possible action by Y that perform $\Rightarrow \xrightarrow{\mathsf{c}!q} \Rightarrow$ is

$$\begin{aligned} Y \Rightarrow \xrightarrow{\mathsf{c}!q} & \frac{1}{2} \bullet (\!|P(q^A)\{0/x\}, |00\rangle\langle 00|_{q^A,q^B} \otimes \rho^E|\!) \\ & \boxplus \frac{1}{2} \bullet (\!|P(q^A)\{1/x\}, |11\rangle\langle 11|_{q^A,q^B} \otimes \rho^E|\!) \stackrel{\text{def}}{\equiv} \nu. \end{aligned}$$

Therefore, it is necessary for bisimilarity to

$$(\!||1\rangle\langle 1|[q^A; x].P(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!) \approx \nu.$$

However, this does not hold because $\mathrm{tr}_{q^A}(EPR_{q^A,q^B}) \neq \mathrm{tr}_{q^A}(|ii\rangle\langle ii|_{q^A,q^B})$ *for* $i \in \{0, 1\}$.

3.2 Simplification of qCCS's Syntax

3.2.1 Motivation

On Formalization of Measurement

qCCS's syntax has the constructors of TPCP map application $op[\tilde{q}].P$ and quantum measurement $M[\tilde{q}, x].P$. Since a quantum measurement can also be formalized as a TPCP map, we have two ways to formalize a measurement. For example, quantum measurement of the quantum state $|+\rangle\langle +|$ is formalized in the following two ways, where the TPCP map $\mathcal{E}^{\texttt{measure}}(\rho)$ that corresponds to $\texttt{measure}[q]$ is

$|0\rangle\langle 0|\rho|0\rangle\langle 0| + |1\rangle\langle 1|\rho|1\rangle\langle 1|$, $\rho^E \in \mathcal{H}_{qVar-\{q\}}$ is an arbitrary quantum states, and $P(q)$ is an arbitrary process with $q \in \mathrm{qv}(P(q))$.

1. $(\!|1\rangle\langle 1|[q;x].P(q), |+\rangle\langle +|_q \otimes \rho^E|\!) \xrightarrow{\tau} \frac{1}{2} \bullet (\!|P(q), |0\rangle\langle 0|_q \otimes \rho^E|\!) \boxplus \frac{1}{2} \bullet (\!|P(q), |1\rangle\langle 1|_q \otimes \rho^E|\!)$
2. $(\!|\mathtt{measure}[q].P(q), |+\rangle\langle +|_q \otimes \rho^E|\!) \xrightarrow{\tau} (\!|P(q), 1/2(|0\rangle\langle 0| + |1\rangle\langle 1|)_q|\!)$

Although the two processes apparently formalize the same deed, they are not bisimilar.

Indeed, the way to formalize a quantum measurement is important in the formal verification of Shor and Preskill's security proof using qCCS. In the transformation step, the EDP-based protocol is converted to the next protocol based on the fact that nobody outside cannot distinguish the following two processes:

A. Alice measures a half of an EPR pair and then sends the other half to the outside.

B. Alice sends a half of an EPR pair to the outside and then measures the other half.

First, when the measurement is formalized using the constructor $M[\tilde{q}, x].P$, the following two configurations are obtained formalizing the above two, where $EPR = (\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger$, $\rho^E \in \mathcal{H}_{qVar-\{q^A,q^B\}}$ is an arbitrary quantum states, and $Q(q^A)$ is the successive process. They are not bisimilar.

A-1. $(\!|\mathtt{c}!q^B.|1\rangle\langle 1|[q^A;x].Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$

B-1. $(\!|1\rangle\langle 1|[q^A;x].\mathtt{c}!q^B.Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$

Second, when the measurement is formalized as a TPCP map, the following two configurations are obtained formalizing the example. They are bisimilar.

A-2. $(\!|\mathtt{c}!q^B.\mathtt{measure}[q^A].Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$,

B-2. $(\!|\mathtt{measure}[q^A].\mathtt{c}!q^B.Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$, where

$$\mathcal{E}^{\mathtt{measure}}_{q^A}(\rho) = |0\rangle\langle 0|_{q^A}\rho|0\rangle\langle 0|_{q^A} + |1\rangle\langle 1|_{q^A}\rho|1\rangle\langle 1|_{q^A}.$$

Criteria to Select the Way to Formalize

By the definition of weak bisimulation relation, whether probabilistic branches evoked by $M[\tilde{q};x]$ *exist or not* is significant in transition trees of qCCS configurations. Therefore, the two different formalization of a quantum measurement are considered to be different from the view of the outsider. In general, it is unnatural that the outsider recognize the existence of probabilistic branches *without viewing configuration's different behaviour* that depends on the result of the branch. Hence, we think that if probabilistic branches are evoked, then the insider must perform a different labelled transition. We accordingly propose a criteria to select one way from the two to formalize a quantum measurement.

- If transitions with different labels occur according to the result of the measurement, the measurement should be formalized using the constructor $M[\tilde{q};x].P$;

- otherwise, it should be formalized as a TPCP map, namely, using the constructor $op[\tilde{q}].P$.

By our criteria, we should formalize the measurement of $|+\rangle\langle+|$ in the first example as

$$1.\ (\!|\,|1\rangle\langle 1|[q;x].P(q), |+\rangle\langle+|_q \otimes \rho^E\,|\!) \xrightarrow{\tau} \frac{1}{2} \bullet (\!|P(q), |0\rangle\langle 0|_q \otimes \rho^E|\!) \boxplus \frac{1}{2} \bullet (\!|P(q), |1\rangle\langle 1|_q \otimes \rho^E|\!)$$

if $P(q)$ performs different labeled transitions according to the result. A typical case is when $P \equiv \texttt{if}\ x = 1\ \texttt{then}\ \texttt{c}!q.P'\ \texttt{fi}$ holds for some $\texttt{c}$ and P'. Otherwise, we should formalize it as

$$2.\ (\!|\texttt{measure}[q].P(q), |+\rangle\langle+|_q \otimes \rho^E|\!) \xrightarrow{\tau} (\!|P(q), 1/2(|0\rangle\langle 0| + |1\rangle\langle 1|)_q|\!).$$

Next, let us consider the processes A and B in the second example. In fact, it is natural that we assume the successive process $Q(q^A)$ does not perform different labeled transitions according to the result of the measurement of q^A. By the definition of the QKD protocols we consider, which channels Alice and Bob use does not depend on the result of the measurement of q^A. Hence, we should formalize them as follows by our criteria.

A-2. $(\!|\texttt{c}!q^B.\texttt{measure}[q^A].Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$

B-2. $(\!|\texttt{measure}[q^A].\texttt{c}!q^B.Q(q^A), EPR_{q^A,q^B} \otimes \rho^E|\!)$.

We simplified the syntax so that it reflects these criteria. We eliminated the constructions $M[\tilde{q};x].P$ and $\texttt{if}\ b\ \texttt{then}\ P\ \texttt{fi}$. Instead, we introduced a new syntax $\texttt{meas}\ q\ \texttt{then}\ P\ \texttt{saem}$, where the observable $|1\rangle\langle 1|$ on the space corresponding to the *qubit*[4] b (i.e. $|b| = 1$ must be satisfied.) is measured, and if the result is 1, then it behaves like P, else it terminates. In the new syntax, the qubit b represents the condition for the branch, which is supposed to be computed beforehand by some TPCP map. Besides, we eliminated classical communications for simplicity. Since classical data can be represented by quantum data, the elimination of the use of classical data does not weaken crucially the expressiveness of the language. Indeed, a distribution where we have the value 0 with probability p and 1 with probability $1-p$ is represented by the diagonal density operator $p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1|$.

On Ownership of Quantum System

By the definition of bisimulation relation, if $(\!|P,\rho|\!) \approx (\!|Q,\sigma|\!)$, then $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) = \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$. Intuitively, $\mathrm{qv}(P)$ is considered as the set of quantum variables of the process P's own, and $qVar - \mathrm{qv}(P)$ is the outsider's. $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) \in \mathcal{D}(\mathcal{H}_{qVar-\mathrm{qv}(P)})$ is considered as the quantum states that the outsider can access. For the bisimulation relation, ownership of quantum variables is significant. In the transitions of qCCS processes, the ownership changes by the communication between the process and the outsider by $\texttt{c}!q$ and $\texttt{c}?q$. However, there are cases where ownership changes without communication between a process and it's outsider.

$$(\!|\texttt{hadamard}[q].\texttt{nil}, |0\rangle\langle 0|_q \otimes \rho^E|\!) \xrightarrow{\tau} (\!|\texttt{nil}, |0\rangle\langle 0|_q \otimes \rho^E|\!)$$

[4]Note that the meta variable b stands for a boolean condition in the original syntax but the meta variable b stands for a quantum variable with length 1 in our simplified syntax.

In the above configurations, $\mathrm{qv}(\texttt{hadamard}[q].\texttt{nil}) = \{q\}$ and $\mathrm{qv}(\texttt{nil}) = \emptyset$. The process loses in the transition the ownership of the variable q without sending it to the outside. We added the restriction that $op[\tilde{q}].P$ is defined only if $\tilde{q} \subseteq \mathrm{qv}(P)$, and changed `nil` to a new constructor $\texttt{discard}(\tilde{q})$ that terminates keeping the quantum variables $\tilde{q}$ inside. In the original qCCS, process's termination keeping quantum variables $\tilde{q}$ inside is realized for example by $\texttt{if false then } I[\tilde{q}].\texttt{nil fi}$, where I is the identity operator.

3.2.2 Simplified Syntax

In the construction $op[\tilde{q}].P$ in the original syntax (Section 3.1), $op \in Op$ is a symbol representing a TPCP map, but we use the set of symbols representing CP maps for a technical reason, which is described in Remark 4.2.2. Let S_{op} be the set. In the construction $op[\tilde{q}].P$, however, only TPCP maps are considered.

Definition 3.2.1. *The simplified qCCS syntax is given as follows.*

$$\begin{aligned}\mathcal{P} \ni P, Q ::= {} & \texttt{discard}(\tilde{q}) \mid \texttt{c!}q.P \mid \texttt{c?}q.P \mid op[\tilde{q}].P \\ & \mid P||Q \mid \texttt{meas } b \texttt{ then } P \texttt{ saem} \mid P\backslash L,\end{aligned}$$

where b is a quantum variable with $|b| = 1$, and $op \in S_{op}$ represents a TPCP map. The set of quantum free variables $\mathrm{qv}(\cdot)$ for the simplified syntax is defined as $\mathrm{qv}(\texttt{discard}(\tilde{q})) = \tilde{q}$ and $\mathrm{qv}(\texttt{meas } b \texttt{ then } P \texttt{ saem}) = \mathrm{qv}(P)$. For a process to be legal, the following conditions are required.

1. $\texttt{c!}q.P \in \mathcal{P}$ *iff* $q \notin \mathrm{qv}(P)$.
2. $\texttt{c?}q.P \in \mathcal{P}$ *iff* $q \in \mathrm{qv}(P)$.
3. $P||Q \in \mathcal{P}$ *iff* $\mathrm{qv}(P) \cap \mathrm{qv}(Q) = \emptyset$.
4. $op[\tilde{q}].P \in \mathcal{P}$ *iff* $\tilde{q} \subseteq \mathrm{qv}(P)$.
5. $\texttt{meas } b \texttt{ then } P \texttt{ saem} \in \mathcal{P}$ *iff* $b \in \mathrm{qv}(P)$.

3.3 Simplification of Operational Semantics

We simplified the operational semantics for convenience of implementation. Instead of considering a probability distribution on configurations, we consider a probability-weighted quantum states represented by probability-weighted density operators. For example, instead of considering

$$\begin{aligned}(\!|\texttt{meas } b \texttt{ then } P \texttt{ saem}, \rho|\!) \xrightarrow{\tau} {} & p \bullet (\!|P, \frac{|1\rangle\langle 1|_b \rho |1\rangle\langle 1|_b}{p}|\!) \boxplus \\ & (1-p) \bullet (\!|\texttt{discard}(\mathrm{qv}(P)), \frac{|0\rangle\langle 0|_b \rho |0\rangle\langle 0|_b}{1-p}|\!),\end{aligned}$$

where $p = \mathrm{tr}(|1\rangle\langle 1|_b \rho)$,

we consider

$$\begin{aligned}(\!|\texttt{meas } b \texttt{ then } P \texttt{ saem}, \rho|\!) &\xrightarrow{\tau} (\!|P, |1\rangle\langle 1|_b \rho |1\rangle\langle 1|_b|\!) \\ (\!|\texttt{meas } b \texttt{ then } P \texttt{ saem}, \rho|\!) &\xrightarrow{\tau} (\!|\texttt{discard}(\mathrm{qv}(P)), |0\rangle\langle 0|_b \rho |0\rangle\langle 0|_b|\!).\end{aligned}$$

For this purpose, we define the set of probability-weighted quantum states $\Delta(\mathcal{H}) := \{p\rho \,|\, p \in [0,1], \rho \in \mathcal{D}(\mathcal{H})\}$. Any element $\rho \in \Delta(\mathcal{H})$ can be converted to an ordinary density operator $\frac{\rho}{\mathrm{tr}(\rho)} \in \mathcal{D}(\mathcal{H})$. If there is no fear of confusion, we may

simply say quantum states instead of probability-weighted quantum states. In our verification tool, elements in $\Delta(\mathcal{H})$ are symbolically represented, which is described in Section 3.3.1. Let $\mathcal{S}$ be the set of the symbolic representations. We again call a pair $\{\!|P, \rho|\!\}$ of a process P and a symbolic representation ρ of a probability-weighted quantum state a configuration, namely, the set of configurations $\mathcal{C}$ is defined as $\mathcal{P} \times \mathcal{S}$. For elements in $\mathcal{C}$, we use the notation $\{\!|$ and $|\!\}$ for pairing. For a configuration $\{\!|P, \rho|\!\} \in \mathcal{C}$, $\mathrm{tr}([\![\rho]\!])$ can be regarded as the probability of reaching it from another configuration, where $[\![\rho]\!] \in \Delta(\mathcal{H})$ is the interpretation of ρ. By this simplification, probability was excluded from the transition system. The simplified transition system is only nondeterministic, not probabilistic. The transition rules are introduced in Section 3.3.2, after the description of the symbolic representation of quantum states (Section 3.3.1).

3.3.1 Symbolic Representation of Quantum States

Since cryptographic protocols are defined with security parameters, the dimensions of quantum states, which are data in protocols, are unfixed. In our verifier, quantum states are represented as symbols. First, let $S_{nat}, S_{stat}, S_{op}$ be finite sets of symbols respectively representing natural numbers, quantum states, and CP maps are assumed.

- S_{nat} is a set of symbols representing natural numbers. A symbol $\mathtt{1}$ is an element of S_{nat}.
- S_{stat} is a set of symbols representing quantum states.
- S_{op} is a set of symbols representing CP maps. The symbols $\mathtt{proj0}$ and $\mathtt{proj1}$ are elements of S_{op}.
- A function $\mathrm{len} : qVar \to S_{nat}$ carries each quantum variable to its qubit-length. $b \in qVar$ is called a qubit variable if $\mathrm{len}(b) = \mathtt{1}$.
- A function $\mathrm{arg} : S_{stat} \cup S_{op} \to \bigcup_{n \in \mathbb{N}_+} (S_{nat})^n$ carries each symbol of quantum states or CP map to qubit-lengths of its arguments. For $i \in \{\mathtt{0}, \mathtt{1}\}$, $\mathrm{arg}(\mathtt{proj}i) = \mathtt{1}$. For example, an EPR pair $(\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger_{q,r}$ is represented as $\mathtt{EPR}[q, r]$, where $\mathtt{EPR} \in S_{stat}$, $\mathrm{len}(q) = \mathrm{len}(r) = \mathtt{1}$, and $\mathrm{arg}(\mathtt{EPR}) = (\mathtt{1}, \mathtt{1})$.

After the sets S_{nat}, S_{stat}, S_{op}, $\mathrm{len}(\cdot)$, and $\mathrm{arg}(\cdot)$ are defined, the syntax of symbolic representations of quantum states are defined.

Definition 3.3.1. *The syntax of symbolic representations of quantum states are given as follows,*

$$\mathcal{S} \ni \rho, \sigma ::= X[\tilde{q}] \mid op[\tilde{q}](\rho) \mid \rho * \sigma \mid \mathtt{Tr}[\tilde{q}](\rho)$$

where b is a qubit variable, $X \in S_{stat}$, and $op \in S_{op}$. The set of quantum variables $\mathrm{qv}(\rho)$ in symbolic representations ρ are defined as follows.

$$\mathrm{qv}(X[\tilde{q}]) = \tilde{q}, \qquad \mathrm{qv}(op[\tilde{q}](\rho)) = \mathrm{qv}(\rho),$$
$$\mathrm{qv}(\rho * \sigma) = \mathrm{qv}(\rho) \cup \mathrm{qv}(\sigma), \qquad \mathrm{qv}(\mathtt{Tr}[\tilde{q}](\rho)) = \mathrm{qv}(\rho) - \tilde{q}.$$

For a symbolic representation to be legal, the following conditions are required.

1. $X[q_1, q_2, ..., q_n] \in \mathcal{S}$ *iff* $\mathrm{arg}(X) = (\mathrm{len}(q_1), \mathrm{len}(q_2), ..., \mathrm{len}(q_n))$.

2. $op[q_1, q_2, ..., q_n](\rho) \in \mathcal{S}$ *iff* $\arg(op) = (\mathrm{len}(q_1), \mathrm{len}(q_2), ..., \mathrm{len}(q_n))$ *and* $\{q_1, q_2, ..., q_n\} \subseteq \mathrm{qv}(\rho)$.

3. $\rho * \sigma \in \mathcal{S}$ *iff* $\mathrm{qv}(\rho) \cap \mathrm{qv}(\sigma) = \emptyset$.

4. $\mathtt{Tr}[\tilde{q}](\rho) \in \mathcal{S}$ *iff* $\tilde{q} \subseteq \mathrm{qv}(\rho)$.

If $\rho, \sigma \in \mathcal{S}$ *are syntactically equal, we write* $\rho \equiv \sigma$.

Intuitive meanings are as follows. The representation $X[\tilde{q}]$ means that $\tilde{q}$'s quantum state is X. The representation $op[\tilde{q}](\rho)$ is a quantum state obtained after application of a CP map op that acts on $\tilde{q}$, to ρ. The representation $\rho * \sigma$ is the tensor product of states ρ and σ. The representation $\mathtt{proj}i[b](\rho)$ means the quantum state ρ obtained after application of the projector $|i\rangle\langle i|_b$. The representation $\mathtt{Tr}[\tilde{q}](\rho)$ means the partial trace of ρ by $\tilde{q}$.

Next, we define the formal interpretation of the symbolic representations. To define it, interpretations of the elements of S_{nat}, S_{stat}, and S_{op} must be defined beforehand. The interpretations depend on the security parameter. For a security parameter $\lambda \in \mathbb{N}_+$, the types of the interpretations $[\![\cdot]\!]_\lambda$ are as follows.

- For $n \in S_{nat}$, $[\![n]\!]_\lambda \in \mathbb{N}_+$. For $q \in qVar$ with $\mathrm{len}(q) = n$, $\mathcal{H}_q$ is $2^{[\![n]\!]_\lambda}$-dimensional.
- For $X \in S_{stat}$ with $\arg(X) = (n_1, n_2, ..., n_m)$, $[\![X]\!]_\lambda$ is an element of a Hilbert space with dimension $2^{[\![n_1]\!]_\lambda + [\![n_2]\!]_\lambda + \cdots [\![n_m]\!]_\lambda}$.
- For $op \in S_{op}$ with $\arg(op) = (n_1, n_2, ..., n_m)$, $[\![op]\!]_\lambda$ is a CP map on a Hilbert space with dimension $2^{[\![n_1]\!]_\lambda + [\![n_2]\!]_\lambda + \cdots [\![n_m]\!]_\lambda}$.

For arbitrarily fixed λ, we may simply write $[\![\cdot]\!]$ as $[\![\cdot]\!]_\lambda$. The interpretation of the symbolic representations is then defined as follows.

- $[\![X[\tilde{q}]]\!] = [\![X]\!] \in \Delta(\mathcal{H}_{\tilde{q}})$
- $[\![op[\tilde{q}](\rho)]\!] = [\![op]\!]_{\tilde{q}}([\![\rho]\!]) \in \Delta(\mathcal{H}_{\mathrm{qv}(\rho)})$
- $[\![\rho * \sigma]\!] = [\![\rho]\!] \otimes [\![\sigma]\!] \in \Delta(\mathcal{H}_{\mathrm{qv}(\rho)} \otimes \mathcal{H}_{\mathrm{qv}(\sigma)})$
- $[\![\mathtt{proj0}[b](\rho)]\!] = |0\rangle\langle 0|_b [\![\rho]\!] |0\rangle\langle 0|_b \in \Delta(\mathcal{H}_{\mathrm{qv}(\rho)})$
- $[\![\mathtt{proj1}[b](\rho)]\!] = |1\rangle\langle 1|_b [\![\rho]\!] |1\rangle\langle 1|_b \in \Delta(\mathcal{H}_{\mathrm{qv}(\rho)})$
- $[\![\mathtt{Tr}[\tilde{q}](\rho)]\!] = \mathrm{tr}_{\tilde{q}}([\![\rho]\!]) \in \Delta(\mathcal{H}_{\mathrm{qv}(\rho) - \tilde{q}})$

Example 3.3.2. *Let* $qVar = \{\mathtt{q}, \mathtt{r}\}$, $\mathrm{len}(\mathtt{q}) = \mathrm{len}(\mathtt{r}) = \mathtt{n}$ *and let* $[\![\mathtt{n}]\!]_\lambda = \lambda$, $[\![\mathtt{EPR}]\!]_\lambda = ((\frac{|00\rangle + |11\rangle}{\sqrt{2}})(\frac{|00\rangle + |11\rangle}{\sqrt{2}})^\dagger)^{\otimes \lambda} \stackrel{\mathrm{def}}{\equiv} EPR$, $\arg(\mathtt{EPR}) = (\mathtt{n}, \mathtt{n})$, $[\![\mathtt{measure}]\!]_\lambda(\rho) = \sum_{j \in \{0,1\}^\lambda} |j\rangle\langle j|\rho|j\rangle\langle j|$, *and* $\arg(\mathtt{measure}) = (\mathtt{n})$. *The interpretation of the symbolic representation* $\mathtt{measure}[\mathtt{q}](\mathtt{EPR}[\mathtt{q}, \mathtt{r}])$ *is calculated as follows.*

$$\begin{aligned}[\![\mathtt{measure}[\mathtt{q}](\mathtt{EPR}[\mathtt{q}, \mathtt{r}])]\!]_\lambda &= [\![\mathtt{measure}]\!]_\lambda \otimes I_{\mathcal{H}_\mathtt{r}}(EPR_{\mathtt{q},\mathtt{r}}) \\ &= \sum_{j \in \{0,1\}^\lambda} \frac{1}{2^\lambda} |jj\rangle\langle jj|_{\mathtt{q},\mathtt{r}}\end{aligned}$$

We next define transition rules on $\mathcal{C} = \mathcal{P} \times \mathcal{S}$.

3.3.2 Simplified Operational Semantics

Definition 3.3.3. *Let a security parameter λ be arbitrarily fixed. Let $\mathcal{A}_\tau := \{\tau\} \cup \{\mathsf{c}!q, \mathsf{c}?q \mid \mathsf{c} \in qChan, q \in qVar\}$ be the set of actions. The transition $\to \subseteq \mathcal{C} \times \mathcal{A}_\tau \times \mathcal{C}$ is defined by the rules in Figure 3.4. The transition $\xrightarrow{\hat{\alpha}}$ is defined as follows.*

$$\xrightarrow{\hat{\alpha}} := \begin{cases} \xrightarrow{\tau} \cup \{(\{\!|P,\rho|\!\}, \{\!|P,\rho|\!\})\} & (\alpha \text{ is } \tau) \\ \xrightarrow{\alpha} & (otherwise) \end{cases}$$

$$\frac{}{\{\!|\mathsf{c}!q.P,\rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|P,\rho|\!\}}\ (\text{In}) \qquad \frac{\{\!|P,\rho|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\} \quad \mathrm{cn}(\alpha) \cap L = \emptyset}{\{\!|P\backslash L,\rho|\!\} \xrightarrow{\alpha} \{\!|P'\backslash L,\rho'|\!\}}\ (\text{Res})$$

$$\frac{r \in qVar - \mathrm{qv}(P)}{\{\!|\mathsf{c}?q.P,\rho|\!\} \xrightarrow{\mathsf{c}?r} \{\!|P\{r/q\},\rho|\!\}}\ (\text{Out}) \qquad \frac{\{\!|Q,\rho|\!\} \xrightarrow{\alpha} \{\!|Q',\rho'|\!\}}{\{\!|P||Q,\rho|\!\} \xrightarrow{\alpha} \{\!|P||Q',\rho'|\!\}}\ (\text{Right})$$

$$\frac{}{\{\!|op[\tilde{q}].P,\rho|\!\} \xrightarrow{\tau} \{\!|P,op[\tilde{q}](\rho)|\!\}}\ (\text{Op}) \qquad \frac{\{\!|P,\rho|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}}{\{\!|P||Q,\rho|\!\} \xrightarrow{\alpha} \{\!|P'||Q,\rho'|\!\}}\ (\text{Left})$$

$$\frac{\{\!|P,\rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|P',\rho|\!\} \quad \{\!|Q,\rho|\!\} \xrightarrow{\mathsf{c}?q} \{\!|Q',\rho|\!\}}{\{\!|P||Q,\rho|\!\} \xrightarrow{\tau} \{\!|P'||Q',\rho|\!\}}\ (\text{Comm})$$

$$\frac{[\![\mathtt{proj1}[\mathsf{b}](\rho)]\!]_\lambda \neq O}{\{\!|\mathtt{meas}\ b\ \mathtt{then}\ P\ \mathtt{saem},\rho|\!\} \xrightarrow{\tau} \{\!|P,\mathtt{proj1}[\mathsf{b}](\rho)|\!\}}\ (\text{Meas1})$$

$$\frac{[\![\mathtt{proj0}[\mathsf{b}](\rho)]\!]_\lambda \neq O}{\{\!|\mathtt{meas}\ b\ \mathtt{then}\ P\ \mathtt{saem},\rho|\!\} \xrightarrow{\tau} \{\!|\mathtt{discard}(\mathrm{qv}(P)),\mathtt{proj0}[\mathsf{b}](\rho)|\!\}}\ (\text{Meas0})$$

Figure 3.4: Simplified Semantics

We call a new formal framework *nondeterministic qCCS* whose set of configurations is $\mathcal{C}$ and transition rules are defined in Definition 3.3.3. Although our verifier, called *Verifier1*, is implemented based on nondeterministic qCCS, it verifies the relation $\approx$ defined by Deng et al [24]. We call the property of Verifier1 *soundness*, which is further discussed in Section 3.5.

Verifier1 handles the configurations $\{\!|P,\rho|\!\}$ that consist of a process $P \in \mathcal{P}$ and a symbolic representation $\rho \in \mathcal{S}$ of a probability-weighted quantum state $[\![\rho]\!]_\lambda \in \Delta(\mathcal{H})$. Verifier1 obeys the simplified transition rules defined in Definition 3.3.3 except for (Meas$i$) for $i = 0, 1$: it performs the transition even if $[\![\mathtt{proj}i[b](\rho)]\!]_\lambda = |i\rangle\langle i|_b [\![\rho]\!]_\lambda |i\rangle\langle i|_b = O$ for $\rho \in \mathcal{S}$.

3.4 Automated Verification of Bisimilarity

3.4.1 Equality Test of Partial Traces

Calculation of Partial Traces

To verify bisimilarity, the equality of partial traces must be checked. In fact, partial traces can be to some extent calculated quite simply focusing on the structure of the expression of the quantum states. For example, suppose there are 2 qubits named q and r, and the the outsider has only r. When the quantum state of the total system is $\mathcal{E}_q(|0\rangle\langle 0|_q \otimes |1\rangle\langle 1|_r)$, the quantum state that the outsider can access is $\mathrm{tr}_q(\mathcal{E}_q(|0\rangle\langle 0|_q \otimes |1\rangle\langle 1|_r))$. We have $\mathrm{tr}_q(\mathcal{E}_q(|0\rangle\langle 0|_q \otimes |1\rangle\langle 1|_r)) = |1\rangle\langle 1|_r$ for an arbitrary operator $\mathcal{E}_q$ that acts on q, simply eliminating the state $|0\rangle\langle 0|_q$ and the operator $\mathcal{E}_q$. This is intuitively interpreted that the outsider cannot observe what happens to quantum system that he or she cannot access. For the symbolic representations, they can be simplified focusing on occurrence of quantum variables. Formulating such calculation, we obtain the following rewriting rules, where interpretation of the right-hand side of $=$ is equal to that of the left-hand side regardless of $S_{nat}, S_{stat}, S_{op}$, their interpretations, and definitions of $\mathrm{len}(\cdot)$ and $\mathrm{arg}(\cdot)$.

$$\mathtt{Tr}[\tilde{q}](\rho) = \mathtt{Tr}[\tilde{r}](\mathtt{Tr}[\tilde{s}](\rho)) \text{ if } \tilde{q} = \tilde{r} \cup \tilde{s} \tag{3.1}$$

$$\mathtt{Tr}[\tilde{q}](op[\tilde{r}](\rho)) = \mathtt{Tr}[\tilde{q}](\rho) \text{ if } \tilde{r} \subseteq \tilde{q} \text{ and } op \text{ represents a TPCP map} \tag{3.2}$$

$$\mathtt{Tr}[\tilde{q}](op[\tilde{r}](\rho)) = op[\tilde{r}](\mathtt{Tr}[\tilde{q}](\rho)) \text{ if } \tilde{q} \cap \tilde{r} = \emptyset \tag{3.3}$$

$$\mathtt{Tr}[\tilde{q}](\rho * \rho_{\tilde{q}} * \sigma) = \rho * \sigma, \text{ where } \mathrm{qv}(\rho_{\tilde{q}}) = \tilde{q} \tag{3.4}$$

Algorithm to Trace Out

Verifier1 uses the rewriting rules above. The procedure goes as follows. Let $\mathtt{Tr}[\tilde{q}](\mathcal{E}_1[\tilde{q}_1](...(\mathcal{E}_n[\tilde{q}_n](\rho_1 * ... * \rho_m))...))$ be the objective quantum state, where $\mathcal{E}_i$ is a symbol representing a map which is either trace preserving (TP) or not.

1. A set S_0 is initialized to be $\tilde{q}$.
2. For each $i\,(1 \leq i \leq n)$, $\mathcal{E}_i$'s are successively processed.
   - If $\mathcal{E}_i$ is TP and $\tilde{q}_i \subseteq S_{i-1}$ holds, then $\mathcal{E}_i[\tilde{q}_i]$ is eliminated by rule (3.2), and S_i is defined to be S_{i-1}.
   - If $\mathcal{E}_i$ is TP and $\tilde{q}_i \subseteq S_{i-1}$ does not hold, then S_i is defined to be $S_{i-1} \backslash \tilde{q}_i$, which is application of rules (3.1) and (3.3).
   - If $\mathcal{E}_i$ is not TP, then S_i is defined to be $S_{i-1} \backslash \tilde{q}_i$, which is application of rules (3.1) and (3.3).
3. A set T, recording which quantum variables related to the state has been deleted by rule (3.6), is initialized to $\emptyset$. For each $j\,(1 \leq j \leq m)$, if $\mathrm{qv}(\rho_j) \subseteq S_n$, then ρ_j is eliminated and T is updated to $T \cup \mathrm{qv}(\rho_j)$.
4. $\mathtt{Tr}[\tilde{q}]$ is rewritten to $\mathtt{Tr}[\tilde{q} - T]$.

Example 3.4.1. *A symbolic representation*

$$\mathtt{Tr[q,r,b](neg[b](proj0[b](hadamard[r](cnot[q,b](EPR[q,s] * R[r] * R[b])))))}$$

is simplified by the trace out procedure as follows.

$$
\begin{aligned}
&\mathtt{Tr}[\mathtt{q},\mathtt{r},\mathtt{b}](\mathtt{neg}[\mathtt{b}](\mathtt{proj0}[\mathtt{b}](\mathtt{hadamard}[\mathtt{r}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{r}]*\mathtt{R}[\mathtt{b}])))))\\
=&\mathtt{Tr}[\mathtt{b}](\mathtt{proj0}[\mathtt{b}](\mathtt{Tr}[\mathtt{q},\mathtt{r}](\mathtt{hadamard}[\mathtt{r}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{r}]*\mathtt{R}[\mathtt{b}])))))\\
=&\mathtt{Tr}[\mathtt{b}](\mathtt{proj0}[\mathtt{b}](\mathtt{Tr}[\mathtt{q},\mathtt{r}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{r}]*\mathtt{R}[\mathtt{b}]))))\\
=&\mathtt{Tr}[\mathtt{b}](\mathtt{proj0}[\mathtt{b}](\mathtt{Tr}[\mathtt{q}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{Tr}[\mathtt{r}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{r}]*\mathtt{R}[\mathtt{b}])))))\\
=&\mathtt{Tr}[\mathtt{b}](\mathtt{proj0}[\mathtt{b}](\mathtt{Tr}[\mathtt{q}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{b}]))))\\
=&\mathtt{Tr}[\mathtt{b},\mathtt{q}](\mathtt{proj0}[\mathtt{b}](\mathtt{cnot}[\mathtt{q},\mathtt{b}](\mathtt{EPR}[\mathtt{q},\mathtt{s}]*\mathtt{R}[\mathtt{b}])))
\end{aligned}
$$

User-defined Equations

Verifier1 also takes user-defined equations to verify equality of quantum states that are symbolically represented. The equations are of the form $\rho = \sigma$, where $\rho, \sigma \in \mathcal{S}$. An equation $\rho = \sigma$ is said to be valid if $[\![\rho]\!] = [\![\sigma]\!]$.

There is a restriction on user-defined equations $\rho = \sigma$: ρ and σ must contain the same number of $\mathtt{proj}i[b]$ for $i = \mathtt{0}, \mathtt{1}$ and for all $b \in qVar$. This makes the proof of the soundness (Theorem 3.5.11) easier.

Application of User-defined Equations

If an objective quantum state has a part that matches to the left-hand side of a user-defined equation, the part is rewritten to the right-hand side. To apply a user-defined equation, Verifier1 automatically solves commutativity of CP maps or partial traces for disjoint sets of quantum variables. For example, if the objective quantum state is $\mathtt{Tr}[\mathtt{q}](\mathtt{hadamard}[\mathtt{s}](\mathtt{EPR}[\mathtt{q},\mathtt{r}]*\mathtt{X}[\mathtt{s}]))$ and a user defines an equation $\mathtt{Tr}[\mathtt{q}](\mathtt{EPR}[\mathtt{q},\mathtt{r}]) = \mathtt{Tr}[\mathtt{q}](\mathtt{PROB}[\mathtt{q},\mathtt{r}])$ (E1), the application procedure goes as follows.

$$
\begin{aligned}
&\mathtt{Tr}[\mathtt{q}](\mathtt{hadamard}[\mathtt{s}](\mathtt{EPR}[\mathtt{q},\mathtt{r}]*\mathtt{X}[\mathtt{s}]))\\
=\ &\mathtt{hadamards}[\mathtt{s}](\mathtt{Tr}[\mathtt{q}](\mathtt{EPR}[\mathtt{q},\mathtt{r}]*\mathtt{X}[\mathtt{s}])) && \text{(by (3.2))}\\
=\ &\mathtt{hadamard}[\mathtt{s}](\mathtt{Tr}[\mathtt{q}](\mathtt{PROB}[\mathtt{q},\mathtt{r}]*\mathtt{X}[\mathtt{s}])) && \text{(by E1)}
\end{aligned}
$$

Since trace-out may have become applicable by application of user-defined rules, trace-out procedure is applied again. In each opportunity to test the equality of quantum states, each user-defined equation is applied only once. This guarantees whatever rules a user defines, the equality test terminates.

Equality Test after the Rewriting

After the rewriting by user-defined equations and trace out, equality of the two symbolic representations are checked up to exchange of the order of CP map application and tensor product. For example, symbolic expressions

$$
\begin{aligned}
&\mathtt{Tr}[\mathtt{q}](\mathtt{hadamard}[\mathtt{s}](\mathtt{bitflip}[\mathtt{r}](\mathtt{EPR}[\mathtt{q},\mathtt{r}]*\mathtt{X}[\mathtt{s}]))) \text{ and}\\
&\mathtt{Tr}[\mathtt{q}](\mathtt{bitflip}[\mathtt{r}](\mathtt{hadamard}[\mathtt{s}](\mathtt{X}[\mathtt{s}]*\mathtt{EPR}[\mathtt{q},\mathtt{r}])))
\end{aligned}
$$

must be judged to be equal. Verifier1 automatically judges the equality by syntactically checking disjointness of CP maps' arguments and by sorting environment symbols by name, which are concatenated by "$*$".

3.4.2 Algorithm to Check Bisimilarity

The recursive procedure to verify bisimilarity is as follows. It returns either *true* or *false*.

1. The procedure takes as input two configurations $\{\!|P_0, \rho_0|\!\}$, $\{\!|Q_0, \sigma_0|\!\}$ and user-defined equations *eqs* on quantum states.

2. If P_0 and Q_0 can perform any τ-transitions of TPCP map applications, they are all performed at this point. Let $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ be the configurations thus obtained.

3. Whether $\mathrm{qv}(P) = \mathrm{qv}(Q)$ is checked. If it does not hold, the procedure returns *false*.

4. Whether $\mathtt{Tr}[\mathrm{qv}(P)](\rho) = \mathtt{Tr}[\mathrm{qv}(Q)](\sigma)$ is checked using *eqs*. The procedure to check equality of quantum states are described in the previous subsection. If it does not hold, the procedure returns *false*.

5. A new TPCP map symbol $\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)]$ that stands for an arbitrary operation is generated.

6. (a) For each $\{\!|P', \rho'|\!\}$ such that
   - $\{\!|P, \mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ holds, and
   - neither
     - $\rho' \equiv \mathtt{proj0}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))$ nor
     - $\rho' \equiv \mathtt{proj1}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))$

     holds for any $b \in \mathrm{qv}(P)$,

   the procedure checks whether there exists $\{\!|Q', \sigma'|\!\}$ such that

   $$\{\!|Q, \mathcal{E}[\mathrm{qv}(\sigma) - \mathrm{qv}(Q)](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$$

   holds and the procedure returns *true* with the input $\{\!|P', \rho'|\!\}$, $\{\!|Q', \sigma'|\!\}$, and *eqs*.

   (b) For each pair $(\{\!|P', \rho'|\!\}, \{\!|P'', \rho''|\!\})$ such that
   - $\{\!|P, \mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho)|\!\} \xrightarrow{\tau} \{\!|P', \rho'|\!\}$,
   - $\rho' \equiv \mathtt{proj0}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))$,
   - $\{\!|P, \mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho)|\!\} \xrightarrow{\tau} \{\!|P'', \rho''|\!\}$, and
   - $\rho'' \equiv \mathtt{proj1}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))$

   hold for some $b \in \mathrm{qv}(P)$, the procedure checks whether there exists a pair $(\{\!|Q', \sigma'|\!\}, \{\!|Q'', \sigma''|\!\})$ such that
   - $\{\!|Q, \mathcal{E}[\mathrm{qv}(\sigma) - \mathrm{qv}(Q)](\sigma)|\!\} \xrightarrow{\tau*} \{\!|\hat{Q}, \hat{\sigma}|\!\}$,
   - $\{\!|\hat{Q}, \hat{\sigma}|\!\} \xrightarrow{\tau} \{\!|\hat{Q}', \mathtt{proj0}[b](\hat{\sigma})|\!\} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$, and
   - $\{\!|\hat{Q}, \hat{\sigma}|\!\} \xrightarrow{\tau} \{\!|\hat{Q}'', \mathtt{proj1}[b](\hat{\sigma})|\!\} \xrightarrow{\tau*} \{\!|Q'', \sigma''|\!\}$

   hold for some $\hat{Q}$, $\hat{Q}'$, and $\hat{Q}''$, and
   - Verifier1 returns *true* with $(\{\!|P', \rho'|\!\}, \{\!|Q', \sigma'|\!\})$ and *eqs*, and
   - Verifier1 returns *true* with $(\{\!|P'', \rho''|\!\}, \{\!|Q'', \sigma''|\!\})$ and *eqs*.

   If there exists, it goes to the next step 7. Otherwise, it returns *false*.

7. For each $\{\!|Q', \sigma'|\!\}$ such that $\{\!|Q, \mathcal{E}[\mathrm{qv}(\sigma) - \mathrm{qv}(Q)](\sigma)|\!\} \xrightarrow{\alpha} \{\!|Q', \sigma'|\!\}$, the procedure checks the symmetric condition of the step 6. If there exists, it returns *true*. Otherwise, it returns *false*.

The procedure always terminates. This is because the transition of the processes is finite and equality check in the step 4 always terminates.

The step 2 prominently decreases the spaces to search. This is based on the fact that $(\!|op_1[\tilde{q}].P||op_2[\tilde{r}].Q, \rho|\!)$ and $(\!|P||Q, \mathcal{F}^{\tilde{r}}_{op_2}(\mathcal{E}^{\tilde{q}}_{op_1}(\rho))|\!)$ are bisimilar, and $\mathcal{F}^{\tilde{r}}_{op_2}(\mathcal{E}^{\tilde{q}}_{op_1}(\rho)) = \mathcal{E}^{\tilde{q}}_{op_1}(\mathcal{F}^{\tilde{r}}_{op_2}(\rho))$ holds because $\tilde{q} \cap \tilde{r} = \emptyset$ and $\mathrm{qv}(P) \cap \mathrm{qv}(Q) = \emptyset$ hold.

In Section 3.5, when we prove soundness of Verifier1, we apply the fact that the numbers of `proj`i's are equal in ρ and σ if Verifier1 returns *true* with $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$. The reason is as follows. There is the restriction that the both sides of an equation contains the same number of `proj`i's. This implies that rewriting by an arbitrary user-defined equation does not change the number of `proj`i's in a symbolic representation. Besides, the trace out procedure does not eliminate `proj`i's. Therefore, the numbers of `proj`i's must be equal to have passed the equality test in the step 4.

We make here a remark about the step 6 (b). Suppose the following transitions are performed.

- $\{\!|P, \mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho)|\!\} \xrightarrow{\tau} \{\!|P', \mathtt{proj0}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))|\!\}$
- $\{\!|P, \mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho)|\!\} \xrightarrow{\tau} \{\!|P'', \mathtt{proj1}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))|\!\}$

To return *true* with $\{\!|P, \rho|\!\}$, $\{\!|Q, \sigma|\!\}$, and *eqs*, it requires the existence of $\{\!|Q', \sigma'|\!\}$ and $\{\!|Q'', \sigma''|\!\}$ satisfying the conditions mentioned in 6 (b), *even if* $[\![\mathtt{proj0}[b](\mathcal{E}[\mathrm{qv}(\rho) - \mathrm{qv}(P)](\rho))]\!] = O$ *holds*, which means the probability of this transition is 0. As for this case, in fact, only the existence of $\{\!|Q'', \sigma''|\!\}$ is necessary in the proof of the soundness but that of $\{\!|Q', \sigma'|\!\}$ is not. Therefore, the condition that Verifier1 returns *true* with $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ is stronger than the condition that the two qCCS configurations corresponding to them are bisimilar.

Memoization

We also employ a memoization technique. Let $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ have the transitions $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ and $\{\!|Q, \sigma|\!\} \xrightarrow{\alpha} \{\!|Q', \sigma'|\!\}$, and assume $\{\!|P', \rho'|\!\} \approx \{\!|Q', \sigma'|\!\}$. When checking whether $\{\!|P, \rho|\!\} \approx \{\!|Q, \sigma|\!\}$ holds, Verifier1 first checks $\{\!|Q, \sigma|\!\}$ simulates $\{\!|P, \rho|\!\}$'s transition. For $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$, Verifier1 finds $\{\!|Q, \sigma|\!\} \xrightarrow{\alpha} \{\!|Q', \sigma'|\!\}$, and then recursively checks $\{\!|P', \rho'|\!\} \approx \{\!|Q', \sigma'|\!\}$. Verifier1 then checks $\{\!|P, \rho|\!\}$ simulates $\{\!|Q, \sigma|\!\}$'s transition. For the transition $\{\!|Q, \sigma|\!\} \xrightarrow{\alpha} \{\!|Q', \sigma'|\!\}$, it finds the transition $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$, and next checks $\{\!|Q', \sigma'|\!\} \approx \{\!|P', \rho'|\!\}$. Since $\approx$ is a symmetric relation, the last condition has been already obtained when checking $\{\!|P', \rho'|\!\} \approx \{\!|Q', \sigma'|\!\}$. Verifier1 reuses the result.

3.5 Soundness of Verifier1

Verifier1 is designed to be sound in the following sense. For arbitrarily-fixed security parameters λ, if Verifier1 returns *true* for two configurations $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\} \in \mathcal{C}$, and some valid user-defined equations, then the corresponding two configurations that are elements of *Con* are bisimilar in the original qCCS. Our goal is to prove Theorem 3.5.11 that states the correspondence formally. In this section, we prepare lemmas to prove it. In the following arguments, let security parameters λ be arbitrarily fixed.

Suppose $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ holds. We say the transition $\xrightarrow{\alpha}$ is caused by *rule's name*, where *rule's name* is either (In), (Out), (Op), (Meas1), or (Meas0), if the derivation tree begins with the application of the rule and (Comm) rule is

not used. If (Comm) rule is used, we say the transition is caused by (Comm). We first prepare the notation to focus on a part of a process that causes a transition.

Definition 3.5.1. *The evaluation contexts are defined as follows.*

$$C[\_] ::= \_ \mid C[\_]\|P \mid P\|C[\_] \mid C[\_]\backslash L$$

Lemma 3.5.2. *If* $\{\!|P, \rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|P', \rho'|\!\}$*, then* $P = C[\mathsf{c}!q.P_0]$ *and* $P' = C[P_0]$ *hold for some process* P_0 *and evaluation context* $C[\_]$ *that does not restrict* c*.*

Proof. We prove it by induction of the number n of application of the transition rules.
(Case 1) Assume $n = 1$. The only rule to derive $\xrightarrow{\mathsf{c}!q}$ with one time application is (Out). Therefore, $P = \mathsf{c}!q.P_0$ holds.
(Case 2) Assume $n > 1$. The last rule applied is either (Res), (Right), or (Left). We prove the case of (Res) as other cases are similar. The last derivation is

$$\frac{\{\!|P_1, \rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|P_1', \rho|\!\} \quad \mathsf{c} \notin L}{\{\!|P_1\backslash L, \rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|P_1'\backslash L, \rho|\!\}},$$

where $P = P_1\backslash L$ and $P' = P_1'\backslash L$ hold, for some L. By I.H., $P_1 = C[\mathsf{c}!q.P_2]$ and $P_1' = C[P_2]$ for some $C[\_]$ and P_2. We take an evaluation context $C'[\_] = C[\_]\backslash L$. As $\mathsf{c} \notin L$, $C'[\_]$ does not restrict c. We then have $P = C'[\mathsf{c}!q.P_2]$ and $P' = C'[P_2]$. □

Lemma 3.5.2 is for transitions caused by (Out). Transitions caused by (In), (Op), (Meas$i$), and (Comm) have a similar property since the derivations start from those and proceed by applying (Left), (Right), (Res) rules. The original qCCS also has a similar property.

Next, we define the correspondence of processes in $\mathcal{P}$ and those in *Proc*, where $\mathcal{P}$ is the set of the processes of nondeterministic qCCS and *Proc* is the set of the processes of original qCCS.

Definition 3.5.3. *The function* $\mathrm{cnv} : \mathcal{P} \to Proc$ *is inductively defined as follows.*

$$\begin{aligned}
&\mathrm{cnv}(\texttt{discard}[\tilde{q}]) = \texttt{if false then } I[\tilde{q}].\texttt{nil fi}\\
&\mathrm{cnv}(\mathsf{c}!q.P) = \mathsf{c}!q.\mathrm{cnv}(P)\\
&\mathrm{cnv}(\mathsf{c}?q.P) = \mathsf{c}?q.\mathrm{cnv}(P)\\
&\mathrm{cnv}(op[\tilde{q}].P) = \overline{op}[\tilde{q}].\mathrm{cnv}(P), \text{ where } \mathcal{E}^{\overline{op}} = [\![op]\!]\\
&\mathrm{cnv}(\texttt{meas } b \texttt{ then } P \texttt{ saem}) = |1\rangle\langle 1|[b; x].\texttt{if } x = 1 \texttt{ then } \mathrm{cnv}(P) \texttt{ fi}\\
&\mathrm{cnv}(P||Q) = \mathrm{cnv}(P)||\mathrm{cnv}(Q)\\
&\mathrm{cnv}(P\backslash L) = \mathrm{cnv}(P)\backslash L
\end{aligned}$$

For an evaluation context $C[\_]$*,* $\mathrm{cnv}(C)[\_]$ *is the context of original qCCS process obtained applying* cnv *to all processes in* $C[\_]$*.*

By the definition, we have the following proposition.

Proposition 3.5.4. $\mathrm{qv}(P) = \mathrm{qv}(\mathrm{cnv}(\mathrm{P}))$ *holds. If* $\mathrm{cnv}(P) = \mathrm{cnv}(Q)$*, then* $P = Q$*.*

We then prove lemmas that state correspondence of original and Verifier1's frameworks.

Lemma 3.5.5. $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ *and* $\mathrm{tr}(\llbracket\rho\rrbracket) = \mathrm{tr}(\llbracket\rho'\rrbracket)$ *hold, then*

$$(\!|\mathrm{cnv}(P), \frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}|\!) \xrightarrow{\alpha} \mu \text{ and } \mu(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P'), \frac{\llbracket\rho'\rrbracket}{\mathrm{tr}(\llbracket\rho'\rrbracket)}|\!)$$

hold for some $\mu \in D(Con)$.

Proof. (α is $\mathsf{c}!q$) By lemma 3.5.2, $P = C[\mathsf{c}!q.P_0]$ and $P' = C[P_0]$ holds for some evaluation context $C[\_]$ and process P_0. Since

$$\mathrm{cnv}(P) = \mathrm{cnv}(C[\mathsf{c}!q.P_0]) = \mathrm{cnv}(C)[\mathsf{c}!q.\mathrm{cnv}(P_0)] \text{ and } \mathrm{cnv}(P') = \mathrm{cnv}(C[P_0])$$

hold, $(\!|\mathrm{cnv}(P), \frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}|\!) \xrightarrow{\mathsf{c}!q} (\!|\mathrm{cnv}(P'), \frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}|\!)$ holds. The conclusion of the lemma holds because identity is a strong bisimulation.
(α is $\mathsf{c}?q$ or τ caused by Comm) Similar to the above case.
(α is τ caused by Op) Similar to the above cases except that the quantum state changes. The correctness of the statement is checked observing that $\rho' = op[\tilde{r}](\rho)$ for some $op[\tilde{q}]$and $\mathcal{E}^{op}_{\tilde{r}}(\frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}) = \frac{\llbracket\rho'\rrbracket}{\mathrm{tr}(\llbracket\rho'\rrbracket)}$ holds because $\mathcal{E}^{op}$ is trace-preserving.
(α is τ caused by Meas1) Similarly to lemma 3.5.2, $P = C[\mathtt{meas}\ b\ \mathtt{then}\ P_0\ \mathtt{saem}]$ and $P' = C[P_0]$ holds for some evaluation context $C[\_]$ and process P_0. By $\mathrm{tr}(\llbracket\rho\rrbracket) = \mathrm{tr}(\llbracket\rho'\rrbracket) = \mathrm{tr}(\llbracket\mathtt{proj1}[b](\rho)\rrbracket) = \mathrm{tr}(|1\rangle\langle 1|_b\llbracket\rho\rrbracket)$,

$$(\!|\mathrm{cnv}(C)[|1\rangle\langle 1|[b; x].\mathtt{if}\ x = 1\ \mathtt{then}\ \mathrm{cnv}(P_0)\ \mathtt{fi}], \frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}|\!) \xrightarrow{\tau}$$
$$1 \bullet (\!|\mathrm{cnv}(C)[\mathtt{if}\ 1 = 1\ \mathtt{then}\ \mathrm{cnv}(P_0)\ \mathtt{fi}], \frac{|1\rangle\langle 1|_b\llbracket\rho\rrbracket}{\mathrm{tr}(|1\rangle\langle 1|_b\llbracket\rho\rrbracket)}|\!)$$

holds. Since

$$(\!|\mathtt{if}\ 1 = 1\ \mathtt{then}\ \mathrm{cnv}(P_0)\ \mathtt{fi}, \frac{|1\rangle\langle 1|_b\llbracket\rho\rrbracket|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b\llbracket\rho\rrbracket)}|\!)(\doteq)^\dagger(\!|\mathrm{cnv}(P_0), \frac{|1\rangle\langle 1|_b\llbracket\rho\rrbracket|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b\llbracket\rho\rrbracket)}|\!)$$

and $|1\rangle\langle 1|_b\llbracket\rho\rrbracket|1\rangle\langle 1|_b = \llbracket\rho'\rrbracket$ hold,

$$1\bullet(\!|\mathrm{cnv}(C)[\mathtt{if}\ 1 = 1\ \mathtt{then}\ \mathrm{cnv}(P_0)\ \mathtt{fi}], \frac{|1\rangle\langle 1|_b\llbracket\rho\rrbracket|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b\llbracket\rho\rrbracket)}|\!)(\doteq)^\dagger 1\bullet(\!|\mathrm{cnv}(P'), \frac{\llbracket\rho'\rrbracket}{\mathrm{tr}(\llbracket\rho'\rrbracket)}|\!).$$

holds by the congruence of $\doteq$ (Proposition 3.1.20).
(α is τ caused by Meas0) Similar to the case of Meas1. □

Lemma 3.5.6. $\{\!|P, \rho|\!\} \xrightarrow{\tau^*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau^*} \{\!|P', \rho'|\!\}$ *and* $\mathrm{tr}(\llbracket\rho\rrbracket) = \mathrm{tr}(\llbracket\rho'\rrbracket)$ *holds, then* $(\!|\mathrm{cnv}(P), \frac{\llbracket\rho\rrbracket}{\mathrm{tr}(\llbracket\rho\rrbracket)}|\!) \Rightarrow (\xrightarrow{\hat{\alpha}})^\dagger \Rightarrow \mu$ *and* $\mu(\doteq)^\dagger(\!|\mathrm{cnv}(P'), \frac{\llbracket\rho'\rrbracket}{\mathrm{tr}(\llbracket\rho'\rrbracket)}|\!)$ *holds for some* $\mu \in D(Con)$.

Proof. By assumption, we have

- $\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|P_1, \rho_1|\!\} \xrightarrow{\tau} \cdots \xrightarrow{\tau} \{\!|P_k, \rho_k|\!\} \xrightarrow{\hat{\alpha}} \{\!|\hat{P}, \hat{\rho}|\!\} \xrightarrow{\tau} \{\!|P'_1, \rho'_1|\!\} \xrightarrow{\tau} \cdots \xrightarrow{\tau} \{\!|P'_m, \rho'_m|\!\}$,
- $\mathrm{tr}(\llbracket\rho\rrbracket) = \mathrm{tr}(\llbracket\rho_1\rrbracket) = \cdots = \mathrm{tr}(\llbracket\rho'\rrbracket)$, and
- $\{\!|P'_m, \rho'_m|\!\} = \{\!|P', \rho'|\!\}$

for some k, m, P_1,...,P_k,$\hat{P}$,P'_1,...,P'_m ρ_1,...,ρ_k,$\hat{\rho}$,ρ'_1,...,ρ'_m. By $\{\!|P,\rho|\!\} \xrightarrow{\tau} \{\!|P_1,\rho_1|\!\}$ and $\mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\rho_1]\!])$ and the previous lemma,

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\tau} \mu_1(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!)$$

holds for some $\mu_1 \in D(Con)$. Next, we prove for all $i (1 \leq i \leq k-1)$ that $\{\!|P_i,\rho_i|\!\} \xrightarrow{\tau} \{\!|P_{i+1},\rho_{i+1}|\!\}$ and $\mathrm{tr}([\![\rho_i]\!]) = \mathrm{tr}([\![\rho_{i+1}]\!])$ and

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_i(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_i), \frac{[\![\rho_i]\!]}{\mathrm{tr}([\![\rho_i]\!])}|\!) \text{ for some } \mu_i$$

imply

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_{i+1}(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_{i+1}), \frac{[\![\rho_{i+1}]\!]}{\mathrm{tr}([\![\rho_{i+1}]\!])}|\!) \text{ for some } \mu_{i+1}.$$

By $\{\!|P_i,\rho_i|\!\} \xrightarrow{\tau} \{\!|P_{i+1},\rho_{i+1}|\!\}$ and $\mathrm{tr}([\![\rho_i]\!]) = \mathrm{tr}([\![\rho_{i+1}]\!])$ and the previous lemma, we have

$$(\!|\mathrm{cnv}(P_i), \frac{[\![\rho_i]\!]}{\mathrm{tr}([\![\rho_i]\!])}|\!) \Rightarrow \mu'_{i+1}(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_{i+1}), \frac{[\![\rho_{i+1}]\!]}{\mathrm{tr}([\![\rho_{i+1}]\!])}|\!) \text{ for some } \mu'_{i+1}.$$

By $\mu_i(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_i), \frac{[\![\rho_i]\!]}{\mathrm{tr}([\![\rho_i]\!])}|\!)$ and $(\!|\mathrm{cnv}(P_i), \frac{[\![\rho_i]\!]}{\mathrm{tr}([\![\rho_i]\!])}|\!) \Rightarrow \mu'_{i+1}$, $\mu_i \Rightarrow \mu_{i+1}$ and $\mu_{i+1}(\doteq)^\dagger \mu'_{i+1}$ holds for some μ_{i+1}. We then have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_i \Rightarrow \mu_{i+1}(\doteq)^\dagger \mu'_{i+1}(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_{i+1}), \frac{[\![\rho_{i+1}]\!]}{\mathrm{tr}([\![\rho_{i+1}]\!])}|\!),$$

namely,

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_{i+1}(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_{i+1}), \frac{[\![\rho_{i+1}]\!]}{\mathrm{tr}([\![\rho_{i+1}]\!])}|\!) \text{ for some } \mu_{i+1}.$$

Applying this argument repeatedly, we have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_k(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P_k), \frac{[\![\rho_k]\!]}{\mathrm{tr}([\![\rho_k]\!])}|\!) \text{ for some } \mu_k.$$

By the similar argument, we have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow (\xrightarrow{\hat{\alpha}})^\dagger \hat{\mu}(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(\hat{P}), \frac{[\![\hat{\rho}]\!]}{\mathrm{tr}([\![\hat{\rho}]\!])}|\!) \text{ for some } \hat{\mu}.$$

Furthermore, we have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow (\xrightarrow{\hat{\alpha}})^\dagger \Rightarrow \mu(\doteq)^\dagger 1 \bullet (\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!) \text{ for some } \mu.$$

□

Lemma 3.5.7. *If*

- $\{\!|P,\rho|\!\} = \{\!|C[\texttt{meas}\ b\ \texttt{then}\ P_0\ \texttt{saem}],\rho|\!\} \xrightarrow{\tau} \{\!|C[P_0],\texttt{proj1}[b](\rho)|\!\} \stackrel{\mathrm{def}}{=} \{\!|P',\rho'|\!\}$ *and*
- $\{\!|P,\rho|\!\} \xrightarrow{\tau} \{\!|C[\texttt{discard}(\mathrm{qv}(P_0))],\texttt{proj0}[b](\rho)|\!\} \stackrel{\mathrm{def}}{=} \{\!|P'',\rho''|\!\}$

hold, then

- $(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\tau} \mu$ *and*
- $\mu(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho']\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!) + \frac{\mathrm{tr}([\![\rho'']\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P''), \frac{[\![\rho'']\!]}{\mathrm{tr}([\![\rho'']\!])}|\!)$

hold for some $\mu \in D(Con)$.

Proof. We have

$$\begin{aligned}
&(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) = (\!|\mathrm{cnv}(C)[|1\rangle\langle 1|[b;x].\texttt{if } x = 1 \texttt{ then } \mathrm{cnv}(P_0) \texttt{ fi}], \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \\
&\xrightarrow{\tau} \frac{\mathrm{tr}(|0\rangle\langle 0|_b[\![\rho]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C)[\texttt{if } 0 = 1 \texttt{ then } \mathrm{cnv}(P_0) \texttt{ fi}], \frac{|0\rangle\langle 0|_b[\![\rho]\!]|0\rangle\langle 0|_b}{\mathrm{tr}(|0\rangle\langle 0|_b[\![\rho]\!])}|\!) \\
&\quad + \frac{\mathrm{tr}(|1\rangle\langle 1|_b[\![\rho]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C)[\texttt{if } 1 = 1 \texttt{ then } \mathrm{cnv}(P_0) \texttt{ fi}], \frac{|1\rangle\langle 1|_b[\![\rho]\!]|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b[\![\rho]\!])}|\!) \stackrel{\mathrm{def}}{=} \mu.
\end{aligned}$$

Besides, we have

$$\begin{aligned}
&(\!|\mathrm{cnv}(C)[\texttt{if } 0 = 1 \texttt{ then } \mathrm{cnv}(P_0) \texttt{ fi}], \frac{|0\rangle\langle 0|_b[\![\rho]\!]|0\rangle\langle 0|_b}{\mathrm{tr}(|0\rangle\langle 0|_b[\![\rho]\!])}|\!) \\
&\mathrel{\dot{\approx}} (\!|\mathrm{cnv}(C)[\texttt{if } 0 = 1 \texttt{ then } I[\mathrm{qv}(P_0)].\texttt{nil fi}], \frac{|0\rangle\langle 0|_b[\![\rho]\!]|0\rangle\langle 0|_b}{\mathrm{tr}(|0\rangle\langle 0|_b[\![\rho]\!])}|\!) = (\!|\mathrm{cnv}(P''), \frac{[\![\rho'']\!]}{\mathrm{tr}([\![\rho'']\!])}|\!),
\end{aligned}$$

and

$$\begin{aligned}
&(\!|\mathrm{cnv}(C)[\texttt{if } 1 = 1 \texttt{ then } \mathrm{cnv}(P_0) \texttt{ fi}], \frac{|1\rangle\langle 1|_b[\![\rho]\!]|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b[\![\rho]\!])}|\!) \\
&\mathrel{\dot{\approx}} (\!|\mathrm{cnv}(C)[\mathrm{cnv}(P_0)], \frac{|1\rangle\langle 1|_b[\![\rho]\!]|1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b[\![\rho]\!])}|\!) = (\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!).
\end{aligned}$$

We have the conclusion of the lemma by the linearity of $(\mathrel{\dot{\approx}})^\dagger$. □

Lemma 3.5.8. *If*

- $\{|P, \rho|\} \xrightarrow{\tau*} \{|C[\texttt{meas } b \texttt{ then } P' \texttt{ saem}], \rho'|\} \xrightarrow{\tau} \{|C[P'], \texttt{proj1}[b](\rho')|\} \xrightarrow{\tau*} \{|P_1, \rho_1|\}$,
- $\{|C[\texttt{meas } b \texttt{ then } P' \texttt{ saem}], \rho'|\} \xrightarrow{\tau} \{|C[\texttt{discard}(\mathrm{qv}(P'))], \texttt{proj0}[b](\rho')|\} \xrightarrow{\tau*} \{|P_0, \rho_0|\}$, *and*
- $\mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\rho']\!]) = \mathrm{tr}([\![\rho_0]\!]) + \mathrm{tr}([\![\rho_1]\!])$

hold, then

- $(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu$ *and*
- $\mu(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho_0]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_0), \frac{[\![\rho_0]\!]}{\mathrm{tr}([\![\rho_0]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!)$

hold for some $\mu \in D(Con)$.

Proof. By the same argument as the proof of Lemma 3.5.6, we have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_0(\mathrel{\dot{\approx}})^\dagger 1 \bullet (\!|\mathrm{cnv}(C[\texttt{meas } b \texttt{ then } P' \texttt{ saem}]), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!)$$

for some μ_0. By the previous lemma, we have

$$1 \bullet (\!|\mathrm{cnv}(C[\mathtt{meas}\ b\ \mathtt{then}\ P'\ \mathtt{saem}]), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!) \Rightarrow \mu_1, \text{ and}$$
$$\mu_1(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C[\mathtt{discard}(\mathrm{qv}(P'))]), \frac{[\![\mathtt{proj0}[b](\rho')]\!]}{\mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!])}|\!)$$
$$+ \frac{\mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C[P']), \frac{[\![\mathtt{proj1}[b](\rho')]\!]}{\mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!])}|\!).$$

for some μ_1. By

$$\mathrm{tr}([\![\rho']\!]) = \mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!]) + \mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!]),$$
$$\mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!]) \geq \mathrm{tr}([\![\rho_0]\!]), \text{ and}$$
$$\mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!]) \geq \mathrm{tr}([\![\rho_1]\!]),$$

we have $\mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!]) = \mathrm{tr}([\![\rho_0]\!])$ and $\mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!]) = \mathrm{tr}([\![\rho_1]\!])$. Let $\frac{\mathrm{tr}([\![\rho_0]\!])}{\mathrm{tr}([\![\rho]\!])} \stackrel{\mathrm{def}}{=} p_0$ and $\frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \stackrel{\mathrm{def}}{=} p_1$. Now, we apply the same argument as the proof of Lemma 3.5.6 to each configuration. We have

$$X \stackrel{\mathrm{def}}{=} (\!|\mathrm{cnv}(C[\mathtt{discard}(\mathrm{qv}(P'))]), \frac{[\![\mathtt{proj0}[b](\rho')]\!]}{\mathrm{tr}([\![\mathtt{proj0}[b](\rho')]\!])}|\!) \Rightarrow \mu_2(\mathrel{\dot{\approx}})^\dagger (\!|\mathrm{cnv}(P_0), \frac{[\![\rho_0]\!]}{\mathrm{tr}([\![\rho_0]\!])}|\!)$$

and

$$Y \stackrel{\mathrm{def}}{=} (\!|\mathrm{cnv}(C[P']), \frac{[\![\mathtt{proj1}[b](\rho')]\!]}{\mathrm{tr}([\![\mathtt{proj1}[b](\rho')]\!])}|\!) \Rightarrow \mu_3(\mathrel{\dot{\approx}})^\dagger (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!)$$

for some μ_2 and μ_3. We then have

$$\mu_1(\mathrel{\dot{\approx}})^\dagger p_0 \bullet X + p_1 \bullet Y \Rightarrow p_0\mu_2 + p_1\mu_3 \quad (\sharp) \quad \text{and}$$
$$p_0\mu_2 + p_1\mu_3(\mathrel{\dot{\approx}})^\dagger p_0 \bullet (\!|\mathrm{cnv}(P_0), \frac{[\![\rho_0]\!]}{\mathrm{tr}([\![\rho_0]\!])}|\!) + p_1 \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!).$$

By $(\sharp)$, we have $\mu_1 \Rightarrow \mu$ and $\mu(\mathrel{\dot{\approx}})^\dagger p_0\mu_2 + p_1\mu_3$ for some μ. Therefore,

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \Rightarrow \mu_0 \Rightarrow \mu_1 \Rightarrow \mu \text{ and}$$
$$\mu(\mathrel{\dot{\approx}})^\dagger p_0 \bullet (\!|\mathrm{cnv}(P_0), \frac{[\![\rho_0]\!]}{\mathrm{tr}([\![\rho_0]\!])}|\!) + p_1 \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!)$$

hold. □

Lemma 3.5.9. *If* $(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\alpha} \mu$ *holds and* μ *is a point distribution, then* $\{\!|P, \rho|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ *and* $\mu(\mathrel{\dot{\approx}})^\dagger 1 \bullet (\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!)$ *for some* $\{\!|P', \rho'|\!\}$.

Proof. (α is $\mathsf{c}!q$) There exist a qCCS's evaluation context $D[\_]$ that does not restrict c and process $\tilde{P} \in \mathit{Proc}$ such that $\mathrm{cnv}(P) = D[\mathsf{c}!q.\tilde{P}]$. There exist an evaluation context $C[\_]$ of simplified processes not restricting c and $P_0 \in \mathcal{P}$ such that $D[\_] = \mathrm{cnv}(C)[\_]$ and $\tilde{P} = \mathrm{cnv}(P_0)$. Therefore, $\mathrm{cnv}(P) = \mathrm{cnv}(C[\mathsf{c}!q.P_0])$ holds. By Proposition 3.5.4, $P = C[\mathsf{c}!q.P_0]$ holds. We then have $\{\!|P, \rho|\!\} \xrightarrow{\mathsf{c}!q} \{\!|C[P_0], \rho|\!\}$. We also have

$$(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\mathsf{c}!q} (\!|D[\tilde{P}], \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) = (\!|\mathrm{cnv}(C[P_0]), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!)$$

(α is $\mathtt{c}?q$) This case is similar to the above case.
(α is τ caused by application of a TPCP map or communication) These cases are also similar to the case of $\mathtt{c}!q$.
(α is τ caused by measurement) We assume the result of the measurement is 1 with probability 1. The argument of the other case is similar. We omit the similar argument as that in the case of $\mathtt{c}!q$. We have

- $P = C[\mathtt{meas}\ b\ \mathtt{then}\ P_0\ \mathtt{saem}]$,
- $(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\tau} (\!|D[\mathtt{if}\ 1 = 1\ \mathtt{then}\ \tilde{P}\ \mathtt{fi}], \frac{|1\rangle\langle 1|_b [\![\rho]\!] |1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b [\![\rho]\!])}|\!)$
- $(\!|D[\mathtt{if}\ 1 = 1\ \mathtt{then}\ \tilde{P}\ \mathtt{fi}], \frac{|1\rangle\langle 1|_b [\![\rho]\!] |1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b [\![\rho]\!])}|\!) (\mathrel{\dot{\approx}})^\dagger (\!|D[\tilde{P}], \frac{|1\rangle\langle 1|_b [\![\rho]\!] |1\rangle\langle 1|_b}{\mathrm{tr}(|1\rangle\langle 1|_b [\![\rho]\!])}|\!)$, and
- $D[\tilde{P}] = \mathrm{cnv}(C[P_0])$

for some $C[\_], b$, P_0, $D[\_]$, and $\tilde{P}$. □

Lemma 3.5.10. *If* $(\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!) \xrightarrow{\alpha} \mu$ *holds and* μ *is not a point distribution, then*

1. α *is* τ,
2. $\{\!|P, \rho|\!\} = \{\!|C[\mathtt{meas}\ b\ \mathtt{then}\ P'\ \mathtt{saem}], \rho|\!\}$ *for some evaluation context* $C[\_]$, *qubit variable* b, *process* P',
3. $\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|C[\mathtt{discard}(\mathrm{qv}(P'))], \rho_1|\!\} \stackrel{\mathrm{def}}{=} \{\!|P_1, \rho_1|\!\}$,
4. $\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|C[P'], \rho_2|\!\} \stackrel{\mathrm{def}}{=} \{\!|P_2, \rho_2|\!\}$, *and*
5. $\mu(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_2), \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!)$

Proof. Since μ is not a point distribution, the transition is caused by measurement (1). Therefore, we have $P = C[\mathtt{meas}\ b\ \mathtt{then}\ P'\ \mathtt{saem}]$ for some evaluation context $C[\_]$, qubit variable b, and process P' (2). We have (3) and (4) immediately. We also have

$$\begin{aligned}
\mu =& \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C)[\mathtt{if}\ 0 = 1\ \mathtt{then}\ \mathrm{cnv}(P')\ \mathtt{fi}], \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!) \\
&+ \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(C)[\mathtt{if}\ 1 = 1\ \mathtt{then}\ \mathrm{cnv}(P')\ \mathtt{fi}], \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!) \\
(\mathrel{\dot{\approx}})^\dagger & \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!) \\
&+ \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_2), \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!).
\end{aligned}$$

□

3.5.1 The Correspondence

The following theorem states the soundness of Verifier1.

Theorem 3.5.11. *If Verifier1 returns true with the input* $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\} \in \mathcal{P} \times \mathcal{S}$ *satisfying* $\mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\sigma]\!]) = 1$, *and a set of valid equations eqs, then* $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!) \approx (\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ *holds.*

Proof. We assume that Verifier1 uses a simplified algorithm without the step 2 in which TPCP maps are performed without any transition. The theorem is still proven to hold with the step 2 extending the proof.

Assume all equations in *eqs* are valid. We define

$$\mathcal{R}_{eqs} := \left\{ (X,Y) \;\middle|\; \begin{array}{l} X \dot{\approx} (\!|\mathrm{cnv}(P), \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}|\!), Y \dot{\approx} (\!|\mathrm{cnv}(Q), \frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])}|\!), \text{ and} \\ \text{Verifier1 returns } \textit{true} \text{ with } \{\!|P,\rho|\!\} \text{ and } \{\!|Q,\sigma|\!\} \text{ using } \textit{eqs}. \end{array} \right\}.$$

We then have $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)\mathcal{R}_{eqs}(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ if Verifier1 returns *true* with the input $\{\!|P,\rho|\!\}, \{\!|Q,\sigma|\!\} \in \mathcal{C}$, $\mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\sigma]\!]) = 1$, and *eqs*. It is sufficient to show that $\mathcal{R}_{eqs}$ is a weak bisimulation relation. Let (X,Y) be an arbitrary element in $\mathcal{R}_{eqs}$. The condition of quantum variable is satisfied by the definition of cnv(). The condition of partial trace is checked as follows.

$$\begin{aligned} \mathrm{tr}_{\mathrm{qv}(\mathrm{cnv}(P))}(\frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}) &= \frac{1}{\mathrm{tr}([\![\rho]\!])}[\![\mathtt{Tr}[\mathrm{qv}(P)](\rho)]\!] && \text{(by the definition of } [\![\cdot]\!]) \\ &= \frac{1}{\mathrm{tr}([\![\sigma]\!])}[\![\mathtt{Tr}[\mathrm{qv}(Q)](\sigma)]\!] && \text{(by validity of } \textit{eqs}) \\ &= \mathrm{tr}_{\mathrm{qv}(\mathrm{cnv}(Q))}(\frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])}) && \text{(by the definition of } [\![\cdot]\!]) \end{aligned}$$

Next, we check the condition of simulation. Let $\mathcal{E}_{\tilde{r}}$ be an arbitrary TPCP map acting on $\tilde{r} \subseteq qVar - \mathrm{qv}(P)$. Assume $X \xrightarrow{\alpha} \mu$. By strong bisimulation, $(\!|\mathrm{cnv}(P), \mathcal{E}_{\tilde{r}}(\frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])})|\!) \xrightarrow{\alpha} \mu'$ and $\mu(\dot{\approx})^\dagger\mu'$ hold.

(Case 1) Assume μ' is a point distribution. By lemma 3.5.9, $\{\!|P, \bar{\mathcal{E}}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ interpreting $\bar{\mathcal{E}}$ as $\mathcal{E}_{\tilde{r}}$, and $\mu'(\dot{\approx})^\dagger(\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!)$ hold for some $\{\!|P',\rho'|\!\}$. Since Verifier1 returns *true*, there exists $\{\!|Q',\sigma'|\!\}$ such that $\{\!|Q, \bar{\mathcal{E}}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ holds and Verifier1 returns *true* with $\{\!|P',\rho'|\!\}$. This implies $\mathrm{tr}([\![\sigma']\!]) = \mathrm{tr}([\![\rho']\!]) = \mathrm{tr}([\![\bar{\mathcal{E}}[\tilde{r}](\rho)]\!]) = \mathrm{tr}([\![\bar{\mathcal{E}}[\tilde{r}](\sigma)]\!])$. Now, we can apply Lemma 3.5.6. We have

$$(\!|\mathrm{cnv}(Q), \mathcal{E}_{\tilde{r}}(\frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])})|\!) \Rightarrow\xrightarrow{\hat{\alpha}}\Rightarrow \nu'(\dot{\approx})^\dagger 1 \bullet (\!|\mathrm{cnv}(Q'), \frac{[\![\sigma']\!]}{\mathrm{tr}([\![\sigma']\!])}|\!)$$

for some ν. Next, by strong bisimulation, $Y \Rightarrow\xrightarrow{\hat{\alpha}}\Rightarrow \nu$ and $\nu'(\dot{\approx})^\dagger\nu$. We then have

$$\mu(\dot{\approx})^\dagger(\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!) \text{ and } \nu(\dot{\approx})^\dagger(\!|\mathrm{cnv}(Q'), \frac{[\![\sigma']\!]}{\mathrm{tr}([\![\sigma']\!])}|\!).$$

By the definition of $(\cdot)^\dagger$, μ can be written as $\sum_i p_i X'_i$ and $X'_i \dot{\approx} (\!|\mathrm{cnv}(P'), \frac{[\![\rho']\!]}{\mathrm{tr}([\![\rho']\!])}|\!)$ for all i. Similarly, ν can be written as $\sum_j q_j Y'_j$ and $Y'_j \dot{\approx} (\!|\mathrm{cnv}(Q'), \frac{[\![\sigma']\!]}{\mathrm{tr}([\![\sigma']\!])}|\!)$ for all j. Since Verifier1 returns *true* with $\{\!|P',\rho'|\!\}$ and $\{\!|Q',\sigma'|\!\}$, $X_i\mathcal{R}_{eqs}Y_j$ for all i,j. holds. Therefore, $\sum_{i,j} p_i q_j X_i \mathcal{R}^\dagger_{eqs} \sum_{i,j} p_i q_j Y_j$ holds. This is equivalent to $\mu\mathcal{R}^\dagger_{eqs}\nu$.

(Case 2) Assume μ' is not a point distribution. By lemma 3.5.10, $\{\!|P, \bar{\mathcal{E}}[\tilde{r}](\rho)|\!\} \xrightarrow{\tau} \{\!|P_1,\rho_1|\!\}$ and $\{\!|P, \bar{\mathcal{E}}[\tilde{r}](\rho)|\!\} \xrightarrow{\tau} \{\!|P_2,\rho_2|\!\}$ interpreting $\bar{\mathcal{E}}$ as $\mathcal{E}_{\tilde{r}}$, and

$$\mu'(\dot{\approx})^\dagger \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_2), \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!)$$

hold. Since Verifier1 returns *true*, there exists configurations $\{\!|Q_1,\sigma_1|\!\}$ and $\{\!|Q_2,\sigma_2|\!\}$ such that

- $\{\!|Q, \bar{\mathcal{E}}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \{\!|D[\texttt{meas}\ b\ \texttt{then}\ Q'\ \texttt{saem}], \sigma'|\!\}$,
- $\{\!|D[\texttt{meas}\ b\ \texttt{then}\ Q'\ \texttt{saem}], \sigma'|\!\} \xrightarrow{\tau} \{\!|D[Q'], \texttt{proj1}[b](\sigma')|\!\} \xrightarrow{\tau*} \{\!|Q_1, \sigma_1|\!\}$,
- $\{\!|D[\texttt{meas}\ b\ \texttt{then}\ Q'\ \texttt{saem}], \sigma'|\!\} \xrightarrow{\tau} \{\!|D[\texttt{discard}(\mathrm{qv}(Q'))], \texttt{proj0}[b](\sigma')|\!\}$ $\xrightarrow{\tau*} \{\!|Q_2, \sigma_2|\!\}$,

hold for some D, b, Q' and σ', and

- Verifier1 returns *true* with $\{\!|P_1, \rho_1|\!\}$, $\{\!|Q_1, \sigma_1|\!\}$, and *eqs*.
- Verifier1 returns *true* with $\{\!|P_2, \rho_2|\!\}$, $\{\!|Q_2, \sigma_2|\!\}$, and *eqs*.

Moreover, $\mathrm{tr}([\![\sigma]\!]) = \mathrm{tr}([\![\sigma']\!])$ holds; Otherwise, $\mathrm{tr}([\![\sigma]\!]) > \mathrm{tr}([\![\sigma']\!])$ holds. Since Verifier1 returns *true* with two pairs $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$), the numbers of $\texttt{proj}i$'s occurring in ρ and σ are equal (Section 3.4.2). Let the number be N. In ρ_1, there are $N+1$ $\texttt{proj}i$'s. By the transition, there are more than $N+2$ $\texttt{proj}i$'s or $N+2$ $\texttt{proj}i$'s in σ_1. This contradicts that Verifier1 returned *true* with $\{\!|P_1, \rho_1|\!\}$ and $\{\!|Q_1, \sigma_1|\!\}$, and thus the numbers of $\texttt{proj}i$'s in ρ_1 and σ_1 are equal.

Next, by the validity of *eqs*, $\mathrm{tr}([\![\rho_1]\!]) = \mathrm{tr}([\![\sigma_1]\!])$ and $\mathrm{tr}([\![\rho_2]\!]) = \mathrm{tr}([\![\sigma_2]\!])$ hold. Thus we have $\mathrm{tr}([\![\sigma_1]\!]) + \mathrm{tr}([\![\sigma_2]\!]) = \mathrm{tr}([\![\rho_1]\!]) + \mathrm{tr}([\![\rho_2]\!]) = \mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\sigma]\!]) = \mathrm{tr}([\![\sigma']\!])$. Now, we can apply the Lemma 3.5.8 to have

$$(\!|\mathrm{cnv}(Q), \frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])}|\!) \Rightarrow \nu'$$
$$\nu'(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(Q_1), \frac{[\![\sigma_1]\!]}{\mathrm{tr}([\![\sigma_1]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(Q_2), \frac{[\![\sigma_2]\!]}{\mathrm{tr}([\![\sigma_2]\!])}|\!)$$

for some ν'. By strong bisimulation, $Y \Rightarrow \nu$ and $\nu'(\mathrel{\dot{\approx}})^\dagger \nu$ hold for some ν. We then have

$$\mu(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(P_2), \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!) \stackrel{\mathrm{def}}{=} U$$
$$\nu(\mathrel{\dot{\approx}})^\dagger \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(Q_1), \frac{[\![\sigma_1]\!]}{\mathrm{tr}([\![\sigma_1]\!])}|\!) + \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])} \bullet (\!|\mathrm{cnv}(Q_2), \frac{[\![\sigma_2]\!]}{\mathrm{tr}([\![\sigma_2]\!])}|\!) \stackrel{\mathrm{def}}{=} V$$

Let A, B, C and D be $(\!|\mathrm{cnv}(P_1), \frac{[\![\rho_1]\!]}{\mathrm{tr}([\![\rho_1]\!])}|\!)$, $(\!|\mathrm{cnv}(P_2), \frac{[\![\rho_2]\!]}{\mathrm{tr}([\![\rho_2]\!])}|\!)$, $(\!|\mathrm{cnv}(Q_1), \frac{[\![\sigma_1]\!]}{\mathrm{tr}([\![\sigma_1]\!])}|\!)$, and $(\!|\mathrm{cnv}(Q_2), \frac{[\![\sigma_2]\!]}{\mathrm{tr}([\![\sigma_2]\!])}|\!)$, respectively. By the definition of $(\cdot)^\dagger$, μ is written as $\sum_{i\in I} p_i X_i + \sum_{j\in J} q_j X_j$ with $I \cap J = \emptyset$, and U is written as $\sum_{i\in I} p_i A + \sum_{j\in J} q_j B$, and $X_i \mathrel{\dot{\approx}} A$ for all $i \in I$, and $X_j \mathrel{\dot{\approx}} B$ for all $j \in J$. $\sum_{i\in I} p_i = \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])}$ and $\sum_{j\in J} q_j = \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])}$ hold. Similarly, ν is written as $\sum_{k\in K} p'_k Y_k + \sum_{l\in L} q'_l Y_l$ with $K \cap L = \emptyset$, and V is written as $\sum_{k\in K} p'_k C + \sum_{l\in L} q'_l D$, and $Y_k \mathrel{\dot{\approx}} C$ for all $k \in K$, and $Y_l \mathrel{\dot{\approx}} D$ for all $l \in L$. $\sum_{k\in K} p'_k = \frac{\mathrm{tr}([\![\rho_1]\!])}{\mathrm{tr}([\![\rho]\!])}$ and $\sum_{l\in L} q'_l = \frac{\mathrm{tr}([\![\rho_2]\!])}{\mathrm{tr}([\![\rho]\!])}$ hold. Since Verifier1 returns *true* with $(\{\!|P_m, \rho_m|\!\}, \{\!|Q_m, \sigma_m|\!\})\ (m = 1, 2)$, $X_i \mathcal{R}_{eqs} Y_k$ for all $i \in I, k \in K$, and $X_j \mathcal{R}_{eqs} Y_l$ for all $j \in J, l \in L$. We then have the conclusion $\mu(\mathcal{R}_{eqs})^\dagger \nu$ observing

$$\mu = \frac{\mathrm{tr}([\![\rho]\!])}{\mathrm{tr}([\![\rho_1]\!])} \sum_{i,k} p_i p'_k X_i + \frac{\mathrm{tr}([\![\rho]\!])}{\mathrm{tr}([\![\rho_2]\!])} \sum_{j,l} q_j q'_l X_j \text{ and}$$
$$\nu = \frac{\mathrm{tr}([\![\rho]\!])}{\mathrm{tr}([\![\rho_1]\!])} \sum_{i,k} p_i p'_k Y_k + \frac{\mathrm{tr}([\![\rho]\!])}{\mathrm{tr}([\![\rho_2]\!])} \sum_{j,l} q_j q'_l Y_l.$$

By definition of $\mathcal{R}$, $\mathcal{R}$ is symmetric and thus $\mathcal{R}^{-1}$ also satisfies the conditions. □

3.6 Discussion

3.6.1 On Completeness

Let us consider an "ideal" verifier that can test equality of partial traces perfectly. It takes two configurations $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$, but does not take a set of user-defined equations. In the step 4 of the algorithm, it goes to the next step if and only if $[\![\mathtt{Tr}[\mathrm{qv}(P)](\rho)]\!] = [\![\mathtt{Tr}[\mathrm{qv}(Q)](\sigma)]\!]$. Let us then consider the following statement. For $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\} \in \mathcal{P} \times \mathcal{S}$, if $\mathrm{tr}([\![\rho]\!]) = \mathrm{tr}([\![\sigma]\!]) = 1$ and $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!) \approx (\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ hold, then the ideal verifier returns *true* with the input $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\}$. We call this *the completeness of nondeterministic qCCS with respect to qCCS*. In fact, this statement is not true. A counter example is as follows.

$$\{\!|P, \rho|\!\} \stackrel{\mathrm{def}}{=} \{\!|\mathtt{meas}\ b\ \mathtt{then}\ \mathtt{discard}(b, q)\ \mathtt{saem}, |\psi\rangle\langle\psi|_b \otimes |0\rangle\langle 0|_q \otimes \rho^E|\!\} \text{ and}$$
$$\{\!|Q, \sigma|\!\} \stackrel{\mathrm{def}}{=} \{\!|\mathtt{discard}(b, q), |\psi\rangle\langle\psi|_b \otimes |0\rangle\langle 0|_q \otimes \rho^E|\!\}, \text{ where}$$
$$|\psi\rangle = \sqrt{\frac{1}{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle.$$

The two transitions of $(\!|P, [\![\rho]\!]|\!)$ are

$$\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|\mathtt{discard}(b, q), \frac{1}{3}|0\rangle\langle 0|_b \otimes |0\rangle\langle 0|_q \otimes \rho^E|\!\} \text{ and}$$
$$\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|\mathtt{discard}(b, q), \frac{2}{3}|1\rangle\langle 1|_b \otimes |0\rangle\langle 0|_q \otimes \rho^E|\!\}$$

but partial traces of them ($\frac{1}{3}\rho^E$ and $\frac{2}{3}\rho^E$) are not equal to that of $\{\!|Q, \sigma|\!\}$ (namely ρ^E). In our simplified formal framework, two configurations $\{\!|P_0, \rho_0|\!\}$ and $\{\!|Q_0, \sigma_0|\!\}$ with $\mathrm{tr}(\rho_0) \neq \mathrm{tr}(\sigma_0)$ are always separated even if $(\!|\mathrm{cnv}(P_0), \frac{[\![\rho_0]\!]}{\mathrm{tr}([\![\rho_0]\!])}|\!)$ and $(\!|\mathrm{cnv}(Q_0), \frac{[\![\sigma_0]\!]}{\mathrm{tr}([\![\sigma_0]\!])}|\!)$ are identified. To identify such configurations, we must withdraw the simplification of operational semantics.

However, there seem to be a number of cases where we can assume that processes behave differently according to the result of quantum measurements. In general, we can formalize a QKD protocol of the form

$$...\mathtt{abort\_flag}[...,b,...].\mathtt{meas}\ b\ \mathtt{then}\ P'\ \mathtt{saem}...,$$

where an operator `abort_flag` calculates the number of errors in check bits and sets a bit b representing whether to abort the protocol. Alice and Bob continue to communicate only if the protocol has not aborted, which the outsider can recognize.

Besides, with our criteria discussed in Section 3.2.1, we expect that there is a transition of R with a label $\mathtt{c}!q$ or $\mathtt{c}?q$, when a process $\mathtt{meas}\ b\ \mathtt{then}\ R\ \mathtt{saem}$ is considered. Only for processes satisfying the condition, it is still possible that the completeness holds. It needs to be discussed precisely.

Chapter 4

Approximate Bisimulation for Quantum Processes

Two notions of approximate bisimulation are defined in the formal framework for the verifier (nondeterministic qCCS): the relations are on $\mathcal{C} = \mathcal{P} \times \mathcal{S}$ with simplified operational semantics. The relation $\sim_{\zeta,\eta}$ is defined in Section 4.2 and the relation $\sim$ is in Section 4.3. The extension of the verifier to check approximate bisimilarity is described in Section 4.4. Finally, application and limitations of approximate bisimulation are discussed in Section 4.5. Before the definitions, we introduce some preliminaries.

4.1 Preliminaries

4.1.1 Negligible Functions

Definition 4.1.1. *A function $f : \mathbb{N}_+ \to [0,1]$ is* negligible *if and only if for all polynomial $p(\cdot)$, there exists a natural number N such that $f(n) \leq \frac{1}{p(n)}$ holds for all $n \geq N$. f is* non-negligible *if f is not negligible.*

Remark 4.1.2. *A function $\frac{f}{g}(n) \stackrel{\text{def}}{=} \frac{f(n)}{g(n)}$ can be non-negligible even if f is negligible and g is non-negligible. For example, let $f(n) = \frac{1}{2^n}$ and*

$$g(n) = \begin{cases} \frac{1}{2^n} & (n \text{ is even}) \\ \frac{1}{n^2} & (\text{otherwise}). \end{cases}$$

In this thesis, we say g is greater than negligible *if the function $\frac{f}{g}$ is negligible for all negligible function f.*

Proposition 4.1.3. *If f and g are negligible, then $f + g$ and cf is negligible for all $c \geq 0$.*

4.1.2 Trace Distance of Probability-Weighted Quantum States

Trace distance is a metric on a set of linear operators. To compare quantum states, trace distance on $\mathcal{D}(\mathcal{H})$ is usually considered [60, Chapter 9]. Since we consider probability-weighted quantum states, we consider trace distance on $\Delta(\mathcal{H})$, and discuss an interpretation of it with respect to our transition system.

Let $\sqrt{A}$ be $\sum_i \sqrt{\lambda_i} P_i$ for an Hermitian operator A with the spectrum decomposition $\sum_i \lambda_i P_i$. Let $|A|$ be $\sqrt{A^\dagger A}$.

Definition 4.1.4. *Trace distance $d : \Delta(\mathcal{H}) \times \Delta(\mathcal{H}) \to [0,1]$ is defined as*

$$d(A,B) = \frac{1}{2}\mathrm{tr}|A - B|.$$

If $\mathrm{tr}(A) = \mathrm{tr}(B) = 1$, trace distance can be thought as a generalization of Kolmogorov distance. Indeed, if A and B are diagonal with respect to orthonormal basis $\{|i\rangle\}_i$, then $A = \sum_i p_i |i\rangle\langle i|$, $B = \sum_i q_i |i\rangle\langle i|$, and $d(A,B) = \frac{1}{2}\sum_i |p_i - q_i|$ hold for some unique p_i's and q_i's satisfying $\sum_i p_i = \sum_i q_i = 1$. We introduce some useful properties of trace distance as follows.

Proposition 4.1.5. *If* $\mathrm{tr}(A) = \mathrm{tr}(B) = 1$ *holds, then*

$$d(A,B) = \max\{\mathrm{tr}(\pi A) - \mathrm{tr}(\pi B) \,|\, \pi \text{ is a projector.}\}$$

holds.

Proposition 4.1.6. *If* $\mathrm{tr}(A) = \mathrm{tr}(B) = 1$ *holds, then* $d(\mathcal{E}(A), \mathcal{E}(B)) \leq d(A,B)$ *holds for all TPCP map* $\mathcal{E}$*.*

Proposition 4.1.6 can be extended to trace non-increasing cases. In the proof, we use the fact that $\mathrm{tr}|\cdot|$ is a norm on a set of linear operators.

Proposition 4.1.7. $d(\mathcal{E}(A), \mathcal{E}(B)) \leq d(A,B)$ *for all trace non-increasing positive map* $\mathcal{E}$*.*

Proof. Let $\sum_i \lambda_i |i\rangle\langle i|$ be an arbitrary eigenvalue decomposition of $A - B$. We obtain the proposition by the following calculation.

$$\begin{aligned}
\mathrm{tr}|\mathcal{E}(A-B)| &= \mathrm{tr}|\sum_i \lambda_i \mathcal{E}(|i\rangle\langle i|)| \\
&\leq \sum_i |\lambda_i| \cdot \mathrm{tr}|\mathcal{E}(|i\rangle\langle i|)| \quad \text{(by the axiom of trace norm)} \\
&\leq \sum_i |\lambda_i| \cdot \mathrm{tr}(|i\rangle\langle i|) \qquad (\mathcal{E} \text{ is trace non-increasing and positive.}) \\
&= \sum_i |\lambda_i| = \mathrm{tr}|A-B|
\end{aligned}$$

□

For a configuration $\{\!|P,\rho|\!\} \in \mathcal{C}$, $\mathrm{tr}([\![\rho]\!])$ is interpreted as the probability to reach $\{\!|P,\rho|\!\}$, and the quantum state is $\frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}$. The next proposition gives a way to interpret that $d([\![\rho]\!], [\![\sigma]\!])$ is small for two configurations $(\!|P,\rho|\!), (\!|Q,\sigma|\!) \in \mathcal{C}$.

Proposition 4.1.8. *Let* $A, B : \mathbb{N}_+ \to \Delta(\mathcal{H})$ *be functions and regard* $d(A,B) : \mathbb{N}_+ \to [0,1]$ *as a function.* $d(A,B)$ *is negligible iff* $|\mathrm{tr}(A) - \mathrm{tr}(B)|$ *and* $\mathrm{tr}(A) \cdot d(\frac{A}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(B)})$ *are negligible.*

Proof. ($\Rightarrow$) As $\mathrm{tr}(\cdot)$ is a TPCP map, $d(A,B) \geq d(\mathrm{tr}(A), \mathrm{tr}(B)) = \frac{1}{2}|\mathrm{tr}(A) - \mathrm{tr}(B)|$. $|\mathrm{tr}(A) - \mathrm{tr}(B)|$ is negligible by Proposition 4.1.3. $\mathrm{tr}(A) \cdot d(\frac{A}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(B)})$ is shown to be negligible by the following calculation.

$$\begin{aligned}
\mathrm{tr}(A) \cdot d(\frac{A}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(B)}) &\leq \mathrm{tr}(A) \cdot (d(\frac{A}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(A)}) + d(\frac{B}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(B)})) \\
&= d(A,B) + |\mathrm{tr}(A) - \mathrm{tr}(B)| \cdot \mathrm{tr}|\frac{B}{\mathrm{tr}(B)}| \\
&= d(A,B) + |\mathrm{tr}(A) - \mathrm{tr}(B)|
\end{aligned}$$

($\Leftarrow$) By triangle inequality, we have $d(A, \frac{\mathrm{tr}(A)}{\mathrm{tr}(B)}B) + d(\frac{\mathrm{tr}(A)}{\mathrm{tr}(B)}B, B) \geq d(A,B)$. This is equivalent to $\mathrm{tr}(A) \cdot d(\frac{A}{\mathrm{tr}(A)}, \frac{B}{\mathrm{tr}(B)}) + d(\frac{\mathrm{tr}(A)}{\mathrm{tr}(B)}B, B) \geq d(A,B)$. The left-hand side is shown to be negligible by the calculation $d(\frac{\mathrm{tr}(A)}{\mathrm{tr}(B)}B, B) = |\mathrm{tr}(A) - \mathrm{tr}(B)| \cdot \mathrm{tr}|\frac{B}{\mathrm{tr}(B)}|$. □

For $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\}$, assume $d([\![\rho]\!], [\![\sigma]\!])$ is negligible. It is equivalent to that $|\mathrm{tr}([\![\rho]\!]) - \mathrm{tr}([\![\sigma]\!])|$ and $\mathrm{tr}([\![\rho]\!]) \cdot d(\frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}, \frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])})$ are negligible. By Proposition 4.1.5, we have that $\mathrm{tr}([\![\rho]\!]) \cdot \mathrm{tr}(\pi \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}) - \mathrm{tr}([\![\rho]\!]) \cdot \mathrm{tr}(\pi \frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])})$ is negligible for all projector π. As $|\mathrm{tr}([\![\rho]\!])$ - $\mathrm{tr}([\![\sigma]\!])|$ is negligible, we have that $\mathrm{tr}([\![\rho]\!]) \cdot \mathrm{tr}(\pi \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])}) - \mathrm{tr}([\![\sigma]\!]) \cdot \mathrm{tr}(\pi \frac{[\![\sigma]\!]}{\mathrm{tr}([\![\sigma]\!])})$ is negligible. $\mathrm{tr}([\![\rho]\!]) \cdot \mathrm{tr}(\pi \frac{[\![\rho]\!]}{\mathrm{tr}([\![\rho]\!])})$ is the joint probability that a process reaches $\{\!|P, [\![\rho]\!]|\!\}$ and observes the measurement result corresponding to the projector π. Therefore, that $d(\mathrm{tr}([\![\rho]\!]), \mathrm{tr}([\![\sigma]\!]))$ is negligible implies that the difference of the joint probabilities is negligible for an arbitrary measurement.

4.2 Approximate Bisimulation with Parameters

We define the first approximate bisimulation relation that is parametrized by $\zeta, \eta \in [0, 1]$. To interpret symbolic representations as quantum states, security parameter λ must be fixed. In this section, a security parameter is arbitrarily fixed, while they are not in the next section (for the relation $\sim$).

In the following discussions, we may simply write $\mathrm{tr}(\rho)$ and $d(\rho, \sigma)$ as $\mathrm{tr}([\![\rho]\!]_\lambda)$ and $d([\![\rho]\!]_\lambda, [\![\sigma]\!]_\lambda)$ for $\rho, \sigma \in \mathcal{S}$ and fixed λ.

Definition 4.2.1. *Let $0 \leq \zeta, \eta \leq 1$. A symmetric relation $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is called an (ζ, η)-bisimulation if for all $\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\}$,$\{\!|P, \rho|\!\} \mathcal{R} \{\!|Q, \sigma|\!\}$ implies*

1. $\mathrm{qv}(P) = \mathrm{qv}(Q) \stackrel{\mathrm{def}}{=} \tilde{q}$,
2. $d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma)) \leq \zeta$, *and*
3. *For an arbitrary CP map $\mathcal{E}[\tilde{r}]$ acting on $\tilde{r} \subseteq qVar - \tilde{q}$, if $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold, then $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|P', \rho'|\!\} \mathcal{R} \{\!|Q', \sigma'|\!\}$ hold for some $\{\!|Q', \sigma'|\!\}$*

We call the conditions 1 and 2 the static conditions, and the condition 3 the simulation condition.

Precisely in the condition 3, it is possible that $[\![\mathcal{E}[\tilde{r}](\rho)]\!]_\lambda = O \notin \Delta(\mathcal{H})$ holds for some interpretation $[\![\mathcal{E}[\tilde{r}]]\!]_\lambda$. We exclude such CP maps in our discussions.

Remark 4.2.2. *An arbitrary* CP *map by the outsider is applied to the quantum state ρ in the condition 3 above, while a* TPCP *map is originally considered (Definition 3.1.13). We needed the assumption to prove that the relations are closed by parallel composition of processes. This assumption does not weaken the ability of the outsider, because CP is more general than TPCP. It also does not matter in verification in most cases, which is discussed in Section 4.4.3.*

Definition 4.2.3. *We define*

$$\sim_{\zeta,\eta} := \{(\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\}) \mid \{\!|P, \rho|\!\} \mathcal{R} \{\!|Q, \sigma|\!\} \text{ holds for some } (\zeta, \eta)\text{-bisimulation } \mathcal{R}\}.$$

We say $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ are (ζ, η)-bisimilar if $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$.

The relation $\sim_{\zeta,\eta}$ has the properties similar to the bisimulation relations defined in qCCS or other process calculi have [58].

Lemma 4.2.4. *$\sim_{\zeta,\eta}$ is a (ζ, η)-bisimulation.*

Proof. By definition of $\sim_{\zeta,\eta}$, it is symmetric. By $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$, there exists a (ζ,η)-bisimulation $\mathcal{R}$ satisfying $\{\!|P,\rho|\!\}\mathcal{R}\{\!|Q,\sigma|\!\}$. The static conditions are easily checked. Next, we have that for all CP map $\mathcal{E}[\tilde{r}]$ that acts on $\tilde{r} \subseteq qVar - \mathrm{qv}(P)$, if $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold, then there exists $\{\!|Q',\sigma'|\!\}$ satisfying $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\}\mathcal{R}\{\!|Q',\sigma'|\!\}$. This implies $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$. □

Lemma 4.2.5. *$\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ if and only if*

1. $\mathrm{qv}(P) = \mathrm{qv}(Q) \stackrel{\mathrm{def}}{=} \tilde{q}$,
2. $d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma)) \leq \zeta$,
3. *and for an arbitrary CP map $\mathcal{E}[\tilde{r}]$ acting on $\tilde{r} \subseteq qVar - \tilde{q}$,*
   - *if $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold, then $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|Q',\sigma'|\!\}$*
   - *if $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\alpha} \{\!|Q',\sigma'|\!\}$ and $\mathrm{tr}(\sigma') \geq \eta$ hold, then $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|P',\rho'|\!\}$*

Proof. ($\Rightarrow$) proven as the previous lemma.
($\Leftarrow$) We define $\hat{\mathcal{R}} := \sim_{\zeta,\eta} \cup \{(\{\!|P,\rho|\!\}, \{\!|Q,\sigma|\!\})\} \cup \{(\{\!|Q,\sigma|\!\}, \{\!|P,\rho|\!\})\}$. $\hat{\mathcal{R}}$ is symmetric by the definition. It is sufficient to show $\hat{\mathcal{R}}$ is a (ζ,η)-bisimulation relation. Let $(\{\!|P_0,\rho_0|\!\}, \{\!|Q_0,\sigma_0|\!\})$ be an arbitrary element of $\hat{\mathcal{R}}$.

1. Suppose $(\{\!|P_0,\rho_0|\!\}, \{\!|Q_0,\sigma_0|\!\}) \in \sim_{\zeta,\eta}$. Since $\sim_{\zeta,\eta}$ is a (ζ,η)-bisimulation, the static conditions are satisfied. Next, let $\mathcal{E}[\tilde{r}]$ be an arbitrary CP map acting on $\tilde{r} \subseteq qVar - \mathrm{qv}(P_0)$, and assume $\{\!|P_0,\mathcal{E}[\tilde{r}](\rho_0)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold. By the previous lemma, we have $\{\!|Q_0,\mathcal{E}[\tilde{r}](\sigma_0)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ for some $\{\!|Q',\sigma'|\!\}$. This implies $\{\!|Q_0,\mathcal{E}[\tilde{r}](\sigma_0)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\}\hat{\mathcal{R}}\{\!|Q',\sigma'|\!\}$ for some $\{\!|Q',\sigma'|\!\}$ since $\sim_{\zeta,\eta} \subseteq \hat{\mathcal{R}}$.
2. Suppose $(\{\!|P_0,\rho_0|\!\}, \{\!|Q_0,\sigma_0|\!\}) = (\{\!|P,\rho|\!\}, \{\!|Q,\sigma|\!\})$. The static conditions are easily checked. The simulation condition holds by the assumption and $\sim_{\zeta,\eta} \subseteq \hat{\mathcal{R}}$.
3. Suppose $(\{\!|P_0,\rho_0|\!\}, \{\!|Q_0,\sigma_0|\!\}) = (\{\!|Q,\sigma|\!\}, \{\!|P,\rho|\!\})$. The proof is similar to the previous case.

□

Proposition 4.2.6. *$\{\!|P||Q,\rho|\!\} \sim_{0,0} \{\!|Q||P,\sigma|\!\}$.*

Proof. This is proved by the definition of the transition rules, where (Left) and (Right) rules are symmetric. □

Lemma 4.2.7. *If $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ and $\{\!|P,\rho|\!\} \xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$, then $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|Q',\sigma'|\!\}$.*

Proof. Assume $\{\!|P,\rho|\!\} \xrightarrow{\tau}{}^{n} \{\!|P',\rho'|\!\}$ and let $\{\!|P_i,\rho_i|\!\}$ be i-th configuration with $0 \leq i \leq n$. The case when $n = 0$ is trivial. Let $n > 0$. Since $\rho = \rho_0 \geq \rho_1 \geq \cdots \geq \rho_n = \rho'$ and $\mathrm{tr}(\rho') \geq \eta$ hold, $\mathrm{tr}(\rho_i) \geq \eta$ holds for all i. Therefore, $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*}{}^{n} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|Q',\sigma'|\!\}$. □

Lemma 4.2.8. *If* $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ *and* $\{\!|P,\rho|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ *and* $\mathrm{tr}(\rho') \geq \eta$ *hold, then* $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ *and* $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$ *hold for some* $\{\!|Q',\sigma'|\!\}$.

Proof. It is proven similarly to the previous lemma. □

The relation $\sim_{\zeta,\eta}$ is closed under application of an arbitrary CP map by the outsider, namely, one acts on $qVar - \tilde{q}$. This is one of the similar properties as the original qCCS's largest bisimulation relation $\approx$ has [24].

Lemma 4.2.9. *If* $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$, *then* $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \sim_{\zeta,\eta} \{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\}$ *holds for all CP map* $\mathcal{E}$ *acting on* $\tilde{r} \subseteq qVar - \mathrm{qv}(P)$.

Proof. We use ($\Leftarrow$) implication of Lemma 4.2.5. By Lemma 4.1.7, $d(\mathcal{E}[\tilde{r}](\rho), \mathcal{E}[\tilde{r}](\sigma)) \leq d(\rho,\sigma)$ holds for all CP map $\mathcal{E}[\tilde{r}]$ and thus the static condition on partial trace holds. Since $\mathcal{E}[\tilde{r}]$ ranges over arbitrary CP map in the definition, the simulation condition holds. □

The relation $\sim_{\zeta,\eta}$ is reflexive and symmetric but not transitive because of the condition of trace distance. Instead, it has the following properties.

Proposition 4.2.10. *1. If* $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$, $\zeta \leq \zeta'$, *and* $\eta \leq \eta'$ *hold, then* $\{\!|P,\rho|\!\} \sim_{\zeta',\eta'} \{\!|Q,\sigma|\!\}$ *holds.*

2. If $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ *and* $\{\!|Q,\sigma|\!\} \sim_{\zeta',\eta'} \{\!|R,\theta|\!\}$ *hold, then*

$$\{\!|P,\rho|\!\} \sim_{\zeta+\zeta',\, \max\{\eta,\eta'\}+2(\zeta+\zeta')} \{\!|R,\theta|\!\}$$

holds.

Proof. 1. It is sufficient to prove that $\sim_{\zeta,\eta}$ is a (ζ',η')-bisimulation. The relation $\sim_{\zeta,\eta}$ is symmetric by Lemma 4.2.4. The static conditions is checked observing $d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma)) \leq \zeta \leq \zeta'$. For the simulation condition, assume $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta'$ for a CP map $\mathcal{E}[\tilde{r}]$. Since $\mathrm{tr}(\rho') \geq \eta' \geq \eta$, there exists a configuration $\{\!|Q',\sigma'|\!\}$ satisfying $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$.

2. We define a relation $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ as follows.

$$\begin{aligned}\mathcal{R}' :=& \{(\{\!|P,\rho|\!\}, \{\!|R,\theta|\!\}) \mid \{\!|P,\rho|\!\} \sim_{\zeta,\eta} \circ \sim_{\zeta',\eta'} \{\!|R,\theta|\!\}\} \\ \mathcal{R} :=& \mathcal{R}' \cup \mathcal{R}'^{-1}\end{aligned}$$

The assumption implies $\{\!|P,\rho|\!\}\mathcal{R}\{\!|R,\theta|\!\}$. It is sufficient to prove that $\mathcal{R}$ is a $(\zeta+\zeta', \max\{\eta,\eta'\}+2(\zeta+\zeta'))$-bisimulation. By definition, $\mathcal{R}$ is symmetric. **(Case 1)** Let an arbitrary element $(\{\!|P,\rho|\!\}, \{\!|R,\theta|\!\}) \in \mathcal{R}'$. There exists $\{\!|Q,\sigma|\!\}$ satisfying $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ and $\{\!|Q,\sigma|\!\} \sim_{\zeta',\eta'} \{\!|R,\theta|\!\}$. The static condition is checked observing $d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\theta)) \leq d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma)) + d(\mathrm{tr}_{\tilde{q}}(\sigma), \mathrm{tr}_{\tilde{q}}(\theta)) \leq \zeta+\zeta'$. For the simulation condition, assume $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \max\{\eta,\eta'\} + 2(\zeta+\zeta')$ for a CP map $\mathcal{E}[\tilde{r}]$. Since $\mathrm{tr}(\rho') \geq \eta$ and $\{\!|P,\rho|\!\} \sim_{\zeta,\eta} \{\!|Q,\sigma|\!\}$ hold, there exists $\{\!|Q',\sigma'|\!\}$ satisfying $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim_{\zeta,\eta} \{\!|Q',\sigma'|\!\}$. Applying Lemma 4.2.9 to $\{\!|Q,\sigma|\!\} \sim_{\zeta',\eta'} \{\!|R,\theta|\!\}$, we have $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \sim_{\zeta',\eta'} \{\!|R,\mathcal{E}[\tilde{r}](\theta)|\!\}$. We also have $\mathrm{tr}(\sigma') \geq \mathrm{tr}(\rho') - 2\zeta \geq \eta'$. By Lemma 4.2.8, there exists $\{\!|R',\theta'|\!\}$ satisfying $\{\!|R,\mathcal{E}[\tilde{r}](\theta)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|R',\theta'|\!\}$ and $\{\!|Q',\sigma'|\!\} \sim_{\zeta',\eta'}$

$\{\!|R', \theta'|\!\}$. We also have $\{\!|P', \rho'|\!\}\mathcal{R}'\{\!|R', \theta'|\!\}$.
(Case 2) Let an arbitrary element $(\{\!|R, \theta|\!\}, \{\!|P, \rho|\!\}) \in \mathcal{R}'^{-1}$. The proof is similar to the previous case.

□

Examples of the relation are as follows. Precisely, ρ is a symbolic representation of a quantum state for $\{\!|P, \rho|\!\} \in \mathcal{C}$ but we write a quantum state in the following examples for convenience. This is not harmful, because the security parameter is arbitrarily fixed in the setting in this section, and thus symbolic representations are interpreted to quantum states.

Example 4.2.11.

(1) $\{\!|\texttt{meas } b \texttt{ then c!}b\texttt{.discard() saem}, |+\rangle\langle+|_b \otimes |+\rangle\langle+|_q|\!\}$
$\sim_{\frac{1}{2},\frac{1}{2}} \{\!|\texttt{meas } b \texttt{ then c!}b\texttt{.discard() saem}, (\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)_{b,q}|\!\}$ *holds.*

(2) $\{\!|\texttt{meas } b \texttt{ then c!}q\texttt{.discard() saem}, |\psi\rangle\langle\psi|_b \otimes |0\rangle\langle 0|_q|\!\}$
$\sim_{0,\frac{1}{4}} \{\!|\texttt{meas } b \texttt{ then c!}q\texttt{.discard() saem}, |\psi\rangle\langle\psi|_b \otimes |1\rangle\langle 1|_q|\!\}$ *holds, where*

$$|\psi\rangle = \frac{\sqrt{3}|0\rangle + |1\rangle}{2}$$

(3) $\{\!|\texttt{discard}(), |+\rangle\langle+|_b|\!\} \sim_{\frac{1}{2},0} \{\!|\texttt{discard}(), |0\rangle\langle 0|_b|\!\}$ *holds.*

We prove that the relation $\sim_{\zeta,\eta}$ is closed under application of evaluation contexts on the condition $\eta > 2\zeta$. We first show that $\sim_{\zeta,\eta}$ is closed under restriction.

Lemma 4.2.12. *If* $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ *holds, then* $\{\!|P\backslash L, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q\backslash L, \sigma|\!\}$ *holds.*

Proof. Let $\mathcal{R} := \{(\{\!|P\backslash L, \rho|\!\}, \{\!|Q\backslash L, \sigma|\!\}) \mid \{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}\}$. It is sufficient to show that $\mathcal{R}$ is an approximate bisimulation relation. $\mathcal{R}$ is symmetric by the definition. Let $(\{\!|P\backslash L, \rho|\!\}, \{\!|Q\backslash L, \sigma|\!\})$ be an arbitrary element of $\mathcal{R}$. The static conditions are easily checked. Assume $\{\!|P\backslash L, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P'\backslash L, \rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$. This implies $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ and $\mathrm{cn}(\alpha) \cap L = \emptyset$. We have $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \sim \{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\}$ from $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$. We then have $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|P', \rho'|\!\} \sim_{\zeta,\eta} \{\!|Q', \sigma'|\!\}$ for some $\{\!|Q', \sigma'|\!\}$. As $\mathrm{cn}(\hat{\alpha}) \cap L = \emptyset$ and $\mathrm{cn}(\tau) = \emptyset$, we have $\{\!|Q\backslash L, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q'\backslash L, \sigma'|\!\}$ and $\{\!|P'\backslash L, \rho'|\!\}\mathcal{R}\{\!|Q'\backslash L, \sigma'|\!\}$. □

The relation $\sim_{\zeta,\eta}$ is not closed under parallel composition of processes for all ζ and η. For example, $\{\!|\texttt{discard}(), |+\rangle\langle+|_b|\!\} \sim_{\frac{1}{2},0} \{\!|\texttt{discard}(), |0\rangle\langle 0|_b|\!\}$ holds (Example 4.2.11) but

$$\{\!|\texttt{discard}()||\texttt{meas } b \texttt{ then c!}b\texttt{.discard() saem}, |+\rangle\langle+|_b|\!\}$$
$$\not\sim_{\frac{1}{2},0} \{\!|\texttt{discard}()||\texttt{meas } b \texttt{ then c!}b\texttt{.discard() saem}, |0\rangle\langle 0|_b|\!\}.$$

Nevertheless, the relation $\sim_{\zeta,\eta}$ is closed under parallel composition of the processes if $\eta > 2\zeta$. The condition $\eta > 2\zeta$ is reasonable, because the difference of outsider's quantum states infects her behavior.

Theorem 4.2.13. *If* $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ *and* $\eta > 2\zeta$ *hold,* $\{\!|P||R, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q||R, \sigma|\!\}$ *holds for all process* R.

Proof. We define

$$\hat{\mathcal{R}} := \{(\{\!|P||R, \rho|\!\}, \{\!|Q||R, \sigma|\!\}) \mid \{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}, R \in \mathcal{P}\}.$$

As $\sim_{\zeta,\eta}$ is symmetric, $\hat{\mathcal{R}}$ is symmetric. It is sufficient to show $\hat{\mathcal{R}}$ is a (ζ, η)-bisimulation. Let $(\{\!|P||R, \rho|\!\}, \{\!|Q||R, \sigma|\!\})$ be an arbitrary element in $\hat{\mathcal{R}}$. The static conditions are checked as follows. By the definition of $\mathrm{qv}(\cdot)$ and the condition $\mathrm{qv}(P) = \mathrm{qv}(Q)$ obtained from $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$, $\tilde{q} \stackrel{\text{def}}{=} \mathrm{qv}(P||R) = \mathrm{qv}(P) \cup \mathrm{qv}(R) = \mathrm{qv}(Q) \cup \mathrm{qv}(R) = \mathrm{qv}(Q||R)$ holds. Next,

$$\begin{aligned} d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma)) &= d(\mathrm{tr}_{\mathrm{qv}(R)}(\mathrm{tr}_{\mathrm{qv}(P)}(\rho)), \mathrm{tr}_{\mathrm{qv}(R)}(\mathrm{tr}_{\mathrm{qv}(Q)}(\sigma))) \\ &\leq d(\mathrm{tr}_{\mathrm{qv}(P)}(\rho), \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)) \leq \zeta. \end{aligned}$$

holds. We then show that the simulation condition is satisfied. Let $\mathcal{E}[\tilde{r}]$ be an arbitrary TPCP map acting on $\tilde{r} \subseteq qVar - \tilde{q}$. A transition of a parallely-composed process is either the 3 cases by the transition rules.
(Case 1) The transition is performed only by P. Assume $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold. By $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$, there exists $\{\!|Q', \sigma'|\!\}$ satisfying $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|P', \rho'|\!\} \sim_{\zeta,\eta} \{\!|Q', \sigma'|\!\}$. Hence, $\{\!|Q||R, \sigma|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q'||R, \sigma'|\!\}$ holds by (Left) rule and $\{\!|P'||R, \rho'|\!\}\hat{\mathcal{R}}\{\!|Q'||R, \sigma'|\!\}$ holds by the definition of $\hat{\mathcal{R}}$.
(Case 2) The transition is performed only by $\{\!|R, \rho|\!\}$. Assume $\{\!|R, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|R', \rho'|\!\}$ and $\mathrm{tr}(\rho') \geq \eta$ hold. Because R has a redex that causes the transition $\xrightarrow{\alpha}$, $\{\!|Q||R, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\alpha} \{\!|Q||R', \sigma'|\!\}$ holds for some σ' even if the transition is trace-decreasing by the following reasons.

- $\rho' = \mathcal{F}[\tilde{s}] \circ \mathcal{E}[\tilde{r}](\rho)$ and $\sigma' = \mathcal{F}[\tilde{s}] \circ \mathcal{E}[\tilde{r}](\sigma)$ hold for some CP map $\mathcal{F}[\tilde{s}]$ acting on $\tilde{s} \subseteq \mathrm{qv}(R)$.
- We have $d(\rho', \sigma') \leq d(\rho, \sigma) \leq \zeta$, which implies $|\mathrm{tr}(\rho') - \mathrm{tr}(\sigma')| \leq 2\zeta$.
- We have $\mathrm{tr}(\sigma') \geq \eta - 2\zeta > 0$.

It is thus sufficient to show $\{\!|P, \rho'|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma'|\!\}$ by the definition of $\hat{\mathcal{R}}$. $\rho' = \mathcal{F}[\tilde{s}] \circ \mathcal{E}[\tilde{r}](\rho)$ and $\sigma' = \mathcal{F}[\tilde{s}] \circ \mathcal{E}[\tilde{r}](\sigma)$ hold for some CP map $\mathcal{F}[\tilde{s}]$ acting on $\tilde{s} \subseteq \mathrm{qv}(R)$. As $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ and $\tilde{s}, \tilde{r} \subseteq qVar - \mathrm{qv}(P)$ hold, $\{\!|P, \rho'|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma'|\!\}$ holds by Lemma 4.2.9. This implies $\{\!|P||R', \rho'|\!\}\hat{\mathcal{R}}\{\!|Q||R', \sigma'|\!\}$.
(Case 3) The transition is performed by communication of P and R. As the communication rule is applied, the $P||R$ can be written as $\{\!|C_1[\mathsf{c}!q.P']||C_2[\mathsf{c}?r.R'], \mathcal{E}[\tilde{r}](\rho)|\!\}$ for some evaluation contexts $C_1[\_], C_2[\_]$, processes P', R', and non-restricted channel c. The transition to consider is

$$\{\!|C_1[\mathsf{c}!q.P']||C_2[\mathsf{c}?r.R'], \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\tau} \{\!|C_1[P']||C_2[R'], \mathcal{E}[\tilde{r}](\rho)|\!\}.$$

Assume $\mathrm{tr}(\rho) \geq \eta$. By $\{\!|C_1[\mathsf{c}!q.P'], \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ and $\{\!|C_1[\mathsf{c}!q.P'], \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\mathsf{c}!q}$ $\{\!|C_1[P'], \mathcal{E}[\tilde{r}](\rho)|\!\}$, $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\mathsf{c}!q}\xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|C_1[P'], \mathcal{E}[\tilde{r}](\rho)|\!\} \sim_{\zeta,\eta}$ $\{\!|Q', \sigma'|\!\}$ hold for some $\{\!|Q', \sigma'|\!\}$. This implies c is not restricted in Q and thus receive redex $\mathsf{c}?r$ in $C_2[\mathsf{c}?r.R']$ can be react. The reaction does not influence the quantum state σ'. Therefore,

$$\{\!|Q||R, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \{\!|Q'||C_2[R'], \sigma'|\!\} \text{ and } \{\!|C_1[P']||C_2[R'], \mathcal{E}[\tilde{r}](\rho)|\!\}\hat{\mathcal{R}}\{\!|Q'||C_2[R'], \sigma'|\!\}$$

hold by the definition of $\hat{\mathcal{R}}$. □

By the two previous lemmas and Proposition 4.2.6, we have the following corollary.

Corollary 4.2.14. *If* $\{\!|P, \rho|\!\} \sim_{\zeta,\eta} \{\!|Q, \sigma|\!\}$ *and* $\eta > 2\zeta$ *hold,* $\{\!|C[P], \rho|\!\} \sim_{\zeta,\eta} \{\!|C[Q], \sigma|\!\}$ *holds for all evaluation context* $C[\_]$.

4.3 Approximate Bisimulation up to Negligible Difference

In this section, we define the second approximate bisimulation based on the notion of negligibility. Although a security parameter is arbitrarily fixed in Chapter 3 and in the previous section, we do not fix if in this section. Instead, we treat $[\![\rho]\!] = [\![\rho]\!](\lambda)$ as a function of a security parameter λ for $\rho \in \mathcal{S}$. Accordingly, trace $\mathrm{tr}([\![\rho]\!])$ and trace distance $d([\![\rho]\!], [\![\sigma]\!])$ are a functions of the security parameter, namely, $\mathrm{tr}([\![\rho]\!])(\lambda) = \mathrm{tr}([\![\rho]\!]_\lambda)$ and $d([\![\rho]\!], [\![\sigma]\!])(\lambda) = d([\![\rho]\!]_\lambda, [\![\sigma]\!]_\lambda)$. Hence, the statements such as "$\mathrm{tr}([\![\rho]\!])$ is non-negligible" and "$d([\![\rho]\!], [\![\sigma]\!])$ is negligible" make sense. If $d([\![\rho]\!], [\![\sigma]\!])$ and $d([\![\sigma]\!], [\![\theta]\!])$ are negligible, then $d([\![\rho]\!], [\![\theta]\!])$ is negligible. As a result, the second approximate bisimulation relation $\sim$ is transitive and thus is an equivalence relation.

Since the operational semantics on $\mathcal{C}$ is defined with fixed security parameter (Definition 3.3.3), we modify the operational semantics on $\mathcal{C}$ with unfixed security parameter. The rules that need modification are (Meas0) and (Meas1), because λ must be fixed to determine the preconditions of them. We modify the rules as follows.

$$\frac{[\![\mathtt{proj1}[\mathtt{b}](\rho)]\!]_\lambda \neq O \text{ for infinitely many } \lambda}{\{\!|\mathtt{meas}\ b\ \mathtt{then}\ P\ \mathtt{saem}, \rho|\!\} \xrightarrow{\tau} \{\!|P, \mathtt{proj1}[\mathtt{b}](\rho)|\!\}} \text{(Meas1')}$$

$$\frac{[\![\mathtt{proj0}[\mathtt{b}](\rho)]\!]_\lambda \neq O \text{ for infinitely many } \lambda}{\{\!|\mathtt{meas}\ b\ \mathtt{then}\ P\ \mathtt{saem}, \rho|\!\} \xrightarrow{\tau} \{\!|\mathtt{discard}(\mathrm{qv}(P)), \mathtt{proj0}[\mathtt{b}](\rho)|\!\}} \text{(Meas0')}$$

In the following discussions, we may simply write $\mathrm{tr}(\rho)$ and $d(\rho, \sigma)$ as $\mathrm{tr}([\![\rho]\!])(\cdot)$ and $d([\![\rho]\!], [\![\sigma]\!])(\cdot)$ for $\rho, \sigma \in \mathcal{S}$. The definition of the approximate bisimulation relation is as follows.

Definition 4.3.1. *A symmetric relation* $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ *is called an approximate bisimulation if for all* $\{\!|P, \rho|\!\}\mathcal{R}\{\!|Q, \sigma|\!\}$,

1. $\mathrm{qv}(P) = \mathrm{qv}(Q) \stackrel{\mathrm{def}}{=} \tilde{q}$,
2. $d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma))$ *is negligible, and*
3. *for an arbitrary CP map* $\mathcal{E}[\tilde{r}]$ *acting on* $\tilde{r} \subseteq qVar - \tilde{q}$, *if* $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ *holds and* $\mathrm{tr}(\rho')$ *is non-negligible, then* $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ *and* $\{\!|P', \rho'|\!\}\mathcal{R}\{\!|Q', \sigma'|\!\}$ *holds for some* $\{\!|Q', \sigma'|\!\}$.

We call the above conditions *1*, *2* the static conditions and *3* the simulation condition.

Remark 4.3.2. *For a configuration* $\{\!|P, \rho|\!\} \in \mathcal{C} = \mathcal{P} \times \mathcal{S}$, *the branching structure of its transition does not depend on the security parameter. Except for transition rules* (Meas0) *and* (Meas1) *(Section 3.3.2), whether each rule can be applicable is determined only by a process, but there is no constructor depending on the security parameter in the syntax of the processes in* $\mathcal{P}$ *(Section 3.2.2). Recall that op in the construction* $op[\tilde{q}].P$ *is a symbol representing a TPCP map, and*

a transition caused by (Op) *is one-step τ-transition. Whether the rules* (Meas0′) *and* (Meas1′) *can be applicable depends on interpretations but does not on values that the security parameter take.*

Definition 4.3.3. *We define*

$$\sim = \{(\{\!|P,\rho|\!\}, \{\!|Q,\sigma|\!\}) \mid \{\!|P,\rho|\!\}\mathcal{R}\{\!|Q,\sigma|\!\} \text{ holds for some approximate bisimulation } \mathcal{R}\}.$$

We say $\{\!|P,\rho|\!\}$ and $\{\!|Q,\sigma|\!\}$ are approximately bisimilar if $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$.

There is another possible definition of the relation. Let us replace the requirement "$\mathrm{tr}(\rho')$ is non-negligible" in the condition 3 with "$\frac{\mathrm{tr}(\rho')}{\mathrm{tr}(\mathcal{E}[\tilde{r}](\rho))}$ is non-negligible", and let $\simeq$ be the relation defined similarly to Definition 4.3.3. Since $\frac{\mathrm{tr}(\rho')}{\mathrm{tr}(\mathcal{E}[\tilde{r}](\rho))} \geq \mathrm{tr}(\rho')$ holds, $\simeq \subseteq \sim$ holds. In fact, the relation $\simeq$ has properties that are similar to those discussed in the following propositions, lemmas, and theorem. We adopt Definition 4.5.1 for the following reason. It is natural that we assume a configuration $\{\!|P_0,\rho_0|\!\}$ satisfies $\mathrm{tr}(\rho_0) = 1$ when it formalizes a protocol. Suppose we have $\{\!|P_0,\rho_0|\!\} \xrightarrow{\alpha_0} \cdots \xrightarrow{\alpha_k} \{\!|P,\rho|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$, that $\mathrm{tr}(\rho)$ is non-negligible and that $\mathrm{tr}(\rho')$ is negligible. The probability to reach $\{\!|P,\rho|\!\}$ is non-negligible and to reach $\{\!|P',\rho'|\!\}$ is negligible. Therefore, we want to care $\{\!|P,\rho|\!\}$ but ignore $\{\!|P',\rho'|\!\}$. However, a case is possible where a configuration $\{\!|Q_0,\sigma_0|\!\}$ must simulate the transition $\{\!|P,\rho|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ to satisfy $\{\!|P_0,\rho_0|\!\} \simeq \{\!|Q_0,\sigma_0|\!\}$. This is because $\frac{\mathrm{tr}(\rho')}{\mathrm{tr}(\rho)}$ can be non-negligible even if $\mathrm{tr}(\rho')$ is negligible and $\mathrm{tr}(\rho)$ is non-negligible by the definition of negligible functions. We thus cannot ignore $\{\!|P',\rho'|\!\}$.

The following lemmas are proven similarly to the previous section.

Lemma 4.3.4. *$\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$ holds, iff*

1. $\mathrm{qv}(P) = \mathrm{qv}(Q) \stackrel{\mathrm{def}}{=} \tilde{q}$,
2. *$d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma))$ is negligible, and*
3. *for an arbitrary CP map $\mathcal{E}[\tilde{r}]$ acting on $\tilde{r} \subseteq qVar - \tilde{q}$,*
   - *if $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ holds and $\mathrm{tr}(\rho')$ is non-negligible, then there exists $\{\!|Q',\sigma'|\!\}$ satisfying $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$, and*
   - *if $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\alpha} \{\!|Q',\sigma'|\!\}$ holds and $\mathrm{tr}(\sigma')$ is non-negligible, then there exists $\{\!|P',\rho'|\!\}$ satisfying $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$.*

Lemma 4.3.5. *If $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$, then $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \sim \{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\}$ for all CP map $\mathcal{E}[\tilde{r}]$ that acts on $\tilde{r}$.*

Proposition 4.3.6. *$\{\!|P||Q,\rho|\!\} \sim \{\!|Q||P,\sigma|\!\}$.*

We then prepare lemmas to prove transitivity of the relation $\sim$.

Lemma 4.3.7. *If $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$ and $\{\!|P,\rho|\!\} \xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho')$ is non-negligible, then $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|Q',\sigma'|\!\}$.*

Proof. Assume $\{\!|P,\rho|\!\} \xrightarrow{\tau}{}^{n} \{\!|P',\rho'|\!\}$ and let $\{\!|P_i,\rho_i|\!\}$ be i-th configuration with $0 \leq i \leq n$. The case when $n = 0$ is trivial. Let $n > 0$. Since $\rho = \rho_0 \geq \rho_1 \cdot \cdot \geq \rho_n = \rho'$ holds and $\mathrm{tr}(\rho')$ is non-negligible, $\mathrm{tr}(\rho_i)$ is non-negligible for all i. Therefore, $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*}{}^{n} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$ hold for some $\{\!|Q',\sigma'|\!\}$. □

Lemma 4.3.8. *If $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$ and $\{\!|P,\rho|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho')$ is non-negligible, then $\{\!|Q,\sigma|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$ for some $\{\!|Q',\sigma'|\!\}$.*

Proof. It is proven similarly to the previous lemma. □

Proposition 4.3.9. *The relation $\sim$ is an equivalence relation.*

Proof. **(Reflexivity)** Let $Id_{\mathcal{C}}$ be the identity relation on $\mathcal{C}$. For all $\{\!|P,\rho|\!\} \in \mathcal{C}$, $\{\!|P,\rho|\!\} Id_{\mathcal{C}} \{\!|P,\rho|\!\}$ holds. It is sufficient to show $Id_{\mathcal{C}}$ is an approximate bisimulation. Assume $(\{\!|P,\rho|\!\}, \{\!|P,\rho|\!\})$ is an arbitrary element in $Id_{\mathcal{C}}$ and $\mathrm{tr}(\rho)$ is non-negligible. The static conditions are easily checked. Let $\mathcal{E}[\tilde{r}]$ be an arbitrary CP map and assume $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho')$ is negligible. As $\{\!|P',\rho'|\!\} Id_{\mathcal{C}} \{\!|P',\rho'|\!\}$ holds, $Id_{\mathcal{C}}$ is an approximate bisimulation.
(Symmetry) ($\Rightarrow$) implication of Lemma 4.3.4 is the condition that $\sim$ is an approximate bisimulation. An approximate bisimulation relation is defined to be symmetric.
(Transitivity) It is sufficient to show $\sim \circ \sim$ is an approximate bisimulation relation. Let $(\{\!|P,\rho|\!\}, \{\!|R,\theta|\!\})$ be an arbitrary element of $\sim \circ \sim$. There exists $\{\!|Q,\sigma|\!\}$ satisfying $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$ and $\{\!|Q,\sigma|\!\} \sim \{\!|R,\theta|\!\}$. The static conditions are easily checked using triangle inequality of trace distance $d(\cdot,\cdot)$. Let $\mathcal{E}[\tilde{r}]$ be an arbitrary CP map acting on $\tilde{r} \subseteq qVar - \mathrm{qv}(P)$ and assume $\{\!|P,\mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P',\rho'|\!\}$ and $\mathrm{tr}(\rho')$ is non-negligible. By $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$, there exists $\{\!|Q',\sigma'|\!\}$ satisfying $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q',\sigma'|\!\}$ and $\{\!|P',\rho'|\!\} \sim \{\!|Q',\sigma'|\!\}$. By its static conditions, we have $d(\mathrm{tr}_{\mathrm{qv}(P')}(\rho'), \mathrm{tr}_{\mathrm{qv}(Q')}(\sigma'))$ is negligible. This implies $|\mathrm{tr}(\rho') - \mathrm{tr}(\sigma')|$ is negligible and thus we have that $\mathrm{tr}(\sigma')$ is non-negligible. We have $\{\!|Q,\mathcal{E}[\tilde{r}](\sigma)|\!\} \sim \{\!|R,\mathcal{E}[\tilde{r}](\theta)|\!\}$ applying lemma 4.3.5 to $\{\!|Q,\sigma|\!\} \sim \{\!|R,\theta|\!\}$. Next by lemma 4.3.8, we have $\{\!|R,\mathcal{E}[\tilde{r}](\theta)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|R',\theta'|\!\}$ and $\{\!|Q',\sigma'|\!\} \sim \{\!|R',\theta'|\!\}$ for some $\{\!|R',\theta'|\!\}$. Therefore, $\{\!|P',\rho'|\!\} \sim \circ \sim \{\!|R',\theta'|\!\}$. □

We prove congruence of the relation $\sim$. We first show that $\sim$ is closed by restriction.

Lemma 4.3.10. *If $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$ holds, then $\{\!|P\backslash L,\rho|\!\} \sim \{\!|Q\backslash L,\sigma|\!\}$ holds.*

Proof. It is proven similarly to Lemma 4.2. □

The next theorem states that the relation $\sim$ is closed under parallel composition of processes. With this theorem and Lemma 4.3.10, we immediately have that $\sim$ is closed by application of an arbitrary evaluation context. The structure of the proof is same as Theorem 4.2.13.

Theorem 4.3.11. *If $\{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}$, then $\{\!|P||R,\rho|\!\} \sim \{\!|Q||R,\sigma|\!\}$ for all process R.*

Proof. We define

$$\mathcal{R} := \{(\{\!|P||R,\rho|\!\}, \{\!|Q||R,\sigma|\!\}) \mid \{\!|P,\rho|\!\} \sim \{\!|Q,\sigma|\!\}, R \in \mathcal{P}\}.$$

It is sufficient to show $\mathcal{R}$ is an approximate bisimulation. $\mathcal{R}$ is symmetric by the definition. Let $(\{\!|P||R, \rho|\!\}, \{\!|Q||R, \sigma|\!\})$ be an arbitrary element in $\mathcal{R}$. The static conditions are checked similarly to Theorem 4.2.13. The simulation condition is checked similarly to Theorem 4.2.13 for **(Case 1)** and **(Case 3)**. For **(Case 2)**, we used the fact that the function $(f-g)(n) \stackrel{\text{def}}{=} f(n) - g(n)$ is non-negligible if f is non-negligible and g is negligible. □

Similarly to the discussion in the previous section, we have the following corollary.

Corollary 4.3.12. *If* $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ *holds, then* $\{\!|C[P], \rho|\!\} \sim \{\!|C[Q], \sigma|\!\}$ *holds for all evaluation context* $C[\_]$.

We have proved that the relation $\sim$ is equivalence and closed by application of an arbitrary evaluation context. We thus say it is congruent. The congruence property is useful in practice. For example, it allows us to infer equivalence of multiple sessions of protocols.

Corollary 4.3.13. *If* $\{\!|P_1, \rho_1 * \rho_1^E|\!\} \sim \{\!|Q_1, \sigma_1 * \rho_1^E|\!\}$, $\{\!|P_2, \rho_2 * \rho_2^E|\!\} \sim \{\!|Q_2, \sigma_2 * \rho_2^E|\!\}$ *and* $\mathrm{qv}(P_1) \cap \mathrm{qv}(P_2) = \mathrm{qv}(P_1) \cap \mathrm{qv}(Q_2) = \mathrm{qv}(Q_1) \cap \mathrm{qv}(P_2) = \mathrm{qv}(Q_1) \cap \mathrm{qv}(Q_2) = \emptyset$ *hold for all* ρ_1^E, ρ_2^E, *then* $\{\!|P_1||P_2, \rho_1 * \rho_2 * \rho^E|\!\} \sim \{\!|Q_1||Q_2, \sigma_1 * \sigma_2 * \rho^E|\!\}$ *holds for all* ρ^E.

Proof. We have $\{\!|P_1, \rho_1 * \rho_2 * \rho^E|\!\} \sim \{\!|Q_1, \sigma_1 * \rho_2 * \rho^E|\!\}$ by substituting $\rho_2 * \rho^E$ for ρ_1^E in the assumption. By congruence, we have $\{\!|P_1||P_2, \rho_1 * \rho_2 * \rho^E|\!\} \sim \{\!|Q_1||P_2, \sigma_1 * \rho_2 * \rho^E|\!\}$. Similarly, we have $\{\!|Q_1||P_2, \sigma_1 * \rho_2 * \rho^E|\!\} \sim \{\!|Q_1||Q_2, \sigma_1 * \sigma_2 * \rho^E|\!\}$. By transitivity of $\sim$, we obtain the conclusion. □

Let configurations $\{\!|P_i, \rho * \rho_i^E|\!\}$ and $\{\!|Q_i, \sigma * \rho_i^E|\!\}$ formalize an actual and an ideal protocols for $i = 1, 2$. By the above corollary, we have $\{\!|P_1||P_2, \rho_1 * \rho_2 * \rho^E|\!\} \sim \{\!|Q_1||Q_2, \sigma_1 * \sigma_2 * \rho^E|\!\}$. This means that $\{\!|P_1||P_2, \rho_1 * \rho_2 * \rho^E|\!\}$ is approximately secure, provided that the ideal protocol is secure even if they run in parallel. The latter condition depends on protocols but possibly be satisfied. In fact, EDP-ideal protocol that we consider in the next chapter satisfies the condition, because Alice and Bob generate a shared key using pre-shared EPR pairs. Another example of application is discussed in the subsection Outputting Secret Keys in Section 5.3.

Although we use only the relation $\sim$ for the verification in this thesis, the relation $\sim_{\zeta,\eta}$ will be useful when we evaluate the gap of two configurations quantitatively. By $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, the value of the trace distance is simply understood to be negligible, but it cannot be evaluated more explicitly. Using the relation $\sim_{\zeta,\eta}$, the gap can be evaluated concretely. For example, if $\{\!|P_i, \rho_i * \rho_i^E|\!\} \sim_{\zeta,\eta} \{\!|Q_i, \sigma_i * \rho_i^E|\!\}$ holds for all ρ_i^E with $\mathrm{qv}(\rho_i^E) = qVar - \mathrm{qv}(P_i)$ and for all $i \in [1..k]$ and $\mathrm{qv}(P_1) \cap \cdots \cap \mathrm{qv}(P_k) = \emptyset$ holds, then we have

$$\{\!|P_1|| \cdots ||P_k, \rho_1 * \cdot * \rho_k * \rho^E|\!\} \sim_{\zeta',\eta'} \{\!|Q_1|| \cdots ||Q_k, \sigma_1 * \cdot * \sigma_k * \rho^E|\!\} \text{ for all } \rho^E,$$

where $\zeta' = k\zeta$ and $\eta' = (k-1)(k+2)\zeta + \eta$.

4.4 Automated Verification of Approximate Bisimulation

4.4.1 Algorithm

We extended Verifier1, which is described in Chapter 3, to verify the approximate bisimilarity. Let us call the extended verifier *Verifier2*. We applied Verifier2 to

the second part of Shor-Preskill's security proof. This is described in the next chapter.

Verifier2 takes as input two elements in $\mathcal{P} \times \mathcal{S}$, a user-defined set of equations *eqs* on symbolic quantum states, and additionally a user-defined set of triples $inds \subseteq \mathcal{S} \times \mathcal{S} \times S_{nat}$, which we call *indistinguishability expressions*. An indistinguishability expression (ρ, σ, n) intuitively means the trace distance of ρ and σ is negligible with respect to n.

We modified the steps 1, 4, 5, 6, and 7 in the algorithm described in Section 3.4.2. The new algorithm of the recursive procedure is as follows.

1. The procedure takes as input two configurations $\{\!|P_0, \rho_0|\!\}$, $\{\!|Q_0, \sigma_0|\!\}$ and user-defined equations *eqs* and indistinguishability expressions *inds* on quantum states.

2. If P_0 and Q_0 can perform any τ-transitions of TPCP map applications, they are all performed at this point. Let $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ be obtained configurations.

3. Whether $\mathrm{qv}(P) = \mathrm{qv}(Q)$ is checked. If it does not hold, the procedure returns *false*.

4. Whether $\mathtt{Tr}[\mathrm{qv}(P)](\rho) = \mathtt{Tr}[\mathrm{qv}(Q)](\sigma)$ is checked using *eqs* and *inds*. If it does not hold, the procedure returns *false*.

5. A new *CP* map symbol $\mathcal{E}\,\mathtt{[qv}(\rho) - \mathrm{qv}(P)\mathtt{]}$ that stands for an arbitrary operation is generated.

6. For each $\{\!|P', \rho'|\!\}$ such that $\{\!|P, \mathcal{E}\,\mathtt{[qv}(\rho) - \mathrm{qv}(P)\mathtt{]}\,\mathtt{(}\rho\mathtt{)}|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$, the procedure checks whether there exists $\{\!|Q', \sigma'|\!\}$ such that $\{\!|Q, \mathcal{E}\,\mathtt{[qv}(\sigma) - \mathrm{qv}(Q)\mathtt{]}\,\mathtt{(}\sigma\mathtt{)}|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and the procedure returns *true* with input $\{\!|P', \rho'|\!\}$, $\{\!|Q', \sigma'|\!\}$, and *eqs*. If there exists, it goes to the next step 7. Otherwise, it returns *false*.

7. For each $\{\!|Q', \sigma'|\!\}$ such that $\{\!|Q, \mathcal{E}\,\mathtt{[qv}(\sigma) - \mathrm{qv}(Q)\mathtt{]}\,\mathtt{(}\sigma\mathtt{)}|\!\} \xrightarrow{\alpha} \{\!|Q', \sigma'|\!\}$, the procedure checks whether there exists $\{\!|P', \rho'|\!\}$ such that $\{\!|P, \mathcal{E}\,\mathtt{[qv}(\rho) - \mathrm{qv}(P)\mathtt{]}\,\mathtt{(}\rho\mathtt{)}|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|P', \rho'|\!\}$ and the procedure returns *true* with input $\{\!|P', \rho'|\!\}$ and $\{\!|Q', \sigma'|\!\}$, and *eqs*. If there exists, it returns *true*. Otherwise, it returns *false*.

The way to use *inds* to test $\mathtt{Tr}[\mathrm{qv}(P)](\rho) = \mathtt{Tr}[\mathrm{qv}(Q)](\sigma)$ is similar to that of *eqs*, that is, for $(\rho, \sigma, n) \in inds$, a part in an objective quantum state that matches to ρ is rewritten to σ.

4.4.2 Correctness of Verifier2

A user-defined set *inds* is said to be valid if for all element $(\rho, \sigma, n) \in inds$, $d([\![\rho]\!], [\![\sigma]\!])$ is a negligible function of $[\![n]\!]$. Let *eqs* and *inds* be valid. Let a relation $\mathcal{R}_{eqs,inds} \subseteq \mathcal{C} \times \mathcal{C}$ be defined as follows.

$$\mathcal{R}_{eqs,inds} := \{(\{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\}) \mid \text{Verifier2 returns } \textit{true} \text{ with } \{\!|P, \rho|\!\}, \{\!|Q, \sigma|\!\} \text{ using } \textit{eqs} \text{ and } \textit{inds}.\}$$

The relation $\mathcal{R}_{eqs,inds}$ is an approximate bisimulation relation.

The argument about the correctness of Verifier2 is basically similar to that about the original one. We focused on the two different points. The first is

that whether the trace distance of objective quantum states is negligible or not is tested instead of the equality of the states. The second point is about the simulation condition.

On the Static Condition of Partial Trace

It is necessary to check the partial rewriting of quantum states done by Verifier2 is correct. It rewrites a symbolic representation of the form $\rho_l * \rho * \rho_r$ to the symbolic representation $\rho_l * \sigma * \rho_r$ given $(\rho, \sigma, n) \in \mathit{inds}$. The correctness is guaranteed from the fact that $d(X, Y) = d(X_l \otimes X \otimes X_r, X_l \otimes Y \otimes X_r)$ holds for all $X, Y, X_l, X_r \in \Delta(\mathcal{H})$. If (ρ, σ, n) is valid, namely $d([\![\rho]\!], [\![\sigma]\!])$ is negligible with respect to $[\![n]\!]$, then $d([\![\rho_l * \rho * \rho_r]\!], [\![\rho_l * \sigma * \rho_r]\!])$ is negligible with respect to $[\![n]\!]$ because $d([\![\rho_l * \rho * \rho_r]\!], [\![\rho_l * \sigma * \rho_r]\!]) = d([\![\rho_l]\!] \otimes [\![\rho]\!] \otimes [\![\rho_r]\!], [\![\rho_l]\!] \otimes [\![\sigma]\!] \otimes [\![\rho_r]\!]) = d([\![\rho]\!], [\![\sigma]\!])$ holds.

On the Simulation Condition

The simulation condition of approximate bisimulation is only required to transitions with non-negligible probability, stating

- If $\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ holds and $\mathrm{tr}(\rho')$ is non-negligible, then $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|P', \rho'|\!\} \mathcal{R} \{\!|Q', \sigma'|\!\}$ hold for some $\{\!|Q', \sigma'|\!\}$.

However, Verifier2 does not check whether the probability of a transition is non-negligible or not. It returns *false* when a transition cannot be simulated even if the probability is negligible. As for simulation, the condition that Verifier2 returns *true* is strictly stronger than the simulation condition of the approximate bisimulation.

4.4.3 Relation between the Verifiers

Before the extension, the steps 6 and 7 required the correspondence of qubit variable b when there is a transition caused by (Meas0) or (Meas1) rules. The proof of Verifier1's soundness was made easier by this condition. Verifier2 does not check such correspondence, which straightforwardly checks the conditions stated in the definition of the relation $\sim$. On this point, Verifier1 checks more strict condition. On the other hand, for the step 5, Verifier2 generates a CP map symbol representing the outsider's operation, which is not necessarily TPCP, while Verifier1 generates a TPCP map symbol. On this point, Verifier2 checks more strict condition.

In fact, whether an outsider's operation is TPCP or CP does not matter in verification in most cases. Only the rewriting rule (3.2) of partial traces in Section 3.4.1 cannot be applied if *op* is a CP map symbol. Except for it, quantum operators are treated equivalently in the both verifiers whether they are TPCP or CP. Moreover, even if outsider's operators are TPCP, the rule (3.2) cannot be applied to them in most cases, where we assume the outsider has her own "local memory". Assume she does not send a quantum variable q^E to the insider (the process). It is in the domain of her operation $\mathcal{E}$ in general. Hence, an arbitrary symbol generated in the step 5 is of the form $\mathcal{E}[\cdots, q^E, \cdots]$. However, she does not send q^E to the process. Therefore, in any expressions of the form

$$\mathtt{Tr}[\tilde{q}](\cdots \mathcal{E}[\cdots, q^E, \cdots] \cdots),$$

which appear in the test of equality of partial traces, $q^E \notin \tilde{q}$ holds and thus the rule (3.2) cannot be applied.

Let us now consider a new verifier, which we call Verifier3, that generates *CP* maps in the step 5 but executes the same algorithm as Verifier1. We have that Verifier3 verifies more strict condition than Verifier2, in other words, if Verifier3 returns *true* with configurations $\{\!|P, \rho|\!\}$, $\{\!|Q, \sigma|\!\}$, and user-defined equations *eqs*, Verifier2 returns *true* with the same input.

4.5 Guarantees and Limitations of Approximate Bisimulation

4.5.1 Application to Verification of QKD protocols' Security

We explain about feasibility of the relation $\sim$ for verification of security of QKD protocols. In the next chapter, we will verify $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, where $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ are configurations formalizing the EDP-based protocol (Section 2.4.2) and EDP-ideal (Section 5.5). Assume $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ and

$$\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P_1, \mathcal{E}^1[\tilde{r}_1](\rho_1)|\!\} \xrightarrow{\alpha_1} \cdots \xrightarrow{\mathsf{ska}!k_A} \{\!|P', \rho'|\!\}$$

and $\mathrm{tr}(\rho')$ is non-negligible, where the last transition $\xrightarrow{\mathsf{ska}!k_A}$ represents that Alice's key k_A is created[1]. Then, there exists the following transition

$$\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q_1, \mathcal{E}^1[\tilde{r}_1](\sigma_1)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}_1} \xrightarrow{\tau*} \cdots \xrightarrow{\tau*} \xrightarrow{\mathsf{ska}!k_A} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$$

such that $d(\mathrm{tr}_{\mathrm{qv}(P')}(\rho'), \mathrm{tr}_{\mathrm{qv}(Q')}(\sigma'))$ is negligible. By Proposition 4.1.8, we have that

$$|\mathrm{tr}(\rho') - \mathrm{tr}(\sigma')| \text{ and } |\mathrm{tr}(\rho') \cdot \mathrm{tr}(\pi \frac{\mathrm{tr}_{\mathrm{qv}(P')}(\rho')}{\mathrm{tr}(\rho')}) - \mathrm{tr}(\sigma') \cdot \mathrm{tr}(\pi \frac{\mathrm{tr}_{\mathrm{qv}(Q')}(\sigma')}{\mathrm{tr}(\sigma')})|$$

are negligible for all projector π. Especially, let π be the projector to the subspace where i-th bits of Alice's key and Eve's key are equal. We can rephrase the above expression as follows.

$$|\Pr(A) - \Pr(B)| \text{ and } |\Pr(A)\Pr(k_{A,i} = k_{E,i}|A) - \Pr(B)\Pr(k'_{A,i} = k'_{E,i}|B)|$$

are negligible, where

- $k_{A,i}$ and $k_{E,i}$ are random variables of i-th bits of Alice's and Eve's keys in the EDP-based protocol,
- $k'_{A,i}$ and $k'_{E,i}$ are those in EDP-ideal, and
- A and B are the events that $\{\!|P, \rho|\!\}$ reaches $\{\!|P', \rho'|\!\}$ and $\{\!|Q, \sigma|\!\}$ reaches $\{\!|Q', \sigma'|\!\}$.

Moreover, we have

$$\begin{aligned}
&|\Pr(A)\Pr(k_{A,i} = k_{E,i}|A) - \Pr(B)\Pr(k'_{A,i} = k'_{E,i}|B)| \\
\leq &\Pr(A)|\Pr(k_{A,i} = k_{E,i}|A) - \Pr(k'_{A,i} = k'_{E,i}|B)| \\
&+ \Pr(k'_{A,i} = k'_{E,i}|B)|\Pr(A) - \Pr(B)| \\
= &\Pr(A)|\Pr(k_{A,i} = k_{E,i}|A) - \frac{1}{2}| + \frac{1}{2}|\Pr(A) - \Pr(B)|.
\end{aligned}$$

The equation $\Pr(k'_{A,i} = k'_{E,i}|B) = \frac{1}{2}$ holds by the definition of EDP-ideal. If $\Pr(A)$ is greater than negligible, we have that $|\Pr(k_{A,i} = k_{E,i}|A) - \frac{1}{2}|$ is negligible. It seems possible to derive that the mutual information of Alice's and Eve's keys is negligible. Similarly, it is derived that Alice's and Bob's keys are identical with overwhelming probability.

[1] In Chapter 5, we actually formalize the protocols as configurations that do such transitions. This point is discussed in Section 5.3

4.5.2 Application to Other Protocols

As long as we aim to verify security of an actual protocol $\{\!|P, \rho|\!\}$ by proving $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ for an ideal protocol $\{\!|Q, \sigma|\!\}$, we can discuss similarly to the previous subsection. Concretely, if $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, then for all $\{\!|P', \rho'|\!\}$ such that

$$\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P_1, \mathcal{E}^1[\tilde{r}_1](\rho_1)|\!\} \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_m} \{\!|P', \rho'|\!\}$$

and $\mathrm{tr}(\rho')$ is non-negligible, there exists $\{\!|Q', \sigma'|\!\}$ such that

$$\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}}\xrightarrow{\tau*} \{\!|Q_1, \mathcal{E}^1[\tilde{r}_1](\sigma_1)|\!\} \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}_1}\xrightarrow{\tau*} \cdots \xrightarrow{\tau*}\xrightarrow{\hat{\alpha}_m}\xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$$

and $\{\!|P', \rho'|\!\} \sim \{\!|Q', \sigma'|\!\}$. The last condition implies $d(\mathrm{tr}_{\mathrm{qv}(P')}(\rho'), \mathrm{tr}_{\mathrm{qv}(Q')}(\sigma'))$ is negligible. By proposition 4.1.8, we have

$$|\mathrm{tr}(\rho') - \mathrm{tr}(\sigma')| \text{ and } \mathrm{tr}(\rho') d\left(\frac{\mathrm{tr}_{\mathrm{qv}(P')}(\rho')}{\mathrm{tr}(\rho')}, \frac{\mathrm{tr}_{\mathrm{qv}(Q')}(\sigma')}{\mathrm{tr}(\sigma')}\right)$$

are negligible. We thus have the following conclusions.

1. The probability to reach $\{\!|P', \rho'|\!\}$ from $\{\!|P, \rho|\!\}$ is negligibly close to that to reach $\{\!|Q', \sigma'|\!\}$ from $\{\!|Q, \sigma|\!\}$.
2. The greater $\mathrm{tr}(\rho')$ we have, the less $d(\frac{\mathrm{tr}_{\mathrm{qv}(P')}(\rho')}{\mathrm{tr}(\rho')}, \frac{\mathrm{tr}_{\mathrm{qv}(Q')}(\sigma')}{\mathrm{tr}(\sigma')})$ we have. Especially, if $\mathrm{tr}(\rho')$ is greater than negligible, then $d(\frac{\mathrm{tr}_{\mathrm{qv}(P')}(\rho')}{\mathrm{tr}(\rho')}, \frac{\mathrm{tr}_{\mathrm{qv}(Q')}(\sigma')}{\mathrm{tr}(\sigma')})$ is negligible.

When the protocol $\{\!|P, \rho|\!\}$ is for generation of certain data, the data will be sent to the outside by the final transition $\xrightarrow{\alpha_m}$ of the form $\xrightarrow{\mathsf{c}!q}$, where $\mathcal{H}_q$ is the state space of the data. By the condition 2 above, we have that whenever the probability to reach $\{\!|P', \rho'|\!\}$ from the start point $\{\!|P, \rho|\!\}$ is greater than negligible, the data have been almost correctly generated at $\{\!|P', \rho'|\!\}$.

4.5.3 Limitations

Since single transition sequences are considered in Section 4.5.1 and 4.5.2, what is verified is that for each transition sequence of an actual protocol $\{\!|P, \rho|\!\}$, there exists a corresponding transition sequence of an ideal protocol $\{\!|Q, \sigma|\!\}$. Such reasoning of security can be applicable only if we could find an ideal protocol, like EDP-ideal, where *security is guaranteed for each sequence*.

However, guarantees of $\sim$ with respect to the original qCCS are not clear: even if we have $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, it seems difficult to prove that $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ are "approximately bisimilar" in the original qCCS by the following reason. When the approximate bisimilarity $\sim$ is considered, each transition caused by (Meas0′) or (Meas1′) is treated *independently* as a nondeterministic transition, while *a pair of the two transitions* caused by a part of a process of the form $\mathtt{meas}\ b_0\ \mathtt{then}\ P_0\ \mathtt{saem}$ (we may call one a *measurement sentence*) actually represents one probabilistic transition in the original qCCS.

Let us explain about the problem more concretely by an example. Let $C[\_]$ be an evaluation context, and let $\{\!|P, \rho|\!\}$ be $\{\!|C[\mathtt{meas}\ b\ \mathtt{then}\ P'\ \mathtt{saem}], \rho|\!\}$. Assume $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$ and we aim to prove that $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ are

"approximately bisimilar". The configuration $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ performs the following probabilistic transition in the original qCCS.

$$\begin{aligned}
&(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!) = (\!|\overline{C}[|1\rangle\langle 1|[b;x].\texttt{if } x = 1 \texttt{ then } \mathrm{cnv}(P')\ \texttt{fi}], [\![\rho]\!]|\!) \\
\xrightarrow{\tau} & p_0 \bullet (\!|\overline{C}[\texttt{if } 0 = 1 \texttt{ then } \mathrm{cnv}(P')\ \texttt{fi}], \frac{[\![\rho_0]\!]}{p_0}|\!) \boxplus \\
& p_1 \bullet (\!|\overline{C}[\texttt{if } 1 = 1 \texttt{ then } \mathrm{cnv}(P')\ \texttt{fi}], \frac{[\![\rho_1]\!]}{p_1}|\!) \\
\dot{\approx} & p_0 \bullet (\!|\mathrm{cnv}(C[\texttt{discard}(\mathrm{qv}(P'))]), \frac{[\![\rho_0]\!]}{p_0}|\!) \boxplus p_1 \bullet (\!|\mathrm{cnv}(C[P']), \frac{[\![\rho_1]\!]}{p_1}|\!) \stackrel{\mathrm{def}}{\equiv} \mu,
\end{aligned}$$

where $\overline{C}[\_] = \mathrm{cnv}(C)[\_], \rho_0 = \texttt{proj0}[b](\rho)$, $\rho_1 = \texttt{proj1}[b](\rho), p_0 = \mathrm{tr}(\rho_0)$, and $p_1 = \mathrm{tr}(\rho_1)$. In our simplified system, the configuration $\{\!|P, \rho|\!\}$ performs the following transitions that are caused by the same measurement sentence. The following pair of the transitions represents the above transition by $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$.

$$\begin{aligned}
&\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|C[\texttt{discard}(\mathrm{qv}(P'))], \rho_0|\!\} \text{ and} \\
&\{\!|P, \rho|\!\} \xrightarrow{\tau} \{\!|C[P'], \rho_1|\!\}
\end{aligned}$$

By $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, there exist configurations $\{\!|Q_0, \sigma_0|\!\}$ and $\{\!|Q_1, \sigma_1|\!\}$ satisfying

$$\begin{aligned}
&\{\!|Q, \sigma|\!\} \xrightarrow{\tau*} \{\!|Q_0, \sigma_0|\!\} \text{ and } \{\!|C[\texttt{discard}(\mathrm{qv}(P'))], \rho_0|\!\} \sim \{\!|Q_0, \sigma_0|\!\} \\
&\{\!|Q, \sigma|\!\} \xrightarrow{\tau*} \{\!|Q_1, \sigma_1|\!\} \text{ and } \{\!|C[P'], \rho_1|\!\} \sim \{\!|Q_1, \sigma_1|\!\}.
\end{aligned}$$

Let ν be a distribution $\mathrm{tr}(\sigma_0) \bullet (\!|\mathrm{cnv}(Q_0), \frac{[\![\sigma_0]\!]}{\mathrm{tr}(\sigma_0)}|\!) \boxplus \mathrm{tr}(\sigma_1) \bullet (\!|\mathrm{cnv}(Q_1), \frac{[\![\sigma_1]\!]}{\mathrm{tr}(\sigma_1)}|\!)$. Even if $\{\!|Q, \sigma|\!\} \xrightarrow{\tau*} \{\!|Q_0, \sigma_0|\!\}$ and $\{\!|Q, \sigma|\!\} \xrightarrow{\tau*} \{\!|Q_1, \sigma_1|\!\}$ hold, the condition $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!) \Rightarrow \nu$ is not guaranteed. This is because measurements performed in the $\tau*$-transitions are not necessarily caused by the same measurement sentence.

To verify "approximate bisimilarity" of $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ focusing on the transitions of $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$ in our simplified system, one possible way is to treat two transitions caused by the same measurement sentence as a pair, similarly to Verifier1 (step 6 (b) in Section 3.4.2). Let us introduce the following new relation which requires similar conditions to what Verifier1 checks.

Definition 4.5.1. *A symmetric relation $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is called a near bisimulation if for all $\{\!|P, \rho|\!\} \mathcal{R} \{\!|Q, \sigma|\!\}$,*

1. $\mathrm{qv}(P) = \mathrm{qv}(Q) \stackrel{\mathrm{def}}{=} \tilde{q}$,
2. *$d(\mathrm{tr}_{\tilde{q}}(\rho), \mathrm{tr}_{\tilde{q}}(\sigma))$ is negligible, and*
3. *for an arbitrary CP map $\mathcal{E}[\tilde{r}]$ acting on $\tilde{r} \subseteq qVar - \tilde{q}$,*
   - *If*
     - *$\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \xrightarrow{\alpha} \{\!|P', \rho'|\!\}$ holds,*
     - *$\mathrm{tr}(\rho')$ is non-negligible, and*
     - *$\mathrm{tr}(\mathcal{E}[\tilde{r}](\rho)) - \mathrm{tr}(\rho')$ is negligible,*

     then $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \xrightarrow{\hat{\alpha}} \xrightarrow{\tau*} \{\!|Q', \sigma'|\!\}$ and $\{\!|P', \rho'|\!\} \mathcal{R} \{\!|Q', \sigma'|\!\}$ holds for some $\{\!|Q', \sigma'|\!\}$.
   - *If*
     - *$\{\!|P, \mathcal{E}[\tilde{r}](\rho)|\!\} \equiv \{\!|C[\texttt{meas } b \texttt{ then } P' \texttt{ saem}], \mathcal{E}[\tilde{r}](\rho)|\!\} \stackrel{\mathrm{def}}{\equiv} X$,*

– $X \xrightarrow{\tau} \{\!|C[\texttt{discard}(\mathrm{qv}(P'))], \rho_0|\!\}$,
– $X \xrightarrow{\tau} \{\!|C[P'], \rho_1|\!\}$, *and*
– *both* $\mathrm{tr}(\rho_0)$ *and* $\mathrm{tr}(\rho_1)$ *are non-negligible*

for some $C[\_], b, P', \rho_0, \rho_1$, *then*

– $\{\!|Q, \mathcal{E}[\tilde{r}](\sigma)|\!\} \xrightarrow{\tau*} \{\!|D[\texttt{meas}\ b\ \texttt{then}\ Q'\ \texttt{saem}], \hat{\sigma}|\!\} \stackrel{\mathrm{def}}{\equiv} Y$,
– $Y \xrightarrow{\tau} \{\!|D[\texttt{discard}(\mathrm{qv}(Q'))], \hat{\sigma}_0|\!\} \xrightarrow{\tau*} \{\!|Q_0, \sigma_0|\!\}$,
– $Y \xrightarrow{\tau} \{\!|D[Q'], \hat{\sigma}_1|\!\} \xrightarrow{\tau*} \{\!|Q_1, \sigma_1|\!\}$,
– $\{\!|C[\texttt{discard}(\mathrm{qv}(P'))], \rho_0|\!\} \mathcal{R} \{\!|Q_0, \sigma_0|\!\}$, *and*
– $\{\!|C[P'], \rho_1|\!\} \mathcal{R} \{\!|Q_1, \sigma_1|\!\}$ *hold*

for some $D[\_], Q', \hat{\sigma}, \hat{\sigma}_0, \hat{\sigma}_1, \{\!|Q_0, \sigma_0|\!\}$, *and* $\{\!|Q_1, \sigma_1|\!\}$.

We write $\{\!|P, \rho|\!\} \simeq \{\!|Q, \sigma|\!\}$ *if* $\{\!|P, \rho|\!\} \mathcal{R} \{\!|Q, \sigma|\!\}$ *holds for some near bisimulation* $\mathcal{R}$, *and say* $\{\!|P, \rho|\!\}$ *and* $\{\!|Q, \sigma|\!\}$ *are nearly bisimilar.*

In fact, the properties similar to Proposition 4.3.9, Theorem 4.3.11, and Corollary 4.3.12 hold for the relation $\simeq$, which are proved by a similar line of arguments to that of the properties of $\sim$.

Proposition 4.5.2. *The relation* $\simeq$ *is an equivalence relation.*

Theorem 4.5.3. *If* $\{\!|P, \rho|\!\} \simeq \{\!|Q, \sigma|\!\}$ *holds, then* $\{\!|C[P], \rho|\!\} \simeq \{\!|C[Q], \sigma|\!\}$ *holds for all evaluation context* $C[\_]$.

If we aim to prove that $\{\!|P, \rho|\!\} \simeq \{\!|Q, \sigma|\!\}$ implies "approximate bisimilarity" of $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$, we could draw similar arguments to those of the soundness of Verifier1 (Section 3.5), replacing "equal" with "approximately equal". Precisely, whether we can prove the statement depends on the definition of approximate bisimilarity of $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$, which is an open problem.

Let us note a problem when defining a notion of approximate bisimulation in the original qCCS. In the notions of approximate bisimulation in nondeterministic qCCS, which are defined in this chapter, transitions with negligible probability are ignored. In general, however, if too many transitions are ignored, the gap of two configurations could be non-negligible in total. In nondeterministic qCCS, the number of ignored transitions is always constant, since the branching structures of the transition trees of configurations do not depend on a security parameter (Remark 4.3.2). In contrast, a probabilistic branch in the original qCCS possibly depends on a security parameter. For example, let n be a security parameter, the qubit length of a variable q be n, and $M = \sum_{i=1}^{2^n} i|i\rangle\langle i|$. The configuration $(\!|M[q; x].P, \rho|\!)$ performs a probabilistic transition with 2^n branches. If we define a notion of approximate bisimulation in the original qCCS similarly to the way in this chapter, we may need to assume (possibly by restricting the syntax) that the branching structures of the transition trees of configurations in qCCS do not depend on a security parameter.

Another important way to examine guarantees of the relation $\simeq$ is to define *observational (or testing) equivalence* [23, 2, 25, 11, 69] with approximation, and compare them. To define "approximate observational equivalence" in our simplified framework is future work. Even before approximation, it is still an open problem to define observational equivalence in the original qCCS that coincides with intuition [69], including formulation of the intuition. For example,

the configurations described in Section 3.2.1,

1. $(\!|1\rangle\langle 1|[q;x].P(q), |+\rangle\langle +|_q \otimes \rho^E|\!) \xrightarrow{\tau} \frac{1}{2} \bullet (\!|P(q), |0\rangle\langle 0|_q \otimes \rho^E|\!) \boxplus \frac{1}{2} \bullet (\!|P(q), |1\rangle\langle 1|_q \otimes \rho^E|\!)$
2. $(\!|\texttt{measure}[q].P(q), |+\rangle\langle +|_q \otimes \rho^E|\!) \xrightarrow{\tau} (\!|P(q), 1/2(|0\rangle\langle 0| + |1\rangle\langle 1|)_q \otimes \rho^E|\!)$

are not bisimilar although they intuitively do the same thing, because both $|1\rangle\langle 1|[q;x]$ and $\texttt{measure}[q]$ are measurement of q. Yasuda defined an observational equivalence that identifies several configurations which are intuitively equivalent, including two configurations similar to the above [69].

Chapter 5

Formal Verification of Quantum Cryptographic Protocols Using the Verifiers

5.1 Overview

We implemented a software tool to verify weak bisimilarity (being in the relation $\approx$) of configurations of the original qCCS [24]. In Chapter 3, we described the design and soundness of the verifier, which we call Verifier1. In Chapter 4, we defined the approximate bisimulation relation $\sim$ and extended the verifier to verify approximate bisimilarity (being in the relation $\sim$). We call the extended one Verifier2. We summarized the difference of them in Table 5.1. The package of

	Verifier1	Verifier2
syntax of processes	$\mathcal{P}$	$\mathcal{P}$
the outsider performs	TPCP maps	CP maps
rewriting using	*eqs*	*eqs* and *inds*
the relation to verify	$\approx$ in the original [24]	$\sim$ defined in Chapter 4
applied to verify	`BB84` $\approx$ `EDPbased`	`BB84` $\sim$ `EDPbased` $\sim$ `EDPideal`

Table 5.1: Difference of the Verifiers

the verifiers is available from the following URL. `http://hagi.is.s.u-tokyo.ac.jp/~tk/qccsverifer.tar.gz`

In this chapter, we describe the applications of the verifiers to Shor and Preskill's security proof of BB84. The formal verification consists of the following 2 steps.

1. BB84 and the EDP-based protocol are formalized as configurations `BB84` and `EDPbased`. Bisimilarity of the configurations is verified by Verifier1.

2. A new protocol EDP-ideal is defined. In the protocol, Alice and Bob initially share EPR-pairs, whose number is the same as that of the secret key's bit length. Apart from that, they executes the same protocol as the EDP-based protocol before creating their secret keys. When the protocol is not aborted, they create their secret keys just measuring their halves of pre-shared EPR-pairs. Since the pre-shared EPR pairs will not be influenced by Eve, Alice and Bob can create a shared secret key without leaking any information. The protocol EDP-ideal is formalized as a configuration `EDPideal`. Approximate bisimilarity of `EDPbased` and `EDPideal` is verified by Verifier2.

5.2 Input and Output for the Verifiers

5.2.1 Scripts

Input files, which we call scripts, contain the following descriptions. Although the algorithms are different, scripts for the verifiers are almost the same: only Verifier2 takes indistinguishability expressions.

Before formalizing processes and quantum states, symbols need to be declared.

- natural number symbols, which are elements of S_{nat}, in the form
  `nat` n`;`.
- channel names, which are elements of $qChan$, in the form
  `channel` c `:` n`;`,
  where n is a natural number symbol defined beforehand. Through channel c, quantum variables with length n are communicated.
- quantum variables, which are elements of $qVar$, in the form
  `qvar` q `:` n`;`,
  where n is the qubit-length of q.
- symbols of quantum states, which are elements of S_{stat}, in the form
  `dsym` X `:` n_1`,...,`n_k`;`.
  X is a quantum state which k quantum variables with qubit-length $n_1, ..., n_k$ are in. "dsym" stands for "density operator symbol".
- symbols of TPCP maps, which are elements of S_{op}, in the form
  `operator` op `:` n_1`,...,`n_k`;`.
  The operator op acts on quantum variables with qubit-length $n_1, ..., n_k$. The set S_{op} is the set of CP maps, including `proj`i for $i \in \{0, 1\}$, but a user can only define TPCP maps. Recall that for the process construction $op[\tilde{q}].P$, op is assumed to be a TPCP map.

Processes, quantum states, configurations, and equations on quantum states are then defined.

- A process is defined in the form
  `process` *process_name*
  P
  `end`.
- A quantum state is defined in the form
  `environment` *environment_name*
  ρ
  `end`.
- A configuration is defined in the form
  `configuration`
  `proc` *process_name*
  `env` *environment_name*
  `end`.
- An equation is defined in the form
  `equation` *equation_name*
  ρ = σ
  `end`,

where ρ and σ are quantum states. For a quantum state, description `__[`$\tilde{q}$`]` is permitted, which matches arbitrary quantum state of $\tilde{q}$.

Only for Verifier2, indistinguishability expressions on quantum states are defined.

- An indistinguishability expression is defined in the form
  `indistinguishable` *indexpression_name* n
  ρ `=` σ
  `end`,
  where n is a natural number symbol, and ρ and σ are quantum state symbols. For a quantum state, description `__[`$\tilde{q}$`]` is permitted, which matches arbitrary quantum state of $\tilde{q}$. Moreover, as a CP map, description `__[`$\tilde{q}$`]` is permitted, which matches an arbitrary CP map acting on $\tilde{q}$.

5.2.2 Outputs

If no option is set, the verifiers find two configuration in a script, verify their (approximate) bisimilarity using the defined equations (and indistinguishability expressions), and then output *true* or *false*. The verifiers have options with which they show information for debugging. The information is about the reason why the recursive procedure returns *false*. Concretely, for configurations $\{\!|P, \rho|\!\}$ and $\{\!|Q, \sigma|\!\}$, the verifiers show

1. P, Q, $\mathrm{qv}(P)$, and $\mathrm{qv}(Q)$ if $\mathrm{qv}(P) \neq \mathrm{qv}(Q)$.
2. $\mathrm{tr}_{\mathrm{qv}(P)}(\rho)$ and $\mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$ if $\mathrm{tr}_{\mathrm{qv}(P)}(\rho) \neq \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma)$
   (or $d(\mathrm{tr}_{\mathrm{qv}(P)}(\rho), \mathrm{tr}_{\mathrm{qv}(Q)}(\sigma))$ is not verified to be negligible.)
3. α, P, and Q if $Q \overset{\alpha}{\not\rightarrow}$ and $P \overset{\alpha}{\rightarrow}$, or $P \overset{\alpha}{\not\rightarrow}$ and $Q \overset{\alpha}{\rightarrow}$.

Especially, the information 2 can be used for finding equations that are necessary for the verification. For more details, readers can find user manual of the verifier contained in the package.

We next introduce as examples formal verification of correctness of the quantum teleportation protocol and the super dense coding protocol using Verifier1. The protocols were formally verified in [31] using bisimulation. Our way of formalization is slightly different from theirs, because we represent classical data as quantum data.

Example 5.2.1. *An example of formal verification of the quantum teleportation protocol is shown in Figure 5.1. The protocol is formalized as a configuration* `Tel`*. A configuration* `TelSpec` *is a specification of the protocol, which merely swaps input's and output's quantum states. With equation* `E1` *and* `E2`*,* `Tel` *and* `TelSpec` *are automatically proven to be bisimilar.*

The interpretations of natural number symbols, TPCP maps and quantum states in the script of Example 5.2.1 are as follows.

- The natural number symbol `2` is interpreted to the natural number 2. `m` is interpreted to an arbitrary natural number m.
- The quantum state symbols are interpreted as follows.
  - $[\![\texttt{EPR}]\!] = (\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger$
  - $[\![\texttt{ZERO}]\!] = |00\rangle\langle 00|$

```
nat 2;
nat m;
channel c : 2;
channel d : 1;
qvar q  : 1;
qvar q1 : 1;
qvar q2 : 1;
qvar x : 2;
qvar qE : m;
dsym EPR : 1,1;
dsym ZERO : 2;
dsym AFTER : 1,1,2;
dsym ANY : 1;
dsym EVE : m;
operator cnot : 1,1;
operator hadamard : 1;
operator measure : 1,1,2;
operator telproc : 2,1;
operator swap : 1,1;

process Tel_Proc
 ((cnot[q,q1].
   hadamard[q].
   measure[q,q1,x].
   c!x.discard(q,q1)
 ||
   c?y.telproc[y,q2].
   d!q2.discard(y)
 )/{c})
end

environment Tel_Env
 EPR[q1,q2] * ZERO[x]
 * ANY[q] * EVE[qE]
end
```

```
configuration Tel
 proc Tel_Proc
 env  Tel_Env
end

process TelSpec_Proc
 swap[q,q2].d!q2.discard(q1,x,q)
end

environment TelSpec_Env
 EPR[q1,q2] * ZERO[x]
 * ANY[q] * EVE[qE]
end

configuration TelSpec
 proc TelSpec_Proc
 env  TelSpec_Env
end

equation E1
 telproc[x,q2](measure[q,q1,x](
 hadamard[q](cnot[q,q1](
 EPR[q1,q2] * ZERO[x] * ANY[q])
 )))
 =
 ANY[q2] * AFTER[q,q1,x]
end

equation E2
 swap[q,q2](EPR[q1,q2] * ANY[q])
 =
 EPR[q1,q] * ANY[q2]
end
```

Figure 5.1: Formalization of Quantum Teleportation

  - $[\![\texttt{AFTER}]\!] = \frac{1}{4}(|0000\rangle\langle 0000| + |0101\rangle\langle 0101| + |1010\rangle\langle 1010| + |1111\rangle\langle 1111|)$
  - `ANY` and `EVE` are interpreted to arbitrary quantum states with dimension 1 and m.

- The TPCP map symbols are interpreted as follows.
  - $[\![\texttt{cnot}]\!]_{q,r}$ is CNOT operator in which the control qubit is q and the target qubit is r.
  - $[\![\texttt{hadamard}]\!]_s$ is Hadamard transformation to s.
  - $[\![\texttt{swap}]\!]_{t,u}$ is the operation swapping the state of t and u.
  - $[\![\texttt{measure}]\!](\cdot) = A(\cdot)A^\dagger$, where $A = |00\rangle\langle 00| \otimes I \otimes I + |01\rangle\langle 01| \otimes I \otimes X + |10\rangle\langle 10| \otimes X \otimes I + |11\rangle\langle 11| \otimes X \otimes X$.
  - $[\![\texttt{telproc}]\!](\cdot) = B(\cdot)B^\dagger$, where $B = |00\rangle\langle 00| \otimes I + |01\rangle\langle 01| \otimes X + |10\rangle\langle 10| \otimes Z + |11\rangle\langle 11| \otimes XZ$.

```
nat 2;
nat m;
channel c : 1;
channel d : 2;
qvar q  : 2;
qvar q1 : 1;
qvar q2 : 1;
qvar x  : 2;
qvar qE : m;
dsym EPR : 1,1;
dsym ZERO : 2;
dsym ANY2bit : 2;
dsym EVE : m;
operator cnot : 1,1;
operator hadamard : 1;
operator swap : 2,2;
operator measure : 1,1,2;
operator sdcproc : 2,1;

process Sdc_Proc
 ((sdcproc[q,q1].
   c!q1.discard(q)
 ||
   c?y.cnot[y,q2].
   hadamard[y].
   measure[y,q2,x].
   d!x.discard(y,q2)
 )/{c})
end

environment Sdc_Env
 EPR[q1,q2] * ZERO[x] *
 ANY2bit[q] * EVE[qE]
end

configuration Sdc
 proc Sdc_Proc
 env  Sdc_Env
end

process SdcSpec_Proc
 swap[q,x].d!x.discard(q,q1,q2)
end

environment SdcSpec_Env
 EPR[q1,q2] * ZERO[x] *
 ANY2bit[q] * EVE[qE]
end

configuration SdcSpec
 proc SdcSpec_Proc
 env  SdcSpec_Env
end

equation E1
 Tr[q1,q2,q](
  measure[q1,q2,x](
  hadamard[q1](
  cnot[q1,q2](
  sdcproc[q,q1](
   EPR[q1,q2] * ZERO[x]
   * ANY2bit[q])))))
 =
 ANY2bit[x]
end

equation E2
 swap[q,x](ZERO[x] * ANY2bit[q])
 =
 ANY2bit[x] * ZERO[q]
end
```

Figure 5.2: Formalization of Super Dense Coding

Under the above definitions of interpretations, validity of equations `E1` and `E2` are checked by hand.

Example 5.2.2. *An example of formal verification of the super dense cording protocol is shown in Figure 5.2. The protocol is formalized as a configuration* `Sdc`*. A configuration* `SdcSpec` *is a specification of the protocol, which merely swaps input's and output's quantum states. With equation* `E1` *and* `E2`*,* `Sdc` *and* `SdcSpec` *are automatically proven to be bisimilar.*

The interpretations of natural number symbols, TPCP maps and quantum states in the script of Example 5.2.2 are as follows.

- The natural number symbol `2` are interpreted to the natural number 2. `m` is interpreted to an arbitrary natural number m.

- The quantum state symbols are interpreted as follows.
  - $[\![\texttt{EPR}]\!] = (\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger$
  - $[\![\texttt{ZERO}]\!] = |00\rangle\langle 00|$
  - The symbol `ANY2bit` is interpreted to either $|00\rangle\langle 00|$, $|01\rangle\langle 01|$, $|10\rangle\langle 10|$, or $|11\rangle\langle 11|$.
  - The symbol `EVE` is interpreted to arbitrary quantum states with dimension m.
- TPCP map symbols are interpreted as follows.
  - $[\![\texttt{cnot}]\!]_{q,r}$ is CNOT operator in which the control qubit is q and the target qubit is r.
  - $[\![\texttt{hadamard}]\!]_s$ is Hadamard transformation to s.
  - $[\![\texttt{swap}]\!]_{t,u}$ is the operation swapping the state of t and u.
  - $[\![\texttt{measure}]\!](\cdot) = A(\cdot)A^\dagger$, where $A = |00\rangle\langle 00| \otimes I \otimes I + |01\rangle\langle 01| \otimes I \otimes X + |10\rangle\langle 10| \otimes X \otimes I + |11\rangle\langle 11| \otimes X \otimes X$.
  - $[\![\texttt{sdcproc}]\!](\cdot) = B(\cdot)B^\dagger$, where $B = |00\rangle\langle 00| \otimes I + |01\rangle\langle 01| \otimes X + |10\rangle\langle 10| \otimes Z + |11\rangle\langle 11| \otimes XZ$.

Under the above definitions of interpretations, validity of equations `E1` and `E2` are checked by hand.

5.3 Policies and Techniques of Formalization

Naming of Quantum Variables

There are principals Alice, Bob, and Eve in QKD protocols that we consider. For readability, quantum variables that Alice, Bob, and Eve initially have are appended with `_A`, `_B` and `_E` respectively in the scripts. The exception is that `EVE_2[r_B]` is initially the state of Eve's variable but she will be able to send it to Bob through `c2?r_B` because `c2` is public. This means that arbitrary quantum state that Eve has prepared can be sent to Bob through the public channel.

Formalization of Channels

As in general QKD protocols, three kinds of channels are used: public quantum channels, private classical channels, and public no-interpolate classical channels. Whether values themselves are quantum or classical does not matter here, since classical values are expressed as quantum states. A diagonal density operator can be regarded to represent a classical value. Let us say that a quantum variable q is assigned a classical value when q's quantum state is represented as a diagonal operator.

Since the syntax has channel restriction $P\backslash L$, formalization of the private channels is straightforward. The public quantum channels and the public no-interpolate classical channels are realized by copying the data. If a quantum variable q that is assigned a classical value is sent through a public no-interpolate channel c, this is formalized as

$$...\texttt{copy[}q\texttt{,}Q\texttt{]}\texttt{.}c\texttt{!}q\texttt{.}d\texttt{!}Q...\backslash\{..., c, ...\},$$

where Q is a new quantum variable, an operator `copy` copies the value of q to Q, and d is a new non-restricted channel. Concretely, the operator `copy[`q, Q`]`

initializes the state of Q to $|0\cdots0\rangle\langle0\cdots0|$ and apply CNOT with each qubit of q as the control and each qubit of Q as the target. The variable q will be securely sent through the restricted channel c and Eve obtains the same value accessing Q through the public channel d.

Aborting

Error checking and aborting is important in QKD protocols. When Alice and Bob decide to abort an execution of a protocol, what they do after the aborting is often not explicitly written [10, 65]. Although there are several possibilities, we merely write processes that do nothing after the aborting.

Outputting Secret Keys

In our formalization, the processes of QKD protocols send the completed secret keys to the outside and terminate keeping quantum variables that need not be sent to the outside. The purpose is to verify that the protocols produce the identical keys in BB84 and the EDP-based protocol (or approximately identical keys in the EDP-based protocol and EDP-ideal).

Congruence of the relation $\sim$ (Theorem 4.3.12, in Chapter 4) is useful in checking the behavior of the configurations of QKD under the presence of additional processes. Let us write the configurations of EDP-ideal and BB84 as $\{\!|EDPideal, \rho|\!\}$ and $\{\!|BB84, \sigma|\!\}$. By congruence, if $\{\!|EDPideal, \rho|\!\} \sim \{\!|BB84, \sigma|\!\}$ holds, which can be verified using Verifier2, we have

$$\{\!|EDPideal||P_{\mathrm{Alice}}||P_{\mathrm{Bob}}, \rho|\!\} \sim \{\!|BB84||P_{\mathrm{Alice}}||P_{\mathrm{Bob}}, \sigma|\!\},$$

where P_{Alice} and P_{Bob} are processes that run after given the secret keys from *EDPideal* or *BB84*. Moreover, we have

$$\{\!|(EDPideal||P_{\mathrm{Alice}}||P_{\mathrm{Bob}})\backslash\{cka, ckb\}, \rho|\!\} \sim \{\!|(BB84||P_{\mathrm{Alice}}||P_{\mathrm{Bob}})\backslash\{cka, ckb\}, \sigma|\!\},$$

where *cka* and *ckb* are secret channels to communicate Alice's and Bob's key. This suggests that we immediately have that P_{Alice} and P_{Bob} behave equivalently with secret keys created by $\{\!|EDPideal, \rho|\!\}$ and $\{\!|BB84, \sigma|\!\}$.

5.4 Formal Verification of Equivalence of BB84 and the EDP-based Protocol

The scripts of formalization of the EDP-based protocol and BB84 is shown in Figure 5.3 and 5.4.

5.4.1 Formalization of the EDP-based Protocol

The EDP-based protocol employs CSS quantum error correcting code (QECC), which is constructed from two classical linear codes C_1, C_2. CSS QECC can be parametrized with $u \in C_2$ and $v \in \{0,1\}^n - C_1$. We write $\mathrm{CSS}_{u,v}(C_1, C_2)$ for CSS code parametrized u and v that employ codes C_1 and C_2.

5.4.2 Symbols and Operators in the EDP-based Protocol

Quantum State Symbols

- Alice first prepares EPR pairs. Let quantum variables q and r be of the length `n`, where `n` interpreted as an arbitrary natural number n. `EPR[q,r]` is interpreted to EPR pairs $((\frac{|00\rangle+|11\rangle}{\sqrt{2}})(\frac{|00\rangle+|11\rangle}{\sqrt{2}})^\dagger)^{\otimes n}_{q,r}$.

```
process EDPbased
 ((hadamards[q2_A,r2_A,s_A].
  shuffle[q2_A,r2_A,t_A].
  c1!q2_A.c2!r2_A.c3?a_A.
  copyN[t_A,T_A].c4!t_A.d1!T_A.
  copy2n[s_A,S_A].c5!s_A.d2!S_A.
  measure[q1_A].
  c6?u_A.
  abort_alice[q1_A,u_A,b1_A].
  copy1[b1_A,b2_A].
  copy1[b1_A,B_A].
  c7!b1_A.d3!B_A.
  meas b2_A then
   css_projection[r1_A,x_A,z_A].
   copyn[x_A,X_A].
   css_decode[r1_A,x_A,z_A].
   measure[r1_A].
   c8!x_A.d4!X_A.
   c9!z_A.barrier!f_A.
   cka!r1_A.
   discard(q1_A,b2_A,a_A,
           u_A,v1_A,v_B)
  saem
 ||
  c1?q_B.c2?r_B.
  c3!a_B.d5!A_B.
  c4?t_B.unshuffle[q_B,r_B,t_B].
  c5?s_B.hadamards[q_B,r_B,s_B].
  measure[q_B].
  copyn[q_B,Q_B].c6!q_B.d6!Q_B.
  c7?b_B.
  meas b_B then
   c8?x_B.c9?z_B.
   css_syndrome[r_B,x_B,z_B,
                sx_B,sz_B].
   css_correct[r_B,sx_B,sz_B].
   css_decode[r_B,x_B,z_B].
   measure[r_B].
   barrier?f_B.
   ckb!r_B.
   discard(b_B,s_B,t_B,x_B,
           z_B,sx_B,sz_B,f_B)
  saem)/{c3, c4, c5, c6, c7, c8,
   c9, barrier})
end

environment EDPbased_ENV
 EPR[q1_A,q2_A] * EPR[r1_A,r2_A]
 * RND_2n[s_A] * RND_N[t_A] *
 Z_1[b1_A] * Z_1[b2_A] * Z_n[x_A]
 * Z_n[z_A] * Z_2n[S_A] *
 Z_N[T_A] *
 Z_1[B_A] * Z_n[X_A] * Z_1[f_A]
 * Z_1[a_B] * Z_1[A_B] * Z_n[Q_B]
 * Z_n[sx_B] * Z_n[sz_B]
 * EVE[q_E] * Z_n_n[v1_A,v_B]
 * EVE1[q_B] * EVE2[r_B]
end

configuration EDPbased
 proc EDPbased
 env EDPbased_ENV
end
```

Figure 5.3: Formalization of the EDP-based Protocol

- `RND_2n[`q`]` `RND_N[`r`]` are interpreted to $(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|)_q^{\otimes 2n}$ and $(\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|)_r^{\otimes N}$, where N is represented by a natural number symbol `N`. `N` is interpreted to $N = \lceil \log_2(2n!) \rceil$. This is the randomness to determine check bits.

- `Z_1[`q`]`, `Z_n[`r`]`, `Z_2n[`s`]`, and `Z_n_n[`t, u`]`, are interpreted to $|0\rangle\langle 0|_q$, $|0\rangle\langle 0|_r^{\otimes n}$, $|0\rangle\langle 0|_s^{\otimes 2n}$, and $|0\rangle\langle 0|_t^{\otimes n} \otimes |0\rangle\langle 0|_u^{\otimes n}$, respectively.

- `EVE`, `EVE1` and `EVE2` are arbitrarily interpreted. They express quantum states that are prepared by the adversary. `EVE` is one for a quantum variable with length `m`, where `m` is interpreted as an arbitrary natural number m. `EVE1` and `EVE2` are ones for quantum variables with length `n`.

TPCP Map Symbols

- `hadamards[`q, r, s`]` randomly performs Hadamard transformation to qubit-string q, r according to a bitstring s which serves as a seed of randomness.

```
process BB84
((hadamards[q2_A,r2_A,s_A].
 shuffle[q2_A,r2_A,t_A].
 c1!q2_A.c2!r2_A.c3?a_A.
 copyN[t_A,T_A].c4!t_A.d1!T_A.
 copy2n[s_A,S_A].c5!s_A.d2!S_A.
 c6?u_A.
 abort_alice[q1_A,u_A,b1_A].
 copy1[b1_A,b2_A].
 copy1[b1_A,B_A].
 c7!b1_A.d3!B_A.
 meas b2_A then
  cnot[r1_A,x_A].
  copyn[x_A,X_A].
  cnot_and_swap[x_A,r1_A].
  key[r1_A].
  c8!x_A.d4!X_A.
  barrier!f_A.
  cka!r1_A.
  discard(q1_A,b2_A,a_A,
          z_A,u_A,v1_A,v_B)
 saem
||
 c1?q_B.c2?r_B.
 c3!a_B.d5!A_B.
 c4?t_B.unshuffle[q_B,r_B,t_B].
 c5?s_B.hadamards[q_B,r_B,s_B].
 measure[q_B].
 copyn[q_B,Q_B].c6!q_B.d6!Q_B.
 c7?b_B.meas b_B then
 c8?x_B.
  measure[r_B].
  cnot[x_B,r_B].
  copyn[x_B,r_B].
  syndrome[r_B,sx_B].
  correct[r_B,sx_B].
  key[r_B].
  barrier?f_B.
  ckb!r_B.
  discard(b_B,s_B,t_B,
          x_B,sx_B,sz_B,f_B)
 saem)/{c3, c4, c5, c6,
        c7, c8, barrier})
end

environment BB84_ENV
 PROB[q1_A,q2_A] *
 PROB[r1_A,r2_A]
 * RND_2n[s_A] * RND_N[t_A]
 * Z_1[b1_A] * Z_1[b2_A]
 * RC1[x_A] * RC2[z_A]
 * Z_2n[S_A]
 * Z_N[T_A] * Z_1[B_A] * Z_n[X_A]
 * Z_1[f_A] *  Z_1[a_B]
 * Z_1[A_B]
 * Z_n[Q_B] * Z_n[sx_B]
 * Z_n[sz_B] * Z_n_n[v1_A,v_B]
 * EVE[q_E] * EVE1[q_B]
 * EVE2[r_B]
end

configuration BB84
 proc BB84
 env BB84_ENV
end
```

Figure 5.4: Formalization of BB84 Protocol

- `shuffle`$[q, r, s]$ randomly permutates the bits of qubit-string q, r according to the randomness s. In the formalization, `q_A` and `r_A` are supposed to be used as check bits and to generate secret keys. By this procedure, they are uniformly shuffled. Later, they are reverted by `unshuffle`$[q, r, s]$ procedure.

- `copy2n`$[q, r]$ copies the value of q with length `2n` to r, where q is supposed to be assigned a classical value. `copyN`$[q, r]$ and `copy1`$[q, r]$ are for quantum variables with length `N` and `1`.

- `measure`$[q]$ is the projective measurement of q.

- `abort_alice`$[q, r, s]$ compares two bitstrings q and r, and sets the value 0 to a bit s if the difference between q and r is lower than the threshold h, else sets the value 1 to s. The threshold h does not occur in the symbolic representation `abort_alice`$[q, r, s]$, but it is defined when the interpretation

$[\![$`abort_alice`$]\!]$ is defined. The threshold h can be defined appropriately so that the indistinguishability expressions are valid that are used to verify approximate bisimilarity of the EDP-based protocol and EDP-ideal.

- `css_projection`$[q, r, s]$ is the measurement of the observable of q's state that is described by the parity check matrix determined from C_1 and C_2 (Section 2.4.2, Step 7). EPR pairs q are converted to a random codeword of $\mathrm{CSS}_{x,y}(C_1, C_2)$, where parameters x, y are also uniformly distributed. The value of x and y are stored in r and s.

- `css_decode`$[q, r, s]$ decodes q as $\mathrm{CSS}_{x,y}(C_1, C_2)$ codeword when the value of r and s are x and y.

- `unshuffle`$[q, r, s]$ is the inverse of `shuffle`$[q, r, s]$.

- `css_syndrome`$[q, r, s, u, v]$ calculates the error syndrome of q as a codeword of $\mathrm{CSS}_{x,y}(C_1, C_2)$ when r and s have the value x and y, and stores the syndrome in u and v.

- `css_correct`$[q, u, v]$ is error correction with the syndrome stored in u, v.

5.4.3 Formalization of BB84

BB84 employs classical codes C_1 and C_2 with $C_2 \subseteq C_1$, which correspond to $\mathrm{CSS}_{x,y}(C_1, C_2)$ in the EDP-based protocol.

5.4.4 Symbols and Operators in BB84

Quantum State Symbols

- Alice first prepares two same random bitstrings. This initial state is represented by `PROB`$[q, r]$ with q for Alice and r for Bob, which is interpreted as $(\frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|)^{\otimes n}_{q,r}$.

- `RC1`$[q]$ is interpreted as $\sum_{u \in C_1} \frac{1}{|C_1|}|u\rangle\langle u|$.

- `RC2`$[q]$ is interpreted as $\sum_{v \in C_2} \frac{1}{|C_2|}|v\rangle\langle v|$.

TPCP Map Symbols

- `syndrome`$[q, r]$ calculates the error syndrome of q using as a codeword in C_1 and store the syndrome to r.

- `correct`$[q, r]$ corrects errors of q with the syndrome r.

- `key`$[q]$ calculates with respect to C_2 the coset of the value that is an element of C_1 and stored in q.

5.4.5 Equations for the Formal Verification

We defined 6 equations in Verifier1. They are described in Figure 5.5. The equations `E1`, `E2`, and `E3` are obtained formalizing the inferences in Shor and Preskill's security proof. The equations `E4`, `E5`, and `E6` are formalization of basic properties of linear operators.

```
equation E1
 measure[r1_A](
 css_decode[r1_A,x_A,z_A](
 copyn[x_A,X_A](
 css_projection[r1_A,x_A,z_A](
  EPR[r1_A,r2_A] *
  Z_n[x_A] * Z_n[z_A] *
  Z_n[X_A]))))
 =
 key[r1_A](
 cnot_and_swap[x_A,r1_A](
 copyn[x_A,X_A](
 cnot[r1_A,x_A](
  PROB[r1_A,r2_A] *
  RC1[x_A] *
  RC2[z_A] * Z_n[X_A]))))
end

equation E2
 measure[r_B](
 css_decode[r_B,x_A,z_A](
 css_correct[r_B,sx_B,sz_B](
 css_syndrome[r_B,x_A,z_A,
              sx_B,sz_B](
 cnot_and_swap[x_A,r1_A](
 copyn[x_A,X_A](
 cnot[r1_A,x_A](
  PROB[r1_A,r2_A] *
  __[r_B] * RC1[x_A] *
  RC2[z_A] * Z_n[sx_B] *
  Z_n[sz_B] * Z_n[X_A])))))))
 =
 key[r_B](
 correct[r_B,sx_B](
 syndrome[r_B,sx_B](
 copyn[x_A,r_B](
 cnot[r_B,x_A](
 measure[r_B](
 cnot_and_swap[x_A,r1_A](
 copyn[x_A,X_A](
 cnot[r1_A,x_A](
  PROB[r1_A,r2_A] * __[r_B]
  * RC1[x_A] * RC2[z_A]
  * Z_n[sx_B] * Z_n[sz_B] *
  Z_n[X_A])))))))))
end

equation E3
 measure[r2_A](
 css_decode[r2_A,x_A,z_A](
 css_correct[r2_A,sx_B,sz_B](
 css_syndrome[r2_A,x_A,z_A,
              sx_B,sz_B](
 cnot_and_swap[x_A,r1_A](
 copyn[x_A,X_A](
 cnot[r1_A,x_A](
  __[r1_A,r2_A] *
  RC1[x_A] * RC2[z_A] *
  Z_n[sx_B] * Z_n[sz_B] *
  Z_n[X_A])))))))
 =
 key[r2_A](
 correct[r2_A,sx_B](
 syndrome[r2_A,sx_B](
 copyn[x_A,r2_A](
 cnot[r2_A,x_A](
 measure[r2_A](
 cnot_and_swap[x_A,r1_A](
 copyn[x_A,X_A](
 cnot[r1_A,x_A](
  __[r1_A,r2_A] *
  RC1[x_A] * RC2[z_A] *
  Z_n[sx_B] * Z_n[sz_B] *
  Z_n[X_A]))))))))))
end

equation E4
 Tr[q1_A](EPR[q1_A,q2_A])
 =
 Tr[q1_A](PROB[q1_A,q2_A])
end

equation E5
 Tr[r1_A](EPR[r1_A,r2_A])
 =
 Tr[r1_A](PROB[r1_A,r2_A])
end

equation E6
 measure[q1_A](EPR[q1_A,q2_A])
 =
 PROB[q1_A,q2_A]
end
```

Figure 5.5: Equations for BB84 and the EDP-based Protocol

5.4.6 Experiment Result

Experiment 1

We ran Verifier1 with the input of `shor-preskill.scr`. We used a laptop with Intel Core i5 CPU M 460 @ 2.53GHz and 1GB memory. The transition tree of the EDP-based protocol has 621 nodes and 165 paths, and that of BB84 has 588 nodes and 165 paths. The verifier checked the bisimilarity of the two protocols in 30.38 seconds. The recursive procedure was called 753 times. The number of configurations in the history was 653 and history was hit 653 times. The number of application of each equation is described as follows. The equations E1, E2, E3, E4, E5, and E6 are applied 55, 24, 9, 73, 271, and 11 times, respectively.

5.5 Formal Verification of Security of the EDP-based Protocol

The last protocol EDP-ideal is a sort of cheating protocol. Alice and Bob initially share EPR pairs in the protocol. They execute the same protocol as the EDP-based protocol until the decision of continue or aborting by the result of the error checking. Only when they decide to continue, they create the secret keys using pre-shared EPR pairs instead of pairs obtained after the entanglement distillation protocol.

5.5.1 Formalization of EDP-ideal

The pre-shared EPR pairs are formalized as $\mathtt{EPR}[\mathtt{rx\_A}, \mathtt{rx\_B}]$. The code of EDP-ideal is almost the same as the EDP-based protocol. When Alice and Bob decide to continue the protocol, Alice creates her secret key from `rx_A` and renames to `r1_A` operating $\mathtt{create\_key}[\mathtt{rx\_A}, \mathtt{r1\_A}]$. Bob creates his key similarly by $\mathtt{create\_key}[\mathtt{rx\_B}, \mathtt{r\_B}]$.

5.5.2 Indistinguishability Expressions for the Verification

We defined 24 indistinguishability expressions Verifier2. One of the expressions `E1` is described in Figure 5.5.2. The expression's meaning is as follows.

1. The probability that they do not abort the protocol is negligibly close in the both protocols.
2. If the both protocols are not aborted, Alice's secret key that is created from halves of qubit pairs whose states are obtained after the entanglement distillation in the EDP-based protocol is indistinguishable from Alice's key that is created from EPR pairs.

The indistinguishability expressions are prepared for each Eve's choice: she can choose to interfere or not to interfere communications through the public quantum channels `c1` and `c2`. For example, if she interferes `c1` and does not interfere `c2`, a possible scheduling is as follows.

$$\begin{aligned} \cdots &\xrightarrow{\tau} \{\!|(\mathtt{c1!q2\_A.c2!r2\_A.}\cdots || \mathtt{c1?q\_B.c2?r\_B.}\cdots)\backslash\{\mathtt{c3}, \cdots\}, \rho|\!\} \\ &\xrightarrow{\mathtt{c1!q2\_A}}\xrightarrow{\mathtt{c1?q\_B}} \{\!|(\mathtt{c2!r2\_A.}\cdots || \mathtt{c2?r\_B.}\cdots)\backslash\{\mathtt{c3}, \cdots\}, \rho'|\!\} \\ &\xrightarrow{\tau} \{\!|(\cdots || \cdots)\backslash\{\mathtt{c3}, \cdots\}, \rho''|\!\} \end{aligned}$$

`E1` is for the case where Eve chooses the scheduling $\xrightarrow{\mathtt{c1!q2\_A}}\xrightarrow{\mathtt{c1?q\_B}}$ and $\xrightarrow{\mathtt{c2!r2\_A}}\xrightarrow{\mathtt{c2?r\_B}}$. The order of sending and receiving does not matter here. As for

```
process EDP-IDEAL
 ((hadamards[q2_A,r2_A,s_A].
  shuffle[q2_A,r2_A,t_A].
  c1!q2_A.c2!r2_A.c3?a_A.
  copyN[t_A,T_A].c4!t_A.d1!T_A.
  copy2n[s_A,S_A].c5!s_A.d2!S_A.
  measure[q1_A].
  c6?u_A.
  abort_alice[q1_A,u_A,b1_A].
  copy1[b1_A,b2_A].
  copy1[b1_A,B_A].
  c7!b1_A.d3!B_A.
  meas b2_A then
   css_projection[r1_A,x_A,z_A].
   css_decode[r1_A,x_A,z_A].
   copyn[x_A,X_A].
   measure[r1_A].
   c8!x_A.d4!X_A.
   c9!z_A.
   create_key[rx_A,r1_A].
   barrier!f_A.
   cka!r1_A.
   discard(q1_A,b2_A,
           a_A,u_A,rx_A)
  saem
 ||
  c1?q_B.c2?r_B.
  c3!a_B.d5!A_B.
  c4?t_B.unshuffle[q_B,r_B,t_B].
  c5?s_B.hadamards[q_B,r_B,s_B].
  measure[q_B].
  copyn[q_B,Q_B].c6!q_B.d6!Q_B.
  c7?b_B.
  meas b_B then
   c8?x_B.c9?z_B.
   css_syndrome[r_B,x_B,
                z_B,sx_B,sz_B].
   css_correct[r_B,sx_B,sz_B].
   css_decode[r_B,x_B,z_B].
   measure[r_B].
   create_key[rx_B,r_B].
   barrier?f_B.
   ckb!r_B.
   discard(b_B,s_B,t_B,x_B,z_B,
           sx_B,sz_B,f_B,rx_B)
  saem)/{c3, c4, c5, c6,
         c7, c8, c9, barrier})
end

environment EDP-IDEAL_ENV
 EPR[q1_A,q2_A] * EPR[r1_A,r2_A]
 * RND_2n[s_A] * RND_N[t_A]
 * Z_1[b1_A] * Z_1[b2_A]
 * Z_n[x_A]
 * Z_n[z_A] * Z_2n[S_A]
 * Z_N[T_A]
 * Z_1[B_A] * Z_n[X_A] * Z_1[f_A]
 * Z_1[a_B] * Z_1[A_B] * Z_n[Q_B]
 * Z_n[sx_B] * Z_n[sz_B]
 * EVE[q_E]
 * EVE1[q_B] * EVE2[r_B]
 * EPR[rx_A,rx_B]
end

configuration EDP-IDEAL
 proc EDP-IDEAL
 env  EDP-IDEAL_ENV
end
```

Figure 5.6: Formalization of EDP-ideal

this point, we needed 4 types of indistinguishability expressions for the cases where Eve interferes both c1, c2, only c1, only c2, and does not interfere both.

The indistinguishability expressions are also prepared for certain steps of the protocols: after completing the keys, Alice and Bob output their keys but who sends the first is non-deterministic. For instance, the quantum variable r_B is in the expression Tr[b1_A,b2_A,q1_A,q_B,r_B,rx_A,rx_B,s_A,t_A,x_A,z_A] in the first line and this is for the step where Alice has already sent her key r1_A to the outside by cka!r1_A but Bob has not yet his secret key r_B by ckb!r_B. As for this point, we needed 3 types of indistinguishability expressions for the cases where the outsider has obtained only Alice's key, only Bob's key, and both.

We have explained the reason why we needed $3 \times 4 = 12$ indistinguishability expressions. Finally, for each expression, we needed one equivalent expression obtained replacing the order of the CP maps, because the pattern matching algorithm for CP maps does not solve commutativity completely. Hence, we

```
indistinguishable E1 n
 Tr[b1_A,b2_A,q1_A,q_B,r_B,rx_A,rx_B,s_A,t_A,x_A,z_A](
  create_key[rx_A,r1_A](proj1[b1_A](measure[r1_A](
  copyn[x_A,X_A](css_decode[r1_A,x_A,z_A](
  css_projection[r1_A,x_A,z_A](proj1[b2_A](
  copy1[b1_A,B_A](copy1[b1_A,b2_A](
  abort_alice[q1_A,q_B,b1_A](measure[q1_A](
  copyn[q_B,Q_B](measure[q_B](
  hadamards[q_B,r_B,s_A](copy2n[s_A,S_A](
  unshuffle[q_B,r_B,t_A](copyN[t_A, T_A](
  __[q2_A,r2_A,q_E,q_B,r_B](
  shuffle[q2_A,r2_A,t_A](hadamards[q2_A,r2_A,s_A](
    EPR[q1_A,q2_A] * EPR[r1_A,r2_A] * EPR[rx_A,rx_B] *
    RND_2n[s_A] * Z_2n[S_A] * RND_N[t_A] * Z_N[T_A] *
    Z_1[b1_A] * Z_1[b2_A] * Z_1[B_A] * Z_n[Q_B] *
    Z_n[x_A] * Z_n[X_A] * Z_n[z_A] *
    __[q_B] * __[r_B] * __[q_E]
 ))))))))))))))))))))
 =
 Tr[b1_A,b2_A,q1_A,q_B,r_B,s_A,t_A,x_A,z_A](
  proj1[b1_A](measure[r1_A](
  copyn[x_A,X_A](css_decode[r1_A,x_A,z_A](
  css_projection[r1_A,x_A,z_A](proj1[b2_A](
  copy1[b1_A,B_A](copy1[b1_A,b2_A](
  abort_alice[q1_A,q_B,b1_A](measure[q1_A](
  copyn[q_B,Q_B](measure[q_B](
  hadamards[q_B,r_B,s_A](copy2n[s_A,S_A](
  unshuffle[q_B,r_B,t_A](copyN[t_A, T_A](
  __[q2_A,r2_A,q_E,q_B,r_B](
  shuffle[q2_A,r2_A,t_A](hadamards[q2_A,r2_A,s_A](
    EPR[q1_A,q2_A] * EPR[r1_A,r2_A] *
    RND_2n[s_A] * Z_2n[S_A] * RND_N[t_A] * Z_N[T_A] *
    Z_1[b1_A] * Z_1[b2_A] * Z_1[B_A] * Z_n[Q_B] *
    Z_n[x_A] * Z_n[X_A] * Z_n[z_A] *
    __[q_B] * __[r_B] * __[q_E]
  ))))))))))))))))))))
end
```

Figure 5.7: Indistinguishability Expression E1

prepared $12 \times 2 = 24$ expressions.

5.5.3 Experiment Result

Experiment 2

We performed the experiment in the same environment as the previous part of the formal verification described in Section 5.4. We ran Verifier2 with the input of `edp-edpideal.scr`, where the EDP based protocol and EDP-ideal are formalized. As for the transition tree, the both protocols have 621 nodes and 165 paths. Verifier2 checked the bisimilarity of the two protocols in 112.50 seconds. The recursive procedure was called 907 times. The number of configurations in

the history was 763 and history was hit 620 times. The number of application of each equation is described as follows. The equations E1, E2, E3, E1-2, E2-2, and E3-2 are applied 6, 12, 6, 12, 24, and 12 times, respectively. The equations F1, F2, F3, F1-2, F2-2, and F3-2 are applied 2, 4, 2, 4, 8, and 4 times, respectively. The equations G1, G2, G3, G1-2, G2-2, and G3-2 are applied 2, 4, 2, 4, 8, and 4 times, respectively. The equations H1, H2, H3, H1-2, H2-2, and H3-2 are applied 1, 2, 1, 2, 4, and 2 times, respectively.

Experiment 3

We ran Verifier2 with the input of `bb84-edp.scr`, which is identical to the script of Experiment 1. It checked the bisimilarity of the two protocols in 39.50 seconds. The recursive procedure was called 1039 times. The transition tree of the EDP-based protocol has 621 nodes and 165 paths, and that of BB84 has 588 nodes and 165 paths, which is the same result as the Experiment 1. The number of configurations in the history was 796 and history was hit 653 times. The number of application of each equation is described as follows. The equations E1, E2, E3, E4, E5, and E6 are applied 132, 24, 9, 73, 458, and 11 times, respectively.

Discussion about the Results

Although the number of the call of the recursive procedure is close, it took more time, 112.50 seconds, in Experiment 2 compared to 30.38 seconds in Experiment 3. There are following two reasons. The first is that the algorithm of the Verifier2 is more complex than that of Verifier1: wild card of CP map symbol `__[`$\tilde{q}$`]` is permitted in indistinguishability expressions. The algorithm to match the left-hand side of the expressions is more complex for the sake of the wild card matching. The second is that both the number of indistinguishability expressions and the sizes of them are larger. For each time of testing indistinguishability, the procedure checks for each indistinguishability expression whether it matches to the left-hand side of the objective quantum states.

Let us compare Experiment 1 and Experiment 3. Although we used the same input file, the number of calling the recursive procedure is different. This is because the existence or absence of the requirement that a branch caused by `meas` should be matched, which we mentioned in Section 4.4.1. With the requirement, to simulate a transition caused by `meas`, the number of transitions which are candidate for success of the simulation is more limited: Verifier1 only seeks a τ transition caused by `meas` by the same qubit variable.

Chapter 6

Conclusions

6.1 Automated Verification of Bisimilarity of Configurations

Impact of Automation

Our automatic verification methods broaden the range of application of qCCS. In security proofs, equivalence of protocols is often discussed. It can be described as bisimilarity but it is difficult to check by hand when state transitions of processes have many long branches. Besides, equality of outsider's views between two protocols must be checked in each step. Outsider's view is calculated from collective quantum state, which is possibly denoted by a huge matrix. One might prove bisimilarity with the insight of state transitions without tracing them. However, it is not always possible and possibly contradicts a purpose of formal methods, namely, to make implicit inferences in proofs explicit. Our verifiers do exhausting parts of proofs of behavioural equivalence: it checks correspondence of all state transitions up to invisible ones and equality of outsider's views using equations and indistinguishability expressions. Let us call equations and indistinguishability expressions *axioms* here. On the other hand, a user only has to examine the correctness of formalization of protocols and validity of axioms. It could be difficult to find all appropriate axioms for a proof immediately. The verifiers are also able to show quantum states, outsider's views, and/or processes when the recursive procedure returns *false*. With the information, a user can modify axioms to input.

Equations and Indistinguishability Expressions

Formal verification of validity of axioms is important but not in the scope of this thesis. Most of the axioms in Chapter 5 are obtained formalizing properties of CSS-QECC [18] or Lo and Chau's theorem [53] and their validity is not self-evident. Nevertheless, the validity can be verified as just *equality or negligible trace distance of density operators*. One does not have to consider *communication* of principals and *nondeterminism* of execution models to verify the axioms, because conditions about them are verified using the process calculus.

Application of a sequential quantum programming language such as QPL [64] is a possible way to verify axioms. In QPL's semantics, the programs are interpreted to TPCP maps. The bodies of symbolically-represented TPCP maps, such as `css_projection`$[q, r, s]$ and `css_syndrome`$[q, r, s]$ described in Chapter 5, are possibly formalized as QPL programs. Validity of axioms could be verified as equivalence or "indistinguishability" of the programs.

6.2 Congruent Approximate Bisimulation Relation

The notion of bisimulation proposed by Feng et al. [24] was applicable to verify equivalence of BB84 and the EDP-based protocol [46]. The configurations that are bisimilar in the original qCCS's definition behave equivalently from the outside. However, it seemed not to applicable directly to verify the security proof of the latter. To do it, it seemed to be a possible way to consider an ideal protocol and prove that they behave *almost* equivalently from an adversary Eve. In Chapter 4, we defined two approximate bisimulation relations $\sim_{\zeta,\eta}$ and $\sim$ on nondeterministic qCCS configurations, and studied properties of them. Some of the properties, such as those stated in Proposition 4.2.6, 4.3.6, Lemma 4.2.5, 4.3.4, and 4.3.7, are analogy of those of existing bisimulation relations [58, 31, 24]. As stated by Lemma 4.2.10, the relation $\sim_{\zeta,\eta}$ has a transitivity-like property. The relation $\sim$ is an equivalence relation, which is stated in Proposition 4.3.9. Furthermore, $\sim_{\zeta,\eta}$ and $\sim$ are closed under application of an arbitrary evaluation context as stated in Corollary 4.2.14 and 4.3.12. The property is useful in practice, concretely, when we consider multiple sessions of a protocol or its behavior employed as a primitive of another protocol.

We discussed guarantees and limitations in Section 4.5. The configurations in the relation $\sim$ reveal quantum states with negligible trace distance to the outsider after transitions. As described in Section 4.5.1 and Section 5.5, the notion of approximate bisimulation $\sim$ was applicable to verify formally the security proof of the EDP-based protocol. However, as described in 4.5.3, we do not have any precise definition of approximate bisimulation in the original qCCS. Even if we have $\{\!|P, \rho|\!\} \sim \{\!|Q, \sigma|\!\}$, it is not clear whether $(\!|\mathrm{cnv}(P), [\![\rho]\!]|\!)$ and $(\!|\mathrm{cnv}(Q), [\![\sigma]\!]|\!)$ behave "approximately indistinguishably" in the original qCCS. It is future work to define approximate bisimulation in the original qCCS and study end guarantees of the relation $\sim$ (and $\simeq$ defined in Section 4.5.3).

6.3 Formal Verification of Security Proofs of Quantum Cryptographic Protocols

We formally verified Shor and Preskill's security proof of BB84 [65] using our verifiers. Verifier checked bisimilarity of `BB84` and `EDPbased` and Verifier2 checked approximate bisimilarity of `BB84`, `EDPbased`, and `EDPideal`. To the best of our knowledge, this is the first work where *cryptographic security* of a quantum cryptographic protocol is mechanically verified using a software tool.

Shor and Preskill's security proof is simple compared to other proofs [56, 44], where more general execution models are assumed. Actually, Shor and Preskill's proof is clearly understood by researchers in quantum cryptography [56, 44, 52] and it seems not to be an emergent task to verify it formally. We consider our work as a step to apply formal methods practically to general quantum protocols.

6.4 Future Work

Although we applied our verifiers to Shor and Preskill's security proof, we did not to other proofs. For B92 [9] and the six-state protocol [17], which are QKD protocols, the proofs [68, 51] that are similar to Shor and Preskill's have been presented: the security of the objective protocol is reduced to an EDP-based protocol, and the latter is proven to be secure. Similarly to what we have done for BB84, it seems possible to make valid equations and indistinguishability expressions for B92 and the six-state protocol.

There are also several security proofs of BB84 with different assumptions. To broaden the range of formal verification using process calculi, we have to consider the way to verify proofs with different patterns of arguments.

As mentioned in Section 6.1, formal verification of validity of axioms is future work. Furthermore, to broaden the range of automation, the algorithms must be improved to test equality and indistinguishability of symbolically-represented partial traces using the axioms. In this thesis, we adopt a quite simple algorithm for the tests: each axiom is applied only once for each test in the order of user's definition. Although such simple algorithm is applicable to verify relatively simple verification objectives such as Shor and Preskill's security proof, improved strategies help us to verify more general ones. It is possible to improve our verifiers to call external procedures for *completion* [43, 38] such as Maxcomp [42] or mkbTT [63] to construct confluent and terminating term rewriting systems from equational systems consisting of the axioms. With a complete rewriting system, equality and indistinguishability of the symbolic representations are decided.

If quantum cryptographic protocols are formalized as configurations and proven (approximately) bisimilar, we expect they are (approximately) equivalent in purely quantum cryptographic sense. However, correspondence between bisimilarity and physical equivalence is only intuitively understood. Compared to the tools we introduced in Section 1.1.2, our verifiers seems to be relatively close to verifiers of quantum cryptographic proofs in that they do not idealize cryptographic primitives. To consider physical semantics of quantum process calculi is also future work.

References

[1] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *ACM SIGPLAN Notices*, volume 36, pages 104–115. ACM, 2001.

[2] Martín Abadi and Andrew D Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 36–47. ACM, 1997.

[3] Pedro Adao and Paulo Mateus. A process algebra for reasoning about quantum security. *Electronic Notes in Theoretical Computer Science*, 170:3–21, 2007.

[4] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, P Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, et al. The avispa tool for the automated validation of internet security protocols and applications. In *Computer Aided Verification*, pages 281–285. Springer, 2005.

[5] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella Béguelin. Computer-aided security proofs for the working cryptographer. In *Advances in Cryptology–CRYPTO 2011*, pages 71–90. Springer, 2011.

[6] Gilles Barthe, Benjamin Grégoire, Yassine Lakhnech, and Santiago Zanella Béguelin. Beyond provable security verifiable ind-cca security of oaep. In *Topics in Cryptology–CT-RSA 2011*, pages 180–196. Springer, 2011.

[7] Gilles Barthe, Benjamin Grégoire, and Santiago Zanella Béguelin. Formal certification of code-based cryptographic proofs. In *ACM SIGPLAN Notices*, volume 44, pages 90–101. ACM, 2009.

[8] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 544–553. ACM, 1994.

[9] Charles Henry Bennett. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters*, 68(21):3121, 1992.

[10] Charles Henry Bennett and Gilles Brassard. Quantum cryptography: Public-key distribution and coin tossing. *IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[11] Bruno Blanchet. A computationally sound mechanized prover for security protocols. *Dependable and Secure Computing, IEEE Transactions on*, 5(4):193–207, 2008.

[12] Bruno Blanchet. Automatically verified mechanized proof of one-encryption key exchange. Cryptology ePrint Archive, Report 2012/173, 2012. `http://eprint.iacr.org/`.

[13] Bruno Blanchet et al. An efficient cryptographic protocol verifier based on prolog rules. In *csfw*, volume 1, pages 82–96, 2001.

[14] Bruno Blanchet, Aaron D Jaggard, Andre Scedrov, and J-K Tsay. Computationally sound mechanized proofs for basic and public-key kerberos. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 87–99. ACM, 2008.

[15] Bruno Blanchet and David Pointcheval. Automated security proofs with sequences of games. In *Advances in Cryptology-CRYPTO 2006*, pages 537–554. Springer, 2006.

[16] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs# 1. In *Advances in Cryptology—CRYPTO'98*, pages 1–12. Springer, 1998.

[17] Dagmar Bruß. Optimal eavesdropping in quantum cryptography with six states. *Physical Review Letters*, 81(14):3018, 1998.

[18] Arthur Robert Calderbank and Peter Williston Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54(2):1098–1105, Aug 1996.

[19] Ran Canetti and Jonathan Herzog. Universally composable symbolic analysis of mutual authentication and key-exchange protocols. In *Theory of Cryptography*, pages 380–403. Springer, 2006.

[20] Iliano Cervesato, Aaron D Jaggard, Andre Scedrov, Joe-Kai Tsay, and Christopher Walstad. Breaking and fixing public-key kerberos. *Information and Computation*, 206(2):402–424, 2008.

[21] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology—CRYPTO'98*, pages 13–25. Springer, 1998.

[22] Timothy A. S. Davidson, Simon J Gay, Rajagopal Nagarajan, and Ittoop Vergheese Puthoor. Analysis of a quantum error correcting code using quantum process calculus. *EPTCS 95*, pages 67–80, 2012.

[23] Rocco De Nicola and Matthew CB Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34(1):83–133, 1984.

[24] Yuxin Deng and Yuan Feng. Open bisimulation for quantum processes. In Jos C.M. Baeten, Tom Ball, and Frank S. Boer, editors, *Theoretical Computer Science*, volume 7604 of *Lecture Notes in Computer Science*, pages 119–133. Springer Berlin Heidelberg, 2012.

[25] Yuxin Deng, Rob Van Glabbeek, Matthew Hennessy, and Carroll Morgan. Testing finitary probabilistic processes. In *CONCUR 2009-Concurrency Theory*, pages 274–288. Springer, 2009.

[26] The Coq development team. *The Coq Proof Assistant Reference Manual Version 8.3*.

[27] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[28] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003.

[29] Yuan Feng, Yuxin Deng, and Mingsheng Ying. Symbolic bisimulation for quantum processes. *arXiv preprint arXiv:1202.3484*, 2012.

[30] Yuan Feng, Runyao Duan, Zhengfeng Ji, and Mingsheng Ying. Probabilistic bisimulations for quantum processes. *Information and Computation*, 205(11):1608–1639, 2007.

[31] Yuan Feng, Runyao Duan, and Mingsheng Ying. Bisimulation for quantum processes. *SIGPLAN Not.*, 46(1):523–534, January 2011.

[32] David Galindo. Boneh-franklin identity based encryption revisited. In *Automata, Languages and Programming*, pages 791–802. Springer, 2005.

[33] Antoine Girard and George J Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 684–689, 2005.

[34] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[35] Jean Goubault-Larrecq, Catuscia Palamidessi, and Angelo Troina. A probabilistic applied pi–calculus. In *Programming Languages and Systems*, pages 175–190. Springer, 2007.

[36] Masami Hagiya and Yasuyuki Tsukada, editors. *Formal Approach to Information Security*. Kyoritsu Shuppan, 2010. Supervised by The Japan Society for Industrial and Applied Mathematics, the series of industrial and applied mathematics volume 1 (in Japanese).

[37] Shai Halevi. A plausible approach to computer-aided cryptographic proofs. *IACR Cryptology ePrint Archive*, 2005:181, 2005.

[38] Gérard Huet. A complete proof of correctness of the knuth-bendix completion algorithm. *Journal of Computer and System Sciences*, 23(1):11–21, 1981.

[39] Satoshi Ishizaka, Tomohiro Ogawa, Akinori Kawachi, Gen Kimura, and Masahito Hayashi. *Introduction to Quantum Information Science (***量子情報科学入門***)*. Kyoritsu Shuppan, 2012. (in Japanese).

[40] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70. ACM, 2005.

[41] Yoshihiko Kakutani. A logic for formal verification of quantum programs. In *Advances in Computer Science-ASIAN 2009. Information Security and Privacy*, pages 79–93. Springer, 2009.

[42] Dominik Klein and Nao Hirokawa. Maximal completion. 2011.

[43] Donald E Knuth and Peter B Bendix. Simple word problems in universal algebras. j. leech, editor, computational problems in abstract algebra, 263–297, 1970.

[44] Masato Koashi and John Preskill. Secure quantum key distribution with an uncharacterized source. *Physical review letters*, 90(5):057902, 2003.

[45] Takahiro Kubota, Yoshihiko Kakutani, Go Kato, and Yasuhito Kawano. A formal approach to unconditional security proofs for quantum key distribution. In *Unconventional Computation*, pages 125–137. Springer, 2011.

[46] Takahiro Kubota, Yoshihiko Kakutani, Go Kato, Yasuhito Kawano, and Hideki Sakurada. Application of a process calculus to security proofs of quantum protocols. *Proceedings of WORLDCOMP/FCS2012*, Jul 2012.

[47] Takahiro Kubota, Yoshihiko Kakutani, Go Kato, Yasuhito Kawano, and Hideki Sakurada. Automated verification of equivalence on quantum cryptographic protocols. *Symbolic Computation in Software Science*, page 64, 2013.

[48] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism 2.0: A tool for probabilistic model checking. In *Proceedings of First International Conference on the Quantitative Evaluation of Systems*, pages 322–323. IEEE, 2004.

[49] Marie Lalire. Relations among quantum processes: bisimilarity and congruence. *Mathematical Structures in Computer Science*, 16(3):407–428, 2006.

[50] Kim G Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and computation*, 94(1):1–28, 1991.

[51] Hoi-Kwong Lo. Proof of unconditional security of six-state quatum key distribution scheme. *Quantum Information and Computation*, 1(2):81–94, 2001.

[52] Hoi-Kwong Lo. Method for decoupling error correction from privacy amplification. *New Journal of Physics*, 5(1):36, 2003.

[53] Hoi-Kwong Lo and Hoi Fung Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Phys. Rev. Lett.*, 283(5410):2050–2056, Mar 1999.

[54] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, 1996.

[55] Dominic Mayers. Unconditional security in quantum cryptography. 1998.

[56] Dominic Mayers. Unconditional security in quantum cryptography. *J. ACM*, 48:351–406, May 2001.

[57] David A Meyer. Quantum strategies. *Physical Review Letters*, 82(5):1052, 1999.

[58] Robin Milner. *Communicating and mobile systems: the pi calculus*. Cambridge university press, 1999.

[59] Rajagopal Nagarajan, Nikolaos Papanikolaou, Garry Bowen, and Simon Gay. An automated analysis of the security of quantum key distribution. In *Proc. 3rd International Workshop on Security Issues in Concurrency (SecCo'05)*, 2005.

[60] Michael A. Nielsen and Issac L Chuang. *Quantum Computation and Quantum Information*. Ohmsha, 2004. Translated to Japanese by Tatsuya Kimura.

[61] Nikolaos K Papanikolaou. Model checking quantum protocols, 2009. Master's Thesis.

[62] Charles Rackoff and Daniel R Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology—CRYPTO' 91*, pages 433–444. Springer, 1992.

[63] Haruhiko Sato, Sarah Winkler, Masahito Kurihara, and Aart Middeldorp. Multi-completion with termination tools (system description). In *Proc. 4th IJCAR*, volume 5195 of *LNAI*, pages 306–312, 2008.

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

[65] Peter Williston Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85(2):441–444, Jul 2000.

[66] Victor Shoup. Oaep reconsidered. In *Advances in Cryptology—CRYPTO 2001*, pages 239–259. Springer, 2001.

[67] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

[68] Kiyoshi Tamaki, Masato Koashi, and Nobuyuki Imoto. Unconditionally secure key distribution based on two nonorthogonal states. *Physical review letters*, 90(16):167904, 2003.

[69] Kazuya Yasuda. Observational equivalence using schedulers for quantum processes. The University of Tokyo, 2014. Bachelor's Thesis.

[70] Mingsheng Ying, Yuan Feng, Runyao Duan, and Zhengfeng Ji. An algebra of quantum processes. *ACM Transactions on Computational Logic (TOCL)*, 10(3):19, 2009.

[71] Mingsheng Ying and Martin Wirsing. Approximate bisimilarity. In *Algebraic Methodology and Software Technology*, pages 309–322. Springer, 2000.

[72] Santiago Zanella-Beguelin, Gilles Barthe, Benjamin Grégoire, and Federico Olmedo. Formally certifying the security of digital signature schemes. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 237–250. IEEE, 2009.