Quantum Information and Variants of Interactive Proof Systems

by

Sarvagya Upadhyay

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

For nearly three decades, the model of interactive proof systems and its variants have been central to many important and exciting developments in computational complexity theory such as exact characterization of some well known complexity classes, development of probabilistically checkable proof systems and theory of hardness of approximation, and formalization of fundamental cryptographic primitives.

On the other hand, the theory of quantum information, which is primarily concerned with harnessing quantum mechanical features for algorithmic, cryptographic, and information processing tasks has found many applications. In the past three decades, quantum information has been used to develop unconditionally secure quantum cryptography protocols, efficient quantum algorithms for certain problems that are believed to be intractable in classical world, and communication efficient protocols.

In this thesis, we study the impact of quantum information on the models of interactive proof systems and their multi-prover variants. We study various quantum models and explore two questions. The first question we address pertains to the expressive power of such models with or without resource constraints. The second question is related to error reduction technique of such proof systems via parallel repetition.

The question related to the expressive power of models of quantum interactive proof systems and their variants lead us to the following results.

1. We show that the expressive power of quantum interactive proof systems is exactly PSPACE, the class of problems that can be solved by a polynomial-space deterministic Turing machines and that also admit a classical interactive proof systems. This result shows that in terms of complexity-theoretic characterization, both the models are equivalent. The result is obtained using an algorithmic technique known as the matrix multiplicative weights update method to solve a semidefinite program that characterizes the success probability of the quantum prover.
2. We show that polynomially many logarithmic-size unentangled quantum proofs are no more powerful than a classical proof if the verifier has the ability to process quantum information. This result follows from an observation that logarithmic-size quantum states can be efficiently represented classically and such classical representation can be used to efficiently generate the quantum state.
3. We also establish that the model of multi-prover quantum Merlin Arthur proof system, where the verifier is only allowed to apply nonadaptive unentangled measurement on each proof and then a quantum circuit on the classical outcomes, is no more powerful than QMA under the restriction that there are only polynomial number of outcomes per proof. This result follows from showing that such proof systems also admit a QMA verification procedure.

The question related to error reduction via parallel repetition lead us to following results on a class of two-prover one-round games with quantum provers and a class of multi-prover QMA proof systems.

1. We establish that for a certain class of two-prover one-round games known as XOR games, admit a *perfect* parallel repetition theorem in the following sense. When the provers play a collection of XOR games, an optimal strategy of the provers is to play each instance of the collection independently and optimally. In particular, the success probability of the quantum provers in the n-fold repetition of an XOR game G with quantum value $\omega_q(G)$ is exactly $(\omega_q(G))^n$.
2. We show a parallel repetition theorem for two-prover one-round unique games. More specifically, we prove that if the quantum value of a unique game is $1-\epsilon$, then the quantum value of n-fold repetition of the game is at most $(1-\Omega(\epsilon^2))^n$. We also establish that for certain class of unique games, the MOD_k games, the quantum value of the n-fold repetition of the game is at most $(1-\Omega(\epsilon))^n$. For the special case of XOR games, our proof technique gives an alternate proof of result mentioned above.
3. Our final result on parallel repetition is concerned with $\text{SepQMA}(m)$ proof systems, where the verifier receives m unentangled quantum proofs and the measurement operator corresponding to outcome *accept* is a fully separable operator. We give an alternate proof of a result of Harrow and Montanaro [HM10] that states that perfect parallel repetition theorem holds for such proof systems.

The first two results follow from the duality of semidefinite programs and the final result follows from cone programming duality.

Acknowledgments

Being a graduate student at Waterloo has been a very exciting and memorable six years of my life. There were moments when the going was very tough or moments when things fell in place. I have shared these moments with so many people that it is extremely difficult to choose and name a few.

I have the utmost respect for my adviser Richard Cleve. From the formative stages of this thesis, he has been a constant source of inspiration. Without his encouragement, guidance and support this thesis would have been a distant dream. My interactions with him has deeply influenced my approach towards life and research. Besides being an extraordinary researcher and an equally amazing teacher, he has also been an excellent mentor. He never let me waver from my ultimate goal. Even when I was getting restless and frustrated, his patient approach and presence was always a calming influence on me. He helped me realize the advantages of collaborative research and the need to approach a problem intuitively. Often he would spot a counterexample to my conjectures, which although frustrating helped me realize the importance of formulating problems. Even when I had correctly solved a problem, his role as "Devil's Advocate" always gave me new insights in my solution and the problem itself. He has been the best adviser I could possibly hope for.

I am extremely grateful to Rahul Jain and John Watrous. The brainstorming sessions with Rahul were always fruitful and I learned a lot from those sessions. His enthusiasm and optimism was one of the driving forces behind our fruitful collaboration. John was always there with his constructive comments and criticisms. I learned a lot from merely observing him during our weekly group meetings and interactions with him helped me learn about various topics in quantum information that later turned out to be extremely useful in my research. I would also take this opportunity to thank the rest of my committee members Ashwin Nayak, Mario Szegedy, and Stephen Vavasis. Ashwin's and Stephen's comments improved the presentation of my thesis a lot.

Most of my thesis is based on three excellent courses I took in Waterloo: Richard's "Introduction to Quantum Information Processing", Levent Tunçel's "Semidefinite Optimization", and John's "Theory of Quantum Information". I learned a lot from these courses and I enjoyed their individual teaching styles.

During the six years of my PhD, I had the pleasure of working with many great people, who provided me with interesting ideas. Some of these ideas resulted in publications. I want to thank my co-authors Richard Cleve, Sevag Gharibian, Rahul Jain, Zhengfeng Ji, Jamie Sikora, William Slofstra, Falk Unger, and John Watrous for all the hard work they put in our joint work. I also had interesting scientific conversations with people that did not lead to a publication. Among them were Andris Ambainis, Joe Fitzsimons, Tsuyoshi Ito, Iordanis Kerenidis, Frédéric Magniez, Oded Regev, Pranab Sen, Simone Severini,

and many others who shared their insights and knowledge with me. My special thanks to Ashwin Nayak for arranging my exchange visit to Paris, and Frédéric Magniez and Iordanis Kerenedis for hosting me at LIAFA during the visit.

My stay in Waterloo has been made memorable by my friends. My roommates were the closest people who were always there when I needed them. Aashish, Akhilesh, and Sachin have fed me appetizing meals and have helped me through difficult times. Aashish's morning tea was a routine part of my stay with him. Doing paperworks with him was always so easy and fun. Akhilesh has been always there with his wise remarks and serene presence. I have learned a great deal from him. Sachin is the one of the sweetest person I know, a very good friend and equally amazing host with his constant desire to play pranks. They never let me feel that I am away from my family. Niraj and Srinath have been two of my closest friends in Waterloo. Spending time with Niraj was always fun, and his anecdotes and stories (sometimes twisted at my expense) were worth listening. Srinath and I shared many coffee breaks in Waterloo and conversations on variety of topics in the last six years. Somehow we have managed to keep our conversations interesting and enjoyable. Over the last six years, I became friends with Abir, Adi, Aniket, Avisek, Chandru, Guru, Naveen, Prasad, Prashant and so many more, and I thank them for the amazing times we had. During the later part of my stay in Waterloo, I met Darya, Nikita, Prachi, Shubham, and Xili. Shubham has been a close friend and an amazing roommate. I will always remember the times I spent with him and Xili. Nikita and Prachi have cooked delicious meals on many occasions, and Darya has been there with her wishes, curiosity, and questions.

Even at work, I was lucky to have fellow graduate students with whom I shared trips and amazing moments in Waterloo. I cannot imagine hanging out with a better set of people than the CS/Math IQC gang. I will always remember our trips to Zürich and Montréal. I cannot forget the antics, jokes, pranks, and manipulating skills of our group members. Abel, Adam, Alessandro, Ansis, Jamie, Rajat, Robin, Sev, Stacey: I want to thank you all for all the fun times we had.

Finally, I cannot describe in words, my love towards my family. My mother has been a pillar of strength and it's her untiring support and encouragement that gave me the determination and will to pursue my endeavors. My father has been my role model and constant source of inspiration. My brother has always been with me and has supported me through all ups and downs in my life.

Contents

Chapter 1

Introduction

The unique feature that separates mathematics from other scientific disciplines is the notion of *proof*. It is the concept that establishes irrevocable truth of mathematical statements by following a sequence of logical rules. The notion of proving the correctness of an assertion can be thought of as a two stage process. The first stage is *proof generation*, which often involves creativity, insight, and patience. The second stage is *proof verification*, which is often considered as burdensome and mechanical task. However, the importance of the second stage cannot be underestimated for the task gives the proof its due value.

The concept of proof and its verification is ubiquitous in mathematics and have been widely studied from computational, logical, and philosophical viewpoint. This thesis focuses on the second stage, the process of proof verification, from a computational perspective. It is an important and active field of study for the last three decades with several applications. This thesis is an attempt to contribute to the vast body of work done on computational notions of verification. The chapter itself attempts to state the significance of proof verification.

What constitutes as a verification procedure? In simplistic terms, proof verification can be thought of as a deterministic procedure that takes the assertion and its proof as input and outputs *accept* or *reject* signifying the correctness or incorrectness of the proof. The procedure satisfies the following two properties.

1. *Completeness.* If the assertion is correct, then the verification procedure should accept a proof with certainty.
2. *Soundness.* If the assertion is incorrect, then no matter what proof is supplied to the verification procedure, it should reject with certainty.

This is the most common notion of verification, often referred as the standard notion of verification. The completeness condition states that if the assertion is valid, then there

exists a proof that will be accepted by the verification process. The soundness condition states that if the assertion is incorrect, then since no such proof of the validity of assertion exists, the verification procedure should always reject the purported proof. Taken together, the two conditions implies that a proof is accepted if and only if the assertion is correct. Of course, this means that the two properties can be summarized in a single sentence. However, having two separate conditions serves a purpose later on while we define computational notions of verification.

Throughout history, eminent mathematicians have tried to understand the impact of mechanical devices on theorem proving. The origin of this question can be traced back to Gottfried Leibniz, who dreamed of building a mechanical device that could tell the truth value of a mathematical statement. He realized that the first step was to formulate a formal language for the device and much of his subsequent work was devoted to achieving this goal. Indeed Leibniz was much ahead of his time.

Leibniz's quest was revisited again after more than two hundred years. In 1928, David Hilbert posed a fundamental question, known as the *Entscheidungsproblem*: Does there exists an algorithm that takes a mathematical statement as an input and produces an output *true* if and only if the statement is true and *false* if and only if the statement is false in finite time steps?

The existence of such an algorithm means that one can get a true/false answer to their favorite conjecture. A few years later, inspired by Gödel's incompleteness theorems, Alonzo Church [Chu36] and Alan Turing [Tur37] showed that existence of such an algorithm is impossible. The importance of their work is not limited to answering the question and exhibiting the power of human creativity over automated procedures. Their work formed the basis of computability theory, articulated the Church-Turing thesis, presented λ-calculi and Turing machines as universal models of computation, and helped introduce the notion of *efficient computation* and theory of computational complexity. Among some great achievements listed above, their work also introduced the notion of *decision problems* as a fundamental concept in computational complexity theory. Roughly speaking, a decision problem can be thought of as a partition of the set

$$\{0,1\}^* = \bigcup_{n \geq 0} \{0,1\}^n$$

into two disjoint subsets L_{yes} and L_{no}. The sets L_{yes} and L_{no} represent the *yes*-instances and *no*-instances of the problem.

With this notion in mind, the notion of *efficient computation* associates a Turing machine with a decision problem in the following way. Given any input string x, the Turing machine runs in polynomial-time in the size of the input ($|x|$) and decides correctly whether x is a yes-instance or a no-instance of the problem. The notion of efficient computation was

first introduced by Edmonds [Edm65], who explained the importance of polynomial-time and possible existence of other classes of problems that seemingly have no polynomial-time algorithm.

Edmonds seminal contribution led to significant breakthrough in theoretical computer science such as the introduction of P as the class of decision problems that admits a polynomial-time algorithm on a Turing machine and the theory of NP-completeness developed by Cook [Coo71], Karp [Kar72], and Levin [Lev73]. Indeed the theory of NP-completeness is based on the standard notion of proof verification discussed above with an added restriction that the verification procedure has to be efficient. These and other models of proof verification are discussed in the following section.

1.1 Notions of efficient verification

The notion of efficient proof verification implicitly involves two players, the *prover* and the *verifier*, and assumes the verification procedure to be efficient. It is one of the most widely studied notion in theoretical computer science with far reaching applications. One of the most fundamental questions in computational complexity asks for the intrinsic complexity of two tasks: proof generation and proof verification. Below we make this question more precise.

Let $(L_{\text{yes}}, L_{\text{no}})$ be a decision problem and x be an input string. Once the input string and the decision problem has been specified, the claim is to establish that $x \in L_{\text{yes}}$. The task of efficient proof generation is to come up with an efficient algorithm that generates a proof of the fact in the following sense. If the claim is true, then the algorithm outputs *accept* and if the claim is false, then the algorithm outputs *reject*. Indeed the class P is the class of decision problems that admit efficient proof generation procedure.

The task of efficient proof verification is to come up with an efficient algorithm that verifier whether a given proof of the claim $x \in L_{\text{yes}}$ is correct or not. In other words, the algorithm takes in two inputs, the string x and the proof, and efficiently decides the correctness of the claim. The class NP is the class of decision problems that admit efficient proof verification procedure. In this sense, the question of intrinsic complexity of the two tasks, proof generation and proof verification, is just the famous P-versus-NP problem, which asks whether the two classes are same.

Let us now turn to various notions of efficient proof verification. As discussed above, the complexity-theoretic abstraction of the standard notion of proof verification is the class NP. This proof system treats proofs as *static* objects, thereby limiting the role of the prover to merely sending the proof. A two-way dialog between the prover and the verifier allows us to treat proofs as *dynamic* objects, thereby increasing the role of the prover. On the

other hand, the verifier is allowed the luxury of two additional resources: *interaction* and *randomness*. Intuitively, interaction allows the verifier to ask questions to the prover who has to reply convincingly and randomness limits the prover's ability to predict the verifier's strategy. Of course, the verification procedure has to be efficient and satisfy the following three properties.

1. *Completeness.* For every x belonging to the yes-instances of the problem, there exists a behavior of the prover that causes the verifier to accept x as a yes-input with very high probability.
2. *Soundness.* For every x belonging to the no-instances of the problem, regardless of what behavior the prover adopts, the verifier accepts x as a yes-input with very small probability.
3. *Efficiency.* The complete verification procedure over all rounds of communication between the prover and the verifier should run in polynomial-time in the size of the string x.

The model is referred to as interactive proof system. Since the verifier is allowed to make error, the acceptance of a proof does not mean that it is a proof in strict mathematical sense. Instead the proof can be seen as an argument that convinces the verifier with overwhelming statistics that the assertion is correct. Also, if the verifier rejects at the end of the protocol, it does not mean that the assertion is incorrect. It means that the prover was unable to convince the verifier that the assertion is correct.

Interactive proof systems highlight the characteristics and role of the two players involved. The verifier is assumed to be curious, naïve, and reliable. He is supposed to protect his own interests, a characteristic that is best highlighted when studying *zero-knowledge proof systems*. The prover possess limitless computational power, is unreliable, and works towards protecting her interests. Both players proceed with the interaction keeping their respective interests in mind, which may be conflicting. In fact, the scenario becomes interesting when their interests are conflicting. That is, the proves tries to prove that an incorrect claim is valid and the verifier tries to catch her. Indeed, while devising an interactive protocol for a problem, our major focus is on this scenario.

The original motivation of studying such a model comes from cryptography and formalization of cryptographic primitives. There exist interactive proofs that are *zero-knowledge*, which means that the verifier gains no knowledge about the proof beyond the fact that the assertion is valid. This is a very peculiar aspect of interactive proof systems and one cannot hope for anything remotely similar in the standard model of proof verification.

Moving away from the original motivation, the model of interactive proof systems has led to the investigation of variants of the model. It is not an overstatement that the subsequent developments in the past twenty five years have huge impact on theoretical computer

science. One of the hallmarks of interactive verification is that the concept itself has raised many important questions, some of which are closely tied with their fundamental properties, while others tied with applications to other areas such as cryptography and hardness of approximation. Below we briefly discuss the variants of interactive proof systems and questions related to them that are pertinent to this thesis.

Variants of interactive proof systems: Apart from the single-prover interactive proof system as discussed above, there have been significant research done on different variants of the model. One such variant is the model of multi-prover interactive proof system wherein multiple provers try to convince a skeptical verifier that the assertion is correct. The nature of the provers towards each other is co-operating, however, they are not allowed to communicate with each other once the protocol starts. Allowing them to communicate with each other makes the model uninteresting for then the model reduces to single-prover model. Multi-prover interactive proof systems are at least as powerful as their single-prover counterpart for the verifier can concentrate on one prover and completely disregard the action of other provers. However, the model is much more powerful than the single prover case in complexity-theoretic sense because the verifier can use the non-communicating provers to his advantage.

Another variant that has been studied in literature is the model of interactive proof systems with competing provers. In this model, there are two provers with conflicting interest: one trying to prove that the assertion is correct, which we call the *yes*-prover and the other trying to prove the exact opposite, which we call the *no*-prover. The conflicting nature of the provers mean that even if one of the provers is honest, the other prover's behavior can be malicious and dishonest. Accordingly the completeness and soundness conditions of verification procedure accommodates for such type of behavior. More specifically, such proof system satisfies the following two properties apart from verifier being efficient.

1. *Completeness.* For every x belonging to the yes-instances of the problem, there exists a behavior of the *yes*-prover such that no matter what behavior the *no*-prover adopts, the verifier accepts x as a yes-input with very high probability.
2. *Soundness.* For every x belonging to the no-instances of the problem, there exists a behavior of the *no*-prover such that no matter what behavior the *yes*-prover adopts, the verifier accepts x as a yes-input with very small probability.

There are other interesting variants of the interactive proof system model which have been widely studied in the literature. One of them is the model of *zero-knowledge proof system*, where informally speaking, interaction helps the prover to prove an assertion without letting the verifier know anything about the proof. Such proof system are specially important from a cryptographic viewpoint and is rarely mentioned in this thesis.

Error reduction: Since the interactive verification allows the verifier to make error, a natural question to ask is whether one can make the proof systems robust against error probabilities. In other words, is it possible to reduce the error probabilities in both completeness and soundness case to an arbitrary small quantity without sacrificing the expressive power of the model. In most cases, if the prover adopts honest behavior and the assertion is correct, then the verifier always accepts the proof. Such proof systems are said to have *perfect* completeness. It is important to note that one cannot avoid the non-zero error probability in the soundness case without losing the expressive power of such proof systems.

There are many techniques employed to reduce the error probabilities of interactive proof systems and their variants. One such technique is the sequential repetition of the protocol. The acceptance criteria of the verifier is to accept if and only if all the repetitions are accepted. This technique works in case of perfect completeness at the expense of increase in the number of rounds of communication. Another way of error reduction (assuming perfect completeness) is to repeat multiple instances of the protocol in parallel and accept if and only if all the instances are accepted. In this case, the verifier executes multiple instances of the protocol simultaneously with the prover(s), which allows the prover(s) to gain additional knowledge in the sense that the prover(s) can reply with answer for each instance that may depend on all the questions received. However in most cases, the additional knowledge is not sufficient enough for the prover(s) to win all instances of the protocol with very high probability even if we assume that the success probability of the prover(s) for one execution of the protocol is very close to one. This error reduction technique is known as *parallel repetition* and is an extremely important technique that has found applications in the multi-prover case.

The power of interaction and randomization: It was established in the 1990's that the model of interactive proof systems, their competing prover variants, and the multi-prover variants are surprisingly very powerful models of computation. In other words, the process of verification with additional resources such as interaction and randomness allows efficient verification for a large class of problems with yes/no answers. For instance, the verifier in the most basic variant of interactive proof systems can verify the assertion for all such problems that are solvable using polynomial amount of space on a Turing machine. The verifier in the competing prover variants are even more powerful for he/she can verify the assertion for any problem that can be solved in exponential-time on a deterministic Turing machine. Finally, the verifier in the multi-prover variant can verify the assertion for any problem that can be solved in exponential-time on a non-deterministic Turing machine. From a complexity-theoretic viewpoint, the multi-prover variants is considered more powerful than the competing-prover variant which in turn is considered more powerful than the single-prover interactive proof system.

Interaction and randomization also allows us to conceptualize *zero-knowledge proofs*, which are of great theoretical and practical interest from a cryptography point of view. As informally stated above, these are proofs where the verifier gains no knowledge about the proof beyond the fact that the assertion is valid. This is indeed very peculiar to interactive proof systems and their multi-prover variants. Apart from cryptographic applications, the role of interaction and randomness have also resulted in surprising breakthroughs in theoretical computer science such as the development of (what we know as) the probabilistically checkable proof systems and a theory of hardness of approximation for optimization problems in computational setting.

With such a wide range of applications and the growing acceptability of quantum information as a resource, it is imperative to study the impact of quantum information on such proof systems. That is, either the prover(s) or the verifier or both are allowed to process quantum information. In scenarios where both the players can process quantum information, they can also exchange quantum information. Quantum information allows the verifier to simulate classical information and randomness and hence gives the verifier additional power over the verifier that can only process classical information. However, as we will explore some of the models in the light of quantum information, we will see that quantum resources are not always beneficial.

1.2 Quantum information as a resource

Quantum mechanics, a theory formulated in the last century, describes physical systems at an atomic scale which are otherwise indescribable by classical physics. Right from its inception the theory has been at the center of exciting developments in various disciplines of science. In the last three decades, a huge body of research have focused on exploiting quantum mechanical features to perform computational, communication, cryptographic, and information-theoretic tasks.

In early 1980s, Benioff [Ben82a, Ben82b] and Feynman [Fey82, Fey86] considered the task of simulating quantum mechanical phenomena on existing models of computation such as Turing machine and concluded that a new computational model based on the laws of quantum mechanics is required to hopefully perform such tasks. This led to a seminal work by David Deutsch [Deu85] in which the concept of quantum computer was made precise in terms of quantum Turing machines. The theory of quantum computation was subsequently developed by seminal work of Bernstein and Vazirani [BV97] (the preliminary version of their paper appeared in 1993), and Yao [Yao93]. After Feynman's proposed idea of building a computer based on laws of quantum mechanics, a huge body of research has been done on quantum computational model and the implications of quantum mechanical

features to various disciplines of theoretical computer science. Indeed, quantum information have found applications in different areas, few of which are briefly mentioned below. Our intention is not to provide a comprehensive list of applications, which is clearly beyond the scope of the thesis, but to briefly mention few of the most important developments that led to huge interest in quantum information. Of course, we still list a handful of those applications, which should be not be treated as the only set of important developments. Indeed, the list is due to the highly biased opinion of ours.

Quantum information in algorithms: The idea of a model of computation based on the principles of quantum mechanics is highly interesting, but what is even more interesting is that they can seemingly violate the *extended Church-Turing thesis*. The extended Church-Turing thesis hypothesizes that any efficiently computable function is efficiently computable of a Turing machine with access to randomness. One of the crowning achievements of quantum information is an efficient quantum algorithm for factoring and discrete-log problem [Sho97], based on Simon's algorithm [Sim97], which seemingly violates the extended Church-Turing thesis. The fact that such an algorithm seemingly violates the thesis is based on an intractability assumption that factoring cannot be solved efficiently on a classical computer with access to randomness. In fact, the best known classical algorithm for factoring, the general number field sieve method, runs in time exponential in some power of the input-size. The quantum algorithm for factoring led to other algorithmic breakthroughs, where similar speed ups were obtained [Hal07].

Quantum information in cryptography: While Shor's algorithm is simply a marvelous achievement and generated tremendous interest in quantum information, it also makes some existing cryptosystems insecure. Should large scale quantum computers become a reality, they will break existing public-key cryptosystems that are based on intractability assumptions like hardness of factoring or discrete-log problem. To salvage the damage done by a quantum computer (from a cryptographer's point of view), one can use cryptographic protocols based on quantum information for secure data transmission. Indeed, such a quantum key distribution protocol was proposed by Bennett and Brassard [BB84] in 1984, much before Shor's algorithm came into existence. Unlike the classical public-key cryptosystems, the quantum key distribution protocol is known to be unconditionally secure [BBBMR06, May01, SP00].

One of the key impact of quantum information has been the development of next generation classical cryptosystems that are believed to be unbreakable even on a quantum computer. In other words, these cryptosystems have so far resisted attacks based on quantum computers. For instance, lattice-based cryptography has generated tremendous interest among cryptographers for its simplicity, efficient implementation, and the fact that it is believed to be secure against quantum attacks [GM02].

Quantum information as proof technique: Much of the applications of quantum information is derived from using quantum information in algorithmic, computational, communication, cryptographic, and information-theoretic tasks. However, quantum information has been successful as a powerful analytical tool too. Its use have led to proving theorems and in some cases providing simpler proof of the existing theorems. One of the earliest applications of quantum arguments in theoretical computer science was proving an exponential lower bound on two-query locally decodable codes [KdW03]. There are few instances where quantum techniques have also been used in providing a simpler proof of important theorems [Aar05, deW06]. Other examples include lower bounds in communication complexity and polynomial approximations. For a more comprehensive overview on such applications of quantum techniques, we refer to the survey [DdW11] and the references therein.

At the heart of these exciting and interesting results are the mystical quantum mechanical features: quantum superposition, quantum interference, and quantum entanglement. Understanding the full potential of these features has been one of the biggest challenges of the last century and our current understanding largely depends on the problem in hand. This thesis, which focuses on specific questions related to quantum interactive proof systems and their variants, can be seen as an endeavor towards understanding the impact of quantum information on the models of interactive proof systems and their variants. Given the enormity of both the fields, the results in this thesis can be seen as partial progress towards understanding the implications of quantum mechanical features to computational complexity theory.

1.3 Results presented in the thesis

The focus of this thesis is on the impact of quantum information on the models of interactive proof systems and their variants. The role of quantum information in these models have been investigated in various papers over the past twelve years and although a lot of results have been known, still our current understanding of these models is incomplete. The results presented in this thesis builds upon the earlier works and focuses on two broad questions: expressive power of the models based on quantum interactive proof systems and their variants, and error reduction via parallel repetition. This section summarizes the main results presented in this thesis.

Introductory chapters: The next three chapters following the current chapter are chapters introducing the basic knowledge required to understand the thesis. Our treatment to

each of the chapters is based on the results presented in the thesis and although extensive, they are not comprehensive. We have pointed our references that give a comprehensive overview of the topics discussed in the next three chapters.

Chapter 2 focuses on the mathematical preliminaries pertinent to this thesis. The first section of the chapter is devoted to linear algebra preliminaries wherein we introduce the notations used throughout the thesis. The second section introduces quantum information and provides required background to understand the subsequent chapters. The final section summarizes the notations used in the thesis for readers who wish to skip the details presented in the previous two sections. Our treatment to these topics are more than what is required for this thesis and are presented for the sake of completeness. Chapter 3 introduces semidefinite programs, the duality theory of semidefinite programs, and their applications in this thesis both as an algorithmic and an analytical tool. It mentions all the facts on semidefinite programs that are necessary to follow the rest of the thesis. Chapter 4 gives a brief overview on computational complexity theory and introduces the relevant complexity classes. It also discusses the models of interactive proof systems and their multi-prover variants in greater details.

Expressive power of QIP: Chapter 5 is devoted to the first result of this thesis. This result characterizes the expressive power of quantum interactive proof systems, wherein the prover and the verifier can exchange and process quantum information, in terms of PSPACE, the class of problems that can be solved in polynomial-space on a Turing machine [JJUW09]. It has been known for twenty years that the model of interactive proof systems are sufficiently powerful to characterize PSPACE [LFKN92, Sha92, She92]. Our result establishes that the the classical and quantum models of interactive proof systems are equivalent.

To establish the desired result we make use of semidefinite programming characterization of an equivalent yet simple model of quantum interactive proof systems, which we refer to as *single-coin quantum interactive proof systems* [MW05], and an alternate characterization of PSPACE. We show that the semidefinite program can be "solved" by a fast parallel algorithm in the size of the program, which is exponential in the size of the input. It is known that such an algorithm can be simulated by a Turing machine that uses polynomial-space [Bor77], thereby establishing the result. We remark that the algorithm for semidefinite program is based on *matrix multiplicative weights update method* [AK07, Kal07, WK06], a technique that has been used to come up with faster algorithms for certain semidefinite programs.

Parallel repetition of quantum XOR games: Chapter 6 establishes a perfect parallel repetition theorem for a certain class of two-prover one-round games, known as XOR

games[CSUU08]. These are games where the verifier asks questions to each prover who respond with a bit, and the verifier's verdict is based on the parity of the bits received . In the quantum setting, the communication channel between the provers and the verifier is classical, however, the provers are allowed to share quantum entanglement among themselves. The main result of this chapter is that when the provers are allowed to process quantum information, the optimum success probability of the provers for simultaneously playing a collection of XOR games is exactly the product of the individual optimal success probabilities.

Our proof uses semidefinite programming techniques. We cast the *quantum bias* of an XOR game as a semidefinite program based on earlier results [CHTW04a, Tsi80, Tsi85] and show that the quantum bias of the parities of two XOR games G_1 and G_2, which we denote $G_1 \oplus G_2$, is equal to the product of quantum biases of the games G_1 and G_2. This property is established via semidefinite programming duality. We then relate this property to parallel repetition of XOR games via Fourier analysis. Our parallel repetition theorem together a result of Cleve, Høyer, Toner and Watrous [CHTW04a] also establishes a parallel repetition theorem for classical XOR games, which has been subsequently generalized by Rao [Rao08] and proven to be tight by Raz [Raz11].

Parallel repetition of unique games: Chapter 7 generalizes the results presented in Chapter 6 to unique games, which are two-prover one-round games wherein conditioned on verifier's acceptance the answer of the first prover uniquely determines the answer of the second prover and likewise the answer of the second prover uniquely determines the answer of the first prover. We show that a parallel repetition theorem for such games where the provers are allowed to share quantum resources [Upa07]. Our parallel repetition theorem is not perfect in the sense explained above. Nevertheless, the bounds obtained for unique games are tight. This result was obtained independently of Kempe, Regev and Toner [KRT08].

Our proof technique uses semidefinite programming duality and the quantum rounding procedure discussed in the reference [KRT08]. We first establish that any feasible solution of a semidefinite program relaxation of quantum value of the game can be rounded to obtain a valid quantum strategy for the provers with very good success probability. Then using duality of semidefinite programs, we establish that the optimum of the relaxation corresponding to the n-fold repetition of the game G, which we denote $G^{\otimes n}$, is no more than $(\bar{\sigma}_q(G))^n$, where $\bar{\sigma}_q(G)$ is the optimum of the relaxation corresponding to game G. Combining both these facts, we obtain the desired parallel repetition theorem. For certain special class of unique games, we also obtain stronger parallel repetition theorem.

QMA variants with polynomially many provers: Chapter 8 concerns with the model

of one-message quantum interactive proof systems, known as quantum Merlin-Arthur proof systems, and their multi-prover variants. In their multi-prover variants, the provers are only allowed to send quantum proofs that are guaranteed to be unentangled with other quantum proofs. We consider three variants of multi-prover quantum Merlin-Arthur proof systems. The first variant is $\mathrm{QMA}_{\log}(\mathrm{poly})$ wherein each proof is logarithmic-size in the size of the input. The second variant is BellQMA(poly) wherein the verifier's computation is a two-stage process as follows. The verifier first applies unentangled, nonadaptive measurements on each proof and then applies another quantum circuit on the outcomes of each measurement to decide whether to accept or reject. The third variant is SepQMA(poly), where the verifier's measurement operator corresponding to outcome *accept* is a *fully separable operator* across the proofs. In each model, we do not place any restriction on the number of provers.

The results mentioned in this chapter are as follows. We show that the first variant $\mathrm{QMA}_{\log}(\mathrm{poly})$ is equivalent to MQA, the collection of problems that admits a classical proof and quantum verification procedure. For the second variant, we show that if the number of measurement outcomes per proof is at most polynomial in the size of the input, then the power of such proof system is no more than QMA, the collection of promise problems admitting a quantum Merlin-Arthur proof system. The final result shows that perfect parallel repetition holds for SepQMA(poly), a result previously obtained by Harrow and Montanaro [HM10]. However, our proof technique is different from theirs in the sense that we use duality of cone programming to establish the result.

The thesis is organized keeping in mind that readers familiar with the required background and notations used in the thesis can directly follow them without going through other results. For a summary of the results presented and future work, we refer the readers to Chapter 9.

Chapter 2

Mathematical Preliminaries

The purpose of this chapter is to summarize the basics of linear algebra and quantum information relevant to the thesis. In the process, we familiarize ourselves with the notations used in this thesis. The chapter itself is divided into three sections.

- In Section 2.1, we give an extensive overview of the linear algebra preliminaries. This is by no means a comprehensive treatment on the topic and readers are referred to excellent texts on linear algebra [Rom08] and matrix analysis [Bha97, HJ85] for further details.
- In Section 2.2, we discuss the quantum information facts pertinent to the thesis. For a more comprehensive treatment on the topic, we refer to the following lecture notes [Wat08], and excellent texts by Kitaev, Shen and Vyali [KSV02], and Nielsen and Chuang [NC00].
- In Section 2.3, we summarize the notations used in the thesis for the benefit of the readers already familiar with the basics of linear algebra and quantum information.

2.1 Linear algebra preliminaries

This sections intends to summarize the basic facts from linear algebra and notations used throughout the thesis. The notations and conventions followed in this thesis are heavily borrowed from the following lecture notes [Wat08]. The section is divided into three subsections devoted to the following: complex Euclidean spaces, linear operators acting on complex Euclidean spaces, and linear super-operators that map operators acting on one space to operators acting on another space.

2.1.1 Vector spaces

The collection of N tuple of complex numbers $\mathbb{C}^N$ for $N > 0$ forms a vector space over the field of complex numbers. Such vector spaces are referred to as complex Euclidean spaces. We identify any such vector space by script uppercase letters $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$. The dimension of a vector space $\mathcal{X}$, when not explicitly stated, is denoted $\dim(\mathcal{X})$. The real vector space is denoted $\mathbb{R}^N$. The vectors are denoted by lowercase Roman letters u, v, and w. For any vector $u \in \mathcal{X} = \mathbb{C}^N$, we typically associate the vector as a column matrix

$$\begin{pmatrix} u[1] \\ u[2] \\ \vdots \\ u[N] \end{pmatrix}.$$

Here $u[i]$ denotes the i-th co-ordinate of the vector u. For a vector u, the *dual* of u is denoted

$$u^* = (\overline{u[1]}, \overline{u[2]}, \ldots, \overline{u[n]})$$

where $\overline{u[i]}$ is the *complex conjugate* of $u[i]$. For two vectors u and v in a complex Euclidean space $\mathcal{X} = \mathbb{C}^n$, their (standard) inner product is defined as

$$\langle u, v \rangle := \sum_{i=1}^{N} \overline{u[i]} v[i] = u^* v.$$

The inner product is conjugate-linear in the first argument and linear in the second argument. The inner product equips the underlying vector space with a norm. The standard *Euclidean norm* of a vector $u \in \mathcal{X}$ is

$$\|u\| = \sqrt{\langle u, u \rangle} = \left(\sum_{i=1}^{N} |u[i]|^2 \right)^{1/2}.$$

The *Cauchy-Schwarz inequality* states that

$$|\langle u, v \rangle| \leq \|u\| \cdot \|v\|$$

with equality if and only if $u = \lambda v$ for $\lambda \in \mathbb{C}$. The Euclidean norm is a special case of the class of *p-norms* for $p \in [1, \infty]$, which are defined as

$$\|u\|_p := \left(\sum_{i=1}^{N} |u[i]|^p \right)^{1/p}$$

for any vector $u \in \mathcal{X}$ and $p \in [1, \infty)$. For $p = \infty$, the corresponding norm is defined as

$$\|u\|_\infty := \max\big\{\, |u[i]| : i \in [N] \big\}.$$

The Euclidean norm corresponds to $p = 2$. This thesis uses Euclidean norm and (occasionally) 1-norm.

A collection of vectors $\big\{u_i : i \in [M]\big\} \subset \mathcal{X}$ is said to be an *orthogonal* set of vectors if for every distinct choices of $i, j \in [M]$, it holds that

$$\langle u_i, u_j \rangle = 0.$$

The notation $[M]$ denotes the set $\{1, 2, \ldots, M\}$. In addition to orthogonality condition, if the Euclidean norm of every vector is 1, then the set is said to be an *orthonormal* set of vectors. For any complex Euclidean space $\mathcal{X}$, an orthogonal (and consequently an orthonormal) set of vectors always exists for $1 \leq M \leq N$. For $M = N$, the set spans $\mathcal{X}$ in the sense that any vector in $\mathcal{X}$ can be written as a linear combination of the orthogonal vectors in the set. The *standard basis* of $\mathcal{X}$ is denoted by the elements of the set $\{e_i : i \in [N]\}$, where

$$e_i[j] = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j. \end{cases}$$

For any two complex Euclidean spaces $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$ with the following standard bases

$$\big\{e_1, e_2, \ldots, e_N\big\} \subset \mathcal{X} \qquad \text{and} \qquad \big\{f_1, f_2, \ldots, f_M\big\} \subset \mathcal{Y}$$

the complex Euclidean space $\mathcal{X} \otimes \mathcal{Y}$ is the space of dimension NM with the following standard bases:

$$g_{i,j} = e_i \otimes f_j \qquad \text{for all} \qquad i \in [N] \text{ and } j \in [M].$$

The complex Euclidean space $\mathcal{X} \otimes \mathcal{Y}$ is called the *tensor product* of vector spaces $\mathcal{X}$ and $\mathcal{Y}$. For any two vectors $u \in \mathcal{X}$ and $v \in \mathcal{Y}$, the vector $u \otimes v$ is called the tensor product of u and v. Not all vectors in $\mathcal{X} \otimes \mathcal{Y}$ can be written in the form of $u \otimes v$. However, the following theorem, called the *singular value theorem* (also known as *Schmidt decomposition* in literature), gives a nice characterization of a vector $w \in \mathcal{X} \otimes \mathcal{Y}$ in terms of vectors of the form $u \otimes v$.

Theorem 1. *Let $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$ be two complex Euclidean spaces. For any vector $w \in \mathcal{X} \otimes \mathcal{Y}$, there exist non-negative real numbers λ_i associated with orthonormal sets of unit vectors*

$$\{u_i : i \in [\min\{N, M\}]\} \subset \mathcal{X} \qquad \textit{and} \qquad \{v_i : i \in [\min\{N, M\}]\} \subset \mathcal{Y}$$

such that

$$w = \sum_{i=1}^{\min\{N,M\}} \sqrt{\lambda_i} u_i \otimes v_i.$$

The tensor product of three such spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ is defined by first taking the tensor product of $\mathcal{X}$ and $\mathcal{Y}$ and then taking the tensor product of $\mathcal{X} \otimes \mathcal{Y}$ with $\mathcal{Z}$. The tensor product of more than three vector spaces is inductively defined in a similar fashion.

For two complex Euclidean spaces $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$, the direct sum $\mathcal{X} \oplus \mathcal{Y}$ defines another complex Euclidean space of dimension $N + M$. For any two vectors $u \in \mathcal{X}$ and $v \in \mathcal{Y}$, the vector $u \oplus v$ is called the *direct sum* of u and v. The standard basis of $\mathcal{X} \oplus \mathcal{Y}$ is given by

$$\left\{ \begin{pmatrix} e_i \\ 0 \end{pmatrix} \cup \begin{pmatrix} 0 \\ f_j \end{pmatrix} \right\}$$

for all $i \in [N]$ and $j \in [M]$. The direct sum of three such spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ is defined by first taking the direct sum of $\mathcal{X}$ and $\mathcal{Y}$ and then taking the direct sum of $\mathcal{X} \oplus \mathcal{Y}$ with $\mathcal{Z}$. The direct sum of more than three vector spaces is inductively defined in a similar fashion.

2.1.2 Linear operators

Each pair of vectors $u \in \mathcal{X}$ and $v \in \mathcal{Y}$ induces a *linear operator* or simply an *operator* $vu^* : \mathcal{X} \to \mathcal{Y}$ defined by the action

$$vu^* w = \langle u, w \rangle v.$$

Given two complex Euclidean spaces $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$, the collection of all linear operators of the form

$$A : \mathcal{X} \to \mathcal{Y}$$

is denoted $\mathrm{L}(\mathcal{X}, \mathcal{Y})$. The collection $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ forms a vector space over complex numbers. The dimension of the vector space $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ is the product of the dimensions of the two vector spaces $\mathcal{X}$ and $\mathcal{Y}$:

$$\dim(\mathrm{L}(\mathcal{X}, \mathcal{Y})) = \dim(\mathcal{X})\dim(\mathcal{Y}) = NM.$$

Throughout the rest of the subsection, we denote $\dim(\mathcal{X}) = N$ and $\dim(\mathcal{Y}) = M$. The set of operators of the form

$$A : \mathcal{X} \to \mathcal{X}$$

is denoted $\mathrm{L}(\mathcal{X})$. The identity operator acting on $\mathcal{X}$ is denotes $\mathbb{I}_{\mathcal{X}}$. An operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is naturally represented by an $M \times N$ matrix, whose (i, j) entry is is denoted

$A[i,j]$ and defined as $\langle f_i, Ae_j \rangle$, where

$$\big\{e_i : i \in [N]\big\} \subset \mathcal{X} \qquad \text{and} \qquad \big\{f_i : i \in [M]\big\} \subset \mathcal{Y}$$

denote the sets of standard bases of $\mathcal{X}$ and $\mathcal{Y}$, respectively. For the rest of the thesis, we will use the terms operator and matrix interchangeably and will not differentiate between an operator and its matrix representation. That is, we will use the same uppercase later to identify an operator as itself and its matrix representation. This does not cause any ambiguity and the usage of both the terms will be clear from the context.

Given an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$, the operator $A^* \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$ is the unique linear operator that satisfies the relationship

$$\langle u, Av \rangle = \langle A^* u, v \rangle$$

for all $u \in \mathcal{X}$ and $v \in \mathcal{Y}$. Alternately, the operator A^* is the operator whose matrix representation is obtained by the relationship

$$A^*[i,j] := \overline{A[j,i]}$$

for all $j \in [M]$ and $i \in [N]$. The operator A^* is called the *adjoint* of A. For an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$, the operators

$$\overline{A} \in \mathrm{L}(\mathcal{X}, \mathcal{Y}) \qquad \text{and} \qquad A^\top \in \mathrm{L}(\mathcal{Y}, \mathcal{X})$$

are the operators whose matrix representation are defined as

$$\overline{A}[i,j] := \overline{A[i,j]} \qquad \text{and} \qquad A^\top[i,j] := A[j,i]$$

for all $i \in [N]$ and $j \in [M]$. The operators $\overline{A}$ and $A^\top$ are called *entry-wise conjugate* and *transpose* of A. If every matrix entry of an operator A is real, then it holds that $A = \overline{A}$ and $A^* = A^\top$.

For the space of operators $\mathrm{L}(\mathcal{X}, \mathcal{Y})$, the Hilbert-Schmidt inner product (or inner product, for short) of two operators A and B is defined as

$$\langle A, B \rangle := \mathrm{Tr}(A^* B) := \sum_{i=1}^{N} (A^* B)[i,i] = \sum_{i=1}^{M} \sum_{j=1}^{N} \overline{A[j,i]} B[i,j]$$

where $\mathrm{Tr}(X)$ is called the *trace* of the operator X. Note that the operation $\mathrm{Tr}(A^* B)$ is well defined because the operator $A^* B$ maps $\mathcal{X}$ to itself. One way of viewing the inner product on the space of operators is by writing the operators as vectors in a complex Euclidean space of dimension NM, and then applying the standard inner product on the resulting

vectors. In more precise terms, define

$$\mathrm{vec} : \mathrm{L}(\mathcal{X}, \mathcal{Y}) \to \mathcal{Y} \otimes \mathcal{X}$$

as the unique *linear* mapping that maps the operator $f_i e_j^* \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ to $f_i \otimes e_j \in \mathcal{Y} \otimes \mathcal{X}$. This mapping is clearly linear and bijective, which means that every operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ uniquely determines a vector $u_A \in \mathcal{Y} \otimes \mathcal{X}$ such that $\mathrm{vec}(A) = u_A$. Moreover,

$$\langle A, B \rangle = \langle \mathrm{vec}(A), \mathrm{vec}(B) \rangle$$

for all $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$.

As in the case of vectors, one can define tensor product of two or more than two operators. For two operators $A \in \mathrm{L}(\mathcal{X}_1, \mathcal{Y}_1)$ and $B \in \mathrm{L}(\mathcal{X}_2, \mathcal{Y}_2)$, their tensor product associate another operator $A \otimes B$ that lives in the vector space $\mathrm{L}(\mathcal{X}_1 \otimes \mathcal{X}_2, \mathcal{Y}_1 \otimes \mathcal{Y}_2)$ and is defined as

$$A \otimes B := \begin{pmatrix} A[1,1]B & A[1,2]B & \cdots & A[1,M_1]B \\ A[2,1]B & A[2,2]B & \cdots & A[2,M_1]B \\ \vdots & \vdots & \ddots & \vdots \\ A[N_1,1]B & A[N_1,2]B & \cdots & A[N_1,M_1]B \end{pmatrix}$$

where $\dim(\mathcal{X}_1) = N_1$ and $\dim(\mathcal{Y}_1) = M_1$. An alternate way of defining $A \otimes B$ is to define it as a unique linear map that satisfies the relationship

$$(A \otimes B)(u \otimes v) = (Au) \otimes (Bv)$$

for all $u \in \mathcal{X}_1$ and $v \in \mathcal{Y}_1$. For three operators $A \in \mathrm{L}(\mathcal{X}_1, \mathcal{Y}_1)$, $B \in \mathrm{L}(\mathcal{X}_2, \mathcal{Y}_2)$, and $C \in \mathrm{L}(\mathcal{X}_3, \mathcal{Y}_3)$, one defines the tensor product of them by first taking the tensor product of A and B and then taking the tensor product of $A \otimes B$ and C. The tensor product of more than three operators is defined similarly.

Relevant sets of operators

Most operators we encounter in this thesis are of the form

$$A : \mathcal{X} \to \mathcal{X}$$

and hence their matrix representation is a square matrix. This section intends to briefly summarize the various classes of operators that are extremely relevant to the thesis. For an operator $A \in \mathrm{L}(\mathcal{X})$ and a non zero vector $u \in \mathcal{X}$, if $Au = \lambda u$ for some $\lambda \in \mathbb{C}$, then λ is said to be the *eigenvalue* of A and u is said to be the corresponding *eigenvector* of A. An

operator $A \in \mathrm{L}(\mathcal{X})$ is *normal* if

$$AA^* = A^*A.$$

The class of normal operators are exactly those operators for which the spectral theorem as given below holds.

Theorem 2. *Let $A \in \mathrm{L}(\mathcal{X})$ be a normal operator and $(\lambda_1(A), \lambda_2(A), \ldots, \lambda_N(A))$ be the tuple of eigenvalues of A. Then there exists an orthonormal basis $\{u_1, u_2, \ldots, u_N\} \subset \mathcal{X}$ such that*

$$A = \sum_{i=1}^{N} \lambda_i(A) u_i u_i^*.$$

Note that the spectral decomposition is in general *not* unique. For example, the identity operator $\mathbb{I}_{\mathcal{X}}$ has a spectral decomposition for any orthonormal basis. The spectral theorem allows us to define functions of the form $f : \mathbb{C} \to \mathbb{C}$ on normal operators as follows:

$$f(A) := \sum_{i=1}^{N} f(\lambda_i(A)) u_i u_i^*.$$

Although there can be more than one spectral decompositions of an operator A, the operator $f(A)$ is *unique* and consequently well defined. The spectral theorem allows us to define the Schatten p-norms of any operator, which is discussed later in this section. Next we briefly mention few special classes of normal operators that will be used throughout the thesis.

Hermitian operators: A normal operator $A \in \mathrm{L}(\mathcal{X})$ is *Hermitian* if $A = A^*$. Such operators are also known as *self-adjoint operators* in literature. The set of Hermitian operators acting on a complex Euclidean space $\mathcal{X}$ is denoted $\mathrm{Herm}(\mathcal{X})$. The eigenvalues of any Hermitian operator are real and we write

$$\lambda(A) = (\lambda_1(A), \ldots, \lambda_N(A))$$

to denote the N-tuple of eigenvalues of A (counting multiplicities), sorted from largest to smallest:

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_N(A).$$

The set of Hermitian operators form a vector space of dimension N^2 over the field of real numbers. If $\{e_1, e_2, \ldots, e_N\}$ denotes the standard basis of $\mathcal{X}$, then one example of a basis of $\mathrm{Herm}(\mathcal{X})$ is

$$\big\{e_i e_i^* : 1 \leq i \leq N\big\} \quad \cup \quad \big\{e_i e_j^* + e_j e_i^* : 1 \leq i < j \leq N\big\} \quad \cup \quad \big\{\imath e_i e_j^* - \imath e_j e_i^* : 1 \leq i < j \leq N\big\}$$

where $\iota = \sqrt{-1}$. We refer to them as the *standard basis* for the space of Hermitian operators. The operator E_{ij} denotes the Hermitian operator with (i, j) and (j, i) entries 1, and 0 elsewhere. Similarly, the operator F_{ij} denotes the Hermitian operator with (i, j) entry ι, (j, i) entry $-\iota$, and 0 elsewhere. An important subset of Hermitian operators are symmetric operators whose representations have only real entries. The set of symmetric operators, denoted $\mathrm{Herm}\left(\mathbb{R}^N\right)$, also form a vector space over real numbers. However, the dimension of such a vector space is $N(N+1)/2$.

Finally, we note that Hermitian operators are closed under addition and real scalar multiplication. If a Hermitian operator is invertible, then its inverse is also a Hermitian operator.

Positive semidefinite operators: A Hermitian operator $A \in \mathrm{L}(\mathcal{X})$ is *positive semidefinite* if all the eigenvalues of A are non-negative. That is,

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_N(A) \geq 0.$$

The set of positive semidefinite operators are denoted $\mathrm{Pos}(\mathcal{X})$. The notation $A \succeq 0$ indicates that A is positive semidefinite, and more generally the notations $B \preceq A$ and $A \succeq B$ indicate that $A - B \succeq 0$ for Hermitian operators A and B. The positive semidefinite inequality induces a partial order on the set of Hermitian operators. There are many equivalent ways of describing positive semidefinite operators summarized in the theorem below.

Theorem 3. *Let $A \in \mathrm{Herm}(\mathcal{X})$. Then the following are equivalent:*

1. *The operator A is positive semidefinite.*
2. *All the eigenvalues of A are non-negative.*
3. *There exists an operator $B \in \mathrm{L}(\mathcal{X})$ such that $A = B^*B$.*
4. *For every choice of $u \in \mathcal{X}$, $\langle u, Au \rangle \geq 0$.*
5. *There exists a collection of vectors $\{u_1, u_2, \ldots, u_n\} \subset \mathcal{X}$ such that $A[i, j] = \langle u_i, u_j \rangle$.*
6. *For any set $S \subseteq [N]$, the symmetric restriction of A defined by the set S, A_S, with entries*

$$A_S[i, j] = \begin{cases} A[i, j] & \text{if} \quad i, j \in S \\ 0 & \text{otherwise.} \end{cases}$$

   is positive semidefinite.

The diagonal entries of positive semidefinite operators are always non-negative and consequently its trace is non-negative. If the diagonal entry of a positive semidefinite operator is

0, then every entry in the row and column corresponding to the diagonal entry must be 0. The alternate ways of describing positive semidefinite operators prove to be useful in different situations. For instance, a consequence of item 3 is that for two positive semidefinite operators P and Q, their inner product $\langle P, Q\rangle$ is always greater than zero with equality if and only if $PQ = 0$.

Note that Theorem 3 holds for positive semidefinite operators over symmetric operators also. For such operators items 3, 4, and 5 hold for $\mathcal{X} = \mathbb{R}^N$. The set of symmetric positive semidefinite operators acting on $\mathbb{R}^N$ is denoted $\mathrm{Pos}\left(\mathbb{R}^N\right)$. Finally, we note that for a symmetric operator $A \in \mathrm{Herm}\left(\mathbb{R}^N\right)$, if

$$2A[i,i] \geq \sum_{j=1}^{N} |A[i,j]|$$

for every $i \in [N]$, then A is positive semidefinite. Such operators are called *diagonally dominant operators*.

Positive definite operators: A positive semidefinite operator $A \in \mathrm{L}(\mathcal{X})$ is *positive definite* if all its eigenvalues are strictly positive. That is,

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_N(A) > 0.$$

The set of positive definite operators are denoted $\mathrm{Pd}(\mathcal{X})$. The notation $A \succ 0$ indicates that A is positive definite, and more generally the notations $B \prec A$ and $A \succ B$ indicate that $A - B \succ 0$ for Hermitian operators A and B. As in the case of positive semidefinite operators, there are alternate ways of describing positive definite operators summarized in the following theorem.

Theorem 4. *Let $A \in \mathrm{Herm}(\mathcal{X})$. Then the following are equivalent:*

1. *The operator A is positive definite.*
2. *All the eigenvalues of A are strictly positive.*
3. *There exists a non-singular operator $B \in \mathrm{L}(\mathcal{X})$ such that $A = B^*B$.*
4. *For every choice of $u \in \mathcal{X}/\{0\}$, $\langle u, Au\rangle > 0$.*
5. *There exists a collection of linearly independent vectors $\{u_1, u_2, \ldots, u_n\} \subset \mathcal{X}$ such that $A[i,j] = \langle u_i, u_j\rangle$.*

The diagonal entries of positive definite operators are always strictly positive and consequently its trace is strictly greater than 0. Moreover, if a positive semidefinite operator P makes zero inner product with a positive definite operator then $P = 0$.

As before, Theorem 4 holds for positive definite operators over symmetric operators, where items 3, 4, and 5 hold for $\mathcal{X} = \mathbb{R}^N$. The set of symmetric positive definite operators acting on $\mathbb{R}^N$ is denoted $\mathrm{Pd}\left(\mathbb{R}^N\right)$. Finally, for a diagonally dominant operator $A \in \mathrm{Pos}\left(\mathbb{R}^N\right)$, if

$$2A[i,i] > \sum_{j=1}^{N} |A[i,j]|$$

for every $i \in [N]$, then A is positive definite.

Projection operators: A positive semidefinite operator $A \in \mathrm{L}(\mathcal{X})$ is called an *orthogonal projection* or simply *projection* if it also satisfies an additional property that $A^2 = A$. Alternately, any Hermitian operator with eigenvalue either 0 or 1 is a projection operator. The projection operator are usually denoted by uppercase Greek letters Π, Δ, and Λ. By spectral theorem, any projection operator Π can be written as

$$\Pi = \sum_{i=1}^{M} u_i u_i^*.$$

Denoting $\mathcal{V} = \mathrm{span}\{u_1, u_2, \ldots, u_M\} \subseteq \mathcal{X}$, one obtains that Π acts as an identity operator on $\mathcal{V}$ and that $\mathcal{V}$ is an M-dimensional subspace of $\mathcal{X}$. The projection on the whole space $\mathcal{X}$ is the identity operator $\mathbb{I}_{\mathcal{X}}$.

Density operators: A positive semidefinite operator $A \in \mathrm{L}(\mathcal{X})$ is called a *density operator* if its trace is equal to one. The set of density operators are denoted $\mathrm{D}(\mathcal{X})$ and they form a convex compact set. The elements of this set will usually be denoted by lowercase Greek letters ρ, σ, and ξ. By the spectral theorem, any density operator ρ can be written as

$$\rho = \sum_{i=1}^{N} \lambda_i u_i u_i^*$$

where $\lambda_1, \lambda_2, \ldots, \lambda_N$ sum up to one. One can view density operators as probability distributions over rank-one projection operators. As mentioned earlier, spectral decomposition is not unique and hence different distributions over rank-one projection operators may be represented by the same density operator. In other words, the mapping

$$\{\lambda_i, u_i u_i^* : i \in [N]\} \longrightarrow \sum_{i=1}^{N} \lambda_i u_i u_i^*$$

is surjective but not injective.

Linear isometry and unitary operators: An operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is a *linear isometry*, or simply an *isometry* if $A^*A = \mathbb{I}_{\mathcal{X}}$. This condition is satisfied if and only if $\dim(\mathcal{Y}) \geq \dim(\mathcal{X})$. The set of linear isometries are denoted $\mathrm{U}(\mathcal{X}, \mathcal{Y})$. Any linear isometry A preserves the inner product between any two vectors and consequently the Euclidean norm of the vectors. In other words,

$$\langle Au, Av \rangle = \langle u, v \rangle \qquad \text{and} \qquad \|Au\| = \|u\|.$$

When $\dim(\mathcal{X}) = \dim(\mathcal{Y})$, the operator is called a *unitary operator* and the set of such operators is denoted $\mathrm{U}(\mathcal{X})$.

Norms of operators

As in the case of vectors, the Hilbert-Schmidt inner product equips the space with a norm similar to the Euclidean norm for vectors. This norm is a type of a class of norms known as Schatten p-norms. For any $p \geq 1$, the Schatten p-norms of any operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ are defined as

$$\|A\|_p := \left[\mathrm{Tr}\left((A^*A)^{p/2}\right)\right]^{1/p}.$$

Clearly A^*A is Hermitian and hence normal. Therefore, by the spectral theorem (Theorem 2)

$$A^*A = \sum_{i=1}^{N} \lambda_i(A^*A) u_i u_i^*$$

for some orthonormal set of vectors $\{u_1, u_2, \ldots, u_N\} \subset \mathcal{X}$. As a consequence of the above, we have

$$\|A\|_p = \left(\sum_{i=1}^{N} \left(\lambda_i(A^*A)\right)^{p/2}\right)^{1/p}.$$

The cases $p = 1$ and $p = 2$ are also known as *trace norm* (or 1-*norm*) and *Frobenius norm* (or 2-*norm*) respectively:

$$\|A\|_1 = \sum_{i=1}^{N} \sqrt{\lambda_i(A^*A)} \qquad \text{and} \qquad \|A\|_2 = \left(\sum_{i=1}^{N} \lambda_i(A^*A)\right)^{1/2}.$$

For $p = \infty$,

$$\|A\|_\infty = \max\left\{\sqrt{\lambda_i(A^*A)} : i \in [N]\right\}.$$

This norm is called the *infinity norm* or *operator norm* of A. An alternate way of defining

the infinity norm of A is as follows:

$$\|A\|_\infty = \max\left\{ \|Au\| : u \in \mathcal{X} \text{ and } \|u\| = 1 \right\}.$$

One can define a family of *induced p-norms* in a similar way. However, we will not require the induced p-norms in this thesis. The Schatten p-norms are invariant under the action of any unitary operator. In other words, for any choice of appropriate unitary operators U and V,

$$\|A\|_p = \|UAV^*\|_p .$$

The Schatten p-norms are *submultiplicative*:

$$||AB||_p \leq ||A||_p \cdot ||B||_p$$

for any $p \in [1, \infty]$.

The following three norms will be useful in the thesis: the trace norm, the Frobenius norm, and the infinity norm. These norms have several nice properties. For instance, the operator norm satisfies

$$\|A^*A\| = \|AA^*\| = \|A\|^2$$

for every $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$. The Frobenius norm satisfies

$$\|A\|_2 = \sqrt{\mathrm{Tr}(A^*A)} = \sqrt{\langle A, A \rangle} = \sqrt{\langle \mathrm{vec}(A), \mathrm{vec}(A) \rangle} = \|\mathrm{vec}(A)\| .$$

The Frobenius and trace norms also satisfy the following inequalities for operators A and B for which the matrix product makes sense:

$$\|AB\|_2 \leq \|A\|_\infty \cdot \|B\|_2 \qquad \text{and} \qquad \|AB\|_1 \leq \|A\|_\infty \cdot \|B\|_1 .$$

The above inequality holds in general for any Schatten p-norm.

2.1.3 Linear super-operators

A *linear super-operator* (or simply a *super-operator*) is a linear map of the form

$$\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$$

where $\mathcal{X}$ and $\mathcal{Y}$ are complex Euclidean spaces. In other words, these action of these maps transforms an operator acting on $\mathcal{X}$ to an operator acting on $\mathcal{Y}$. The set of all such maps form a vector space. Such maps are always denoted by Φ, Ψ, and Ξ in this thesis. A super-operator can be represented in several ways. Before we proceed to do that, we define the *partial trace* of an operator. For two complex Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, the partial

trace is the unique linear map

$$\mathrm{Tr}_{\mathcal{Y}} : \mathrm{L}\left(\mathcal{X} \otimes \mathcal{Y}\right) \rightarrow \mathcal{X}$$

that satisfies

$$\mathrm{Tr}_{\mathcal{Y}}(A \otimes B) = \mathrm{Tr}(B)A$$

for all $A \in \mathrm{L}\left(\mathcal{X}\right)$ and $B \in \mathrm{L}\left(\mathcal{Y}\right)$. Similarly, one can define $\mathrm{Tr}_{\mathcal{X}} : \mathrm{L}\left(\mathcal{X} \otimes \mathcal{Y}\right) \rightarrow \mathcal{Y}$ to be the unique linear map that satisfies $\mathrm{Tr}_{\mathcal{X}}(A \otimes B) = \mathrm{Tr}(A)B$ for all $A \in \mathrm{L}\left(\mathcal{X}\right)$ and $B \in \mathrm{L}\left(\mathcal{Y}\right)$. For any operator $A \in \mathrm{L}\left(\mathcal{X} \otimes \mathcal{Y}\right)$, the partial trace with respect to $\mathcal{X}$ and $\mathcal{Y}$ has the following *operator sum* representation:

$$\mathrm{Tr}_{\mathcal{X}}(A) = \sum_{i=1}^{N} (e_i^* \otimes \mathbb{I}_{\mathcal{Y}}) A (e_i \otimes \mathbb{I}_{\mathcal{Y}}) \qquad \text{and} \qquad \mathrm{Tr}_{\mathcal{Y}}(A) = \sum_{i=1}^{M} (I_{\mathcal{X}} \otimes f_i^*) A (I_{\mathcal{X}} \otimes f_i).$$

By definition, the partial trace is independent of the choice of the basis used. Having defined partial trace, we now proceed to discuss three different super-operator representations and the relationships among them.

Kraus representation: The Kraus representation of a linear super-operator

$$\Phi : \mathrm{L}\left(\mathcal{X}\right) \rightarrow \mathrm{L}\left(\mathcal{Y}\right)$$

is given by

$$\Phi(X) = \sum_{i=1}^{K} A_i X B_i^*$$

for every choice of $X \in \mathrm{L}\left(\mathcal{X}\right)$ and for some operators $\left\{A_i \in \mathrm{L}\left(\mathcal{X}, \mathcal{Y}\right) : i \in [K]\right\}$ and $\left\{B_i \in \mathrm{L}\left(\mathcal{X}, \mathcal{Y}\right) : i \in [K]\right\}$. The operators A_i's and B_i's are referred to as *Kraus operators*. The Kraus representation exists for all super-operators. However, a super-operator Φ can have multiple Kraus representations. Note that it suffices to use NM operators to represent a super-operator in Kraus form. Moreover, there exist super-operators that require at least NM Kraus operators to describe them in the above form.

Stinespring representation: The Stinespring representation of a linear super-operator of the form $\Phi : \mathrm{L}\left(\mathcal{X}\right) \rightarrow \mathrm{L}\left(\mathcal{Y}\right)$ is given by

$$\Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXB^*)$$

for every choice of $X \in \mathrm{L}\left(\mathcal{X}\right)$ and for some operators $A, B \in \mathrm{L}\left(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z}\right)$, where $\mathcal{Z}$ is some vector space. The Stinespring representation of a super-operator always exists

but is in general not unique. Moreover, it is sufficient to take $\mathcal{Z} = \mathbb{C}^{NM}$ to describe a super-operator in this representation.

Choi-Jamiołkowski representation: The Choi-Jamiołkowski representation[1] of a linear super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is given by

$$J(\Phi) = \sum_{i,j=1}^{N} \Phi(e_i e_j^*) \otimes e_i e_j^* \in \mathrm{L}(\mathcal{Y} \otimes \mathcal{X}). \tag{2.1}$$

Alternately, one can write

$$J(\Phi) = \left(\Phi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{X})}\right)(\mathrm{vec}(\mathbb{I}_{\mathcal{X}})\mathrm{vec}(\mathbb{I}_{\mathcal{X}})^*)$$

where $\mathbb{I}_{\mathrm{L}(\mathcal{X})} : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{X})$ is the *identity* super-operator that maps any linear operator $X \in \mathrm{L}(\mathcal{X})$ to itself. The Choi-Jamiołkowski representation is basis dependent and is always defined with respect to the standard basis of $\mathcal{X}$. Unlike Kraus and Stinespring representations, Choi-Jamiołkowski representation is unique and one can recover the action of Φ by the following relationship:

$$\Phi(X) = \mathrm{Tr}_{\mathcal{X}}\left(J(\Phi)\left(\mathbb{I}_{\mathcal{Y}} \otimes X^{\top}\right)\right). \tag{2.2}$$

The Kraus, Stinespring, and Choi-Jamiołkowski representations are related to each other in the following way. For a super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$, let

$$\Phi(X) = \sum_{i=1}^{K} A_i X B_i^*$$

for some operators $\left\{A_i \in \mathrm{L}(\mathcal{X}, \mathcal{Y}) : i \in [K]\right\}$ and $\left\{B_i \in \mathrm{L}(\mathcal{X}, \mathcal{Y}) : i \in [K]\right\}$. Then the Stinespring representation can be defined as

$$\Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXB^*)$$

[1]Historically speaking, the definition of Choi-Jamiołkowski representation given above is due to Choi [Cho75]. Jamiołkowski [Jam72] came up with a slightly different representation, which for a super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is given by

$$J'(\Phi) = \sum_{i,j=1}^{N} e_j e_i^* \otimes \Phi(e_i e_j^*) \in \mathrm{L}(\mathcal{X} \otimes \mathcal{Y}).$$

However, we will refer to (2.1) as the Choi-Jamiołkowski representation due to the apparent similarity between the two representations with respect to definitions and properties.

where $\mathcal{Z} = \mathbb{C}^K$, and

$$A = \sum_{i=1}^{K} A_i \otimes g_i \qquad \text{and} \qquad B = \sum_{i=1}^{K} B_i \otimes g_i$$

for $\{g_i : i \in [K]\}$, the standard bases of $\mathcal{Z}$. The Choi-Jamiołkowski representation in terms of Kraus operators is

$$J(\Phi) = \sum_{i=1}^{K} \mathrm{vec}(A_i)\mathrm{vec}(B_i)^*.$$

Finally, for every super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$, there exists a *unique* super-operator $\Phi^* : \mathrm{L}(\mathcal{Y}) \to \mathrm{L}(\mathcal{X})$ that satisfies

$$\langle Y, \Phi(X) \rangle = \langle \Phi^*(Y), X \rangle$$

for all $X \in \mathrm{L}(\mathcal{X})$ and $Y \in \mathrm{L}(\mathcal{Y})$. The map Φ^* is called the *adjoint* super-operator of Φ.

Important sets of super-operators

We next proceed to briefly discuss some interesting classes of super-operators that are pertinent to this thesis.

1. A super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is *positive* if for every choice of $P \in \mathrm{Pos}(\mathcal{X})$, it holds that $\Phi(P) \in \mathrm{Pos}(\mathcal{Y})$.
2. A super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is *completely positive* if for every choice of vector space $\mathcal{Z}$ and $P \in \mathrm{Pos}(\mathcal{X} \otimes \mathcal{Z})$, it holds that $\left(\Phi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{Z})}\right)(P) \in \mathrm{Pos}(\mathcal{Y} \otimes \mathcal{Z})$.
3. A super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is *trace preserving* if for every choice of $X \in \mathrm{L}(\mathcal{X})$, it holds that $\mathrm{Tr}(\Phi(X)) = \mathrm{Tr}(X)$.
4. A super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is *Hermiticity preserving* if for every choice of $X \in \mathrm{Herm}(\mathcal{X})$, it holds that $\Phi(X) \in \mathrm{Herm}(\mathcal{Y})$.

Of the four different class of super-operators defined above, completely positive and trace preserving super-operators are relevant from quantum information viewpoint. A completely positive super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ has the following Kraus and Stinespring representations:

$$\Phi(X) = \sum_{i=1}^{K} A_i X A_i^* \qquad \text{and} \qquad \Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXA^*).$$

Consequently, the Choi-Jamiołkowski representation $J(\Phi) \in \mathrm{Pos}\left(\mathcal{Y} \otimes \mathcal{X}\right)$. This follows easily from the fact that

$$J(\Phi) = \sum_{i=1}^{K} \mathrm{vec}(A_i)\mathrm{vec}(A_i)^*.$$

In fact, the converse also holds for these representations. For instance, if $J(\Phi)$ is positive semidefinite, then the corresponding super-operator is completely positive.

A trace preserving super-operator $\Phi : \mathrm{L}\left(\mathcal{X}\right) \to \mathrm{L}\left(\mathcal{Y}\right)$ with Kraus and Stinespring representations

$$\Phi(X) = \sum_{i=1}^{K} A_i X B_i^* \qquad \text{and} \qquad \Phi(X) = \mathrm{Tr}_{\mathcal{Z}}(AXB^*)$$

necessarily satisfy

$$\sum_{i=1}^{K} A_i^* B_i = \mathbb{I}_{\mathcal{X}} \qquad \text{and} \qquad A^*B = \mathbb{I}_{\mathcal{X}}.$$

Moreover, the Choi-Jamiołkowski representation satisfies the relationship $\mathrm{Tr}_{\mathcal{Y}}(J(\Phi)) = \mathbb{I}_{\mathcal{X}}$. The converse also holds for these representations.

Finally, a Hermiticity preserving super-operator have the following alternate characterizations:

$$J(\Phi) \in \mathrm{Herm}\left(\mathcal{Y} \otimes \mathcal{X}\right) \qquad \text{and} \qquad \Phi(X) = \Phi_0(X) - \Phi_1(X)$$

for two completely positive super-operators $\Phi_0, \Phi_1 : \mathrm{L}\left(\mathcal{X}\right) \to \mathrm{L}\left(\mathcal{Y}\right)$. It also holds that $\big(\Phi(X)\big)^* = \Phi(X^*)$ for all $X \in \mathrm{L}\left(\mathcal{X}\right)$.

Norms of super-operators

We briefly discuss the following two norms on super-operator from the point of view of this thesis. For more discussion, we refer to Chapter 18 of the lecture notes on quantum information [Wat08].

Super-operator trace norm: For any super-operator $\Phi : \mathrm{L}\left(\mathcal{X}\right) \to \mathrm{L}\left(\mathcal{Y}\right)$, the super-operator trace norm is defined as

$$\left\|\Phi\right\|_1 = \max\left\{ \left\|\Phi(X)\right\|_1 : X \in \mathrm{L}\left(\mathcal{X}\right), \left\|X\right\|_1 \leq 1 \right\}.$$

Given that the trace norm is a convex function, one can replace the maximization over linear operators by maximization over linear operators of the form uv^*, where $u, v \in \mathcal{X}$ are

unit vectors. The super-operator trace norm is an example of an induced super-operator norm for every $p \geq 1$ defined by replacing the subscript "1" by p in the equation above.

Diamond norm: The diamond norm of a super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is defined as

$$\|\Phi\|_\diamond = \left\|\Phi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{X})}\right\|_1.$$

The diamond norm has several nice properties. One such property states that the choice of identity super-operator $\mathbb{I}_{\mathrm{L}(\mathcal{X})}$ acting on $\mathrm{L}(\mathcal{X})$ is not a particular choice, and one can replace it by $\mathbb{I}_{\mathrm{L}(\mathcal{Z})}$ for any vector space $\mathcal{Z}$ such that $\dim(\mathcal{Z}) \geq \dim(\mathcal{X})$. In other words, it holds that

$$\|\Phi\|_\diamond = \left\|\Phi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{Z})}\right\|_1$$

for all $\mathcal{Z}$ such that $\dim(\mathcal{Z}) \geq \dim(\mathcal{X})$. This can be interpreted as diamond norm being stable with respect to every choice of vector space $\mathcal{Z}$ such that $\dim(\mathcal{Z}) \geq \dim(\mathcal{X})$. Another nice property of diamond norm is that for two super-operators $\Phi_1 : \mathrm{L}(\mathcal{X}_1) \to \mathrm{L}(\mathcal{Y}_1)$ and $\Phi_2 : \mathrm{L}(\mathcal{X}_2) \to \mathrm{L}(\mathcal{Y}_2)$,

$$\|\Phi_1 \otimes \Phi_2\|_\diamond = \|\Phi_1\|_\diamond \cdot \|\Phi_2\|_\diamond.$$

2.2 Basic quantum information

In this section, we summarize the quantum information preliminaries required to understand the thesis. Our objective is to introduce the basics of quantum information from a linear algebra viewpoint. Our treatment to this subject is focused towards the results discussed in this thesis.

2.2.1 Quantum states, operations, and measurements

Quantum states: Associated with every N-level closed physical system X is a complex Euclidean space $\mathcal{X} = \mathbb{C}^N$. We refer to them as *quantum registers*. For physical systems X, Y, and Z, the corresponding complex Euclidean spaces as $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$. Each distinct level of the physical system is denoted by the standard basis of $\mathcal{X}$, which in the context of quantum information will be represented by the set

$$\{\, |1\rangle, |2\rangle, \ldots, |N\rangle \,\}.$$

One identifies the set of standard bases of $\mathcal{X}$ with the above set by the relationship $|i\rangle = e_i$. Here $|\cdot\rangle$ is the Dirac's notation for unit vectors and the associated *dual* vector is denoted $\langle\cdot|$. A *quantum state* of such a system can be described by the set of density operators

acting on $\mathcal{X}$: the set of positive semidefinite operators with trace one. The quantum state of a physical system is said to be *pure*, if the associated density operator ρ has rank one. That is, $\rho = uu^*$ for some unit vector $u \in \mathcal{X}$. In the context of quantum information, we often write u in the Dirac's notation as

$$|\psi_u\rangle = \sum_{i=1}^{N} u[i] \, |i\rangle \, .$$

For notational convenience, we never use the subscript u, and instead use $|\psi\rangle, |\phi\rangle, |\varphi\rangle$ to represent *unit* vectors in $\mathcal{X}$ when representing pure quantum states. A quantum state is called *mixed* if it cannot be represented as a rank one density operator. It follows from the spectral decomposition of normal operators (Theorem 2) that any quantum state ρ can be viewed as a probability distribution over pure states. Note that this ensemble is in general not unique.

For two physical systems X and Y with associated complex Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, one can view the two physical system as one system (X, Y) with the associated complex Euclidean space $\mathcal{X} \otimes \mathcal{Y}$. For a quantum state $\rho_{\mathsf{XY}} \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ of the larger system, the quantum state of the subsystems X and Y (also called the *reduced states*) are respectively given by

$$\rho_{\mathsf{X}} = \mathrm{Tr}_{\mathcal{Y}}(\rho_{\mathsf{XY}}) \qquad \text{and} \qquad \rho_{\mathsf{Y}} = \mathrm{Tr}_{\mathcal{X}}(\rho_{\mathsf{XY}}).$$

The relationship $\rho_{\mathsf{XY}} = \rho_{\mathsf{X}} \otimes \rho_{\mathsf{Y}}$ almost never holds. A quantum state $\rho \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ is a *product state* if it can be expressed as $\rho \otimes \sigma$ for $\rho \in \mathrm{D}(\mathcal{X})$ and $\sigma \in \mathrm{D}(\mathcal{Y})$. A quantum state is called *separable* if it can be expressed as a convex combination of product states. In other words,

$$\rho = \sum_{i=1}^{K} p_i \rho_i \otimes \sigma_i \tag{2.3}$$

for some $\{\rho_i : i \in [K]\} \subset \mathrm{D}(\mathcal{X})$ and $\{\sigma_i : i \in [K]\} \subset \mathrm{D}(\mathcal{Y})$. We remark that this decomposition is in general not unique. The usage of term "separable" is not limited to density operators. More generally, a positive semidefinite operator $R \in \mathrm{Pos}(\mathcal{X} \otimes \mathcal{Y})$ is called a *separable operator* if it can be expressed in the form

$$R = \sum_{i=1}^{K} P_i \otimes Q_i$$

for some $\{P_i : i \in [K]\} \subset \mathrm{Pos}(\mathcal{X})$ and $\{\sigma_i : i \in [K]\} \subset \mathrm{Pos}(\mathcal{Y})$. A quantum state $\rho \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ is called *entangled* if it cannot be written in the form as described in (2.3). In different settings, entangled states are used to show non-local behaviors that are indescribable by the laws of classical physics. A few example of pure entangled states in

$\mathrm{D}(\mathbb{C}^2 \otimes \mathbb{C}^2)$ are

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}, \qquad \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \qquad \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \qquad \text{and} \qquad \frac{|01\rangle - |10\rangle}{\sqrt{2}}.$$

These states are often referred to as *maximally* entangled state on two qubits. The term "maximal" refers to the fact that under some measure of entanglement, no other state in same complex Euclidean space can exhibit more entanglement that the above state. Without getting into any details of entanglement measures, we can alternately say that a pure state in $\mathrm{D}(\mathcal{X} \otimes \mathcal{X})$ is maximally entangled if the reduced states of both the systems is $\mathbb{I}_{\mathcal{X}}/\dim(\mathcal{X})$. This only applies to bipartite pure states.

Quantum operations: An operation is said to be a *quantum operation* if, roughly speaking, it maps quantum states to quantum states. In more formal terms, a quantum operation is a super-operator of the form $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ that satisfies the following two conditions.

1. For every choices of vector space $\mathcal{Z}$ and $A \in \mathrm{Pos}(\mathcal{X} \otimes \mathcal{Z})$, it holds that

$$(\Phi \otimes \mathbb{I}_{\mathrm{L}(\mathcal{X})})(A) \in \mathrm{Pos}(\mathcal{Y} \otimes \mathcal{X}).$$

2. For every choice of $A \in \mathrm{L}(\mathcal{X})$, it holds that

$$\mathrm{Tr}(\Phi(A)) = \mathrm{Tr}(A).$$

In other words, a quantum operation is a completely positive and trace preserving map. Taken together, the two constraints imply that density operators are always mapped to density operators. Such maps are also called *admissible operation*. One such quantum operation is the map $X \mapsto UXU^*$ for some unitary $U \in \mathrm{U}(\mathcal{X})$. The unitary operators are sufficient to describe any quantum operation in the following sense. The action of a quantum operation $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ on any state $\rho \in \mathrm{D}(\mathcal{X})$ can be expressed as

$$\Phi(\rho) = \mathrm{Tr}_{\mathcal{Z}}\left(U(\rho \otimes |\psi\rangle\langle\psi|)U^*\right)$$

for some vector space $\mathcal{Z}$ of sufficiently large dimension, a unitary $U \in \mathrm{U}(\mathcal{X} \otimes \mathcal{Z})$, and a fixed quantum state $|\psi\rangle \in \mathcal{Z}$ independent of ρ. A few example of one qubit unitary operators that will be referred to in this thesis are

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -\iota \\ \iota & 0 \end{pmatrix}, \qquad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The operators X, Y, and Z are referred to as *Pauli operators* and they anti-commute with each other. Together with $\mathbb{I}_{\mathbb{C}^2}$, they form an orthogonal basis for the space of one qubit operators.

Quantum measurements: In order to obtain classical information from a quantum state, one can perform *measurement* on the system. There are two ways of formalizing measurement operators. The first formalism allows us to describe measurement statistics and the post-measurement states conditioned on outcomes. The second formalism allows us to describe the measurement statistics only and implicitly assumes that the state is destroyed after the measurement is applied. This thesis is not concerned with post-measurement quantum states and therefore the latter formalism is more suited in our context.

Formally, a *quantum measurement* on a physical system X consists of a set of positive semidefinite operators

$$\mathcal{M} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\} \subset \mathrm{Pos}\left(\mathcal{X}\right)$$

that satisfies

$$\sum_{i=1}^{m} \Lambda_i = \mathbb{I}_{\mathcal{X}}.$$

The outcome of the measurement $\mathcal{M}$ is the one of the labels in the set $[m]$. For any state $\rho \in \mathrm{D}\left(\mathcal{X}\right)$, the probability that the measurement operation described by $\mathcal{M}$ yields an outcome i is given by

$$\mathbb{P}[\mathcal{M}(\rho) = i] = \langle \Lambda_i, \rho \rangle .$$

For most parts of the thesis, we work with a special kind of measurement operation called *projective measurements*. Projective measurement can be identified with the set of measurement operators

$$\mathcal{P} = \{\Pi_1, \Pi_2, \ldots, \Pi_m\} \subset \mathrm{Pos}\left(\mathcal{X}\right),$$

where Π_i's are orthogonal projection and $\langle \Pi_i, \Pi_j \rangle = 0$ for every choice of i and j such that $i \neq j$. For the purpose of this thesis, we do not lose any generality by assuming that the measurement operators are projection operators. Indeed, by Naimark's theorem, an arbitrary measurement on a physical system can be simulated by a projective measurement over a larger system. More formally, for any measurement $\mathcal{M} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\} \subset \mathcal{X}$, there exists a vector space $\mathcal{Z}$, a unit vector $|\psi\rangle \in \mathcal{Z}$, and projective measurement described by operators $\mathcal{P} = \{\Pi_1, \Pi_2, \ldots, \Pi_m\} \subset \mathrm{Pos}\left(\mathcal{X} \otimes \mathcal{Z}\right)$ such that

$$\langle \Lambda_i, \rho \rangle = \langle \Pi_i, \rho \otimes |\psi\rangle \langle\psi| \rangle$$

for all $\rho \in \mathrm{D}\left(\mathcal{X}\right)$. In other words, the projective measurement operation $\mathcal{P}$ generate the same statistics as $\mathcal{M}$.

2.2.2 Relevant facts in quantum information theory

Purifications: Suppose the state of a physical system is $\rho \in \mathrm{D}(\mathcal{X})$. For a vector space $\mathcal{Y}$, we say that $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ is a *purification* of ρ if

$$\rho = \mathrm{Tr}_{\mathcal{Y}}(|\psi\rangle \langle\psi|).$$

Such a purification always exists for any choice of $\mathcal{Y}$ such that $\dim(\mathcal{Y}) \geq \mathrm{rank}(\rho)$, where $\mathrm{rank}(\rho)$ denotes the rank of ρ. This is an easy implication of spectral theorem. In fact, if $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ is a purification of ρ, then $\dim(\mathcal{Y}) \geq \mathrm{rank}(\rho)$. The purification of a density operator ρ is never unique because one can always apply a unitary on the system Y to obtain a different quantum state that purifies the density operator. However, up to a unitary transformation, the purification is always unique as summarized by the following theorem.

Theorem 5. *Let $\rho \in \mathrm{D}(\mathcal{X})$ and $\mathcal{Y}$ be a vector space such that $\dim(\mathcal{Y}) \geq \mathrm{rank}(\rho)$. Then for every two purifications $|\psi\rangle, |\phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ of ρ, there exists a unitary $U \in \mathrm{U}(\mathcal{Y})$ such that $|\psi\rangle = (\mathbb{I}_{\mathcal{X}} \otimes U)|\phi\rangle$.*

The theorem above is often referred to as *unitary equivalence of purification*. It can be used to show the following. If two density operators $\rho_1, \rho_2 \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ agree on the subsystem X, then there exists a quantum operation on subsystem Y that maps ρ_1 to ρ_2, a fact used in Chapter 5. Finally, we note that the notion of purification extends to positive semidefinite operators in the most natural way. The square of the Euclidean norm of the vector that purifies the positive semidefinite operator is equal to trace of the operator.

Distance measures for quantum states: Given two states $\rho_0 \in \mathrm{D}(\mathcal{X})$ and $\rho_1 \in \mathrm{D}(\mathcal{X})$, the *trace distance* between them is defined as

$$\|\rho_0 - \rho_1\|_1 = \sum_{i=1}^{\dim(\mathcal{X})} |\lambda_i(\rho_0 - \rho_1)|$$

The trace distance can be seen as a generalization of statistical distance (or variational distance) between two probability distributions to quantum states and quantifies how close or far two quantum states are from each other. The quantity lies in the interval $[0, 2]$, where it is 0 and 2 if and only if

$$\rho_0 = \rho_1 \qquad \text{and} \qquad \langle \rho_0, \rho_1 \rangle = 0$$

respectively. The trace distance is closely related to the optimum probability of distinguishing two quantum states given uniformly at random. In other words, given two states

ρ_0 and ρ_1 uniformly at random, the optimum success probability of guessing the correct state is

$$\frac{1}{2} + \frac{1}{4} \left\| \rho_0 - \rho_1 \right\|_1 .$$

Another quantity that quantifies the similarity or difference between two quantum states is the notion of *fidelity*. Given two states $\rho, \sigma \in \mathrm{D}(\mathcal{X})$, the fidelity between ρ_0 and ρ_1 is

$$\mathrm{F}(\rho, \sigma) = \left\| \sqrt{\rho}\sqrt{\sigma} \right\|_1 .$$

The fidelity between two pure states $|\psi\rangle$ and $|\phi\rangle$ is $|\langle\phi|\psi\rangle|$. The fidelity function lies in the interval $[0, 1]$, where it is 0 and 1 if and only if

$$\langle \rho, \sigma \rangle = 0 \qquad \text{and} \qquad \rho = \sigma$$

respectively. The fidelity function has some nice properties. One such property is that it is multiplicative with respect to tensor products. That is, for $\rho_0, \sigma_0 \in \mathrm{D}(\mathcal{X})$ and $\rho_1, \sigma_1 \in \mathrm{D}(\mathcal{Y})$, it holds that

$$\mathrm{F}(\rho_0 \otimes \rho_1, \sigma_0 \otimes \sigma_1) = \mathrm{F}(\rho_0, \sigma_0)\, \mathrm{F}(\rho_1, \sigma_1).$$

The fidelity of two density operators can be characterized in terms of their purifications by the following theorem known as Uhlmann's theorem.

Theorem 6. *Let $\rho, \sigma \in \mathrm{D}(\mathcal{X})$ and $\mathcal{Y}$ be a vector space such that the purifications of ρ and σ exist in $\mathcal{X} \otimes \mathcal{Y}$. For any purification $|\psi\rangle$ of ρ,*

$$\mathrm{F}(\rho, \sigma) = \max \left\{ |\langle \psi | \phi \rangle| : \phi \in \mathcal{X} \otimes \mathcal{Y} \text{ is a purification of } \sigma \right\}.$$

The two notions discussed above, fidelity and trace distance, are related to each other by the following inequalities, known as Fuchs-van de Graaf inequalities. The inequalities relate the trace distance and fidelity of two density operators $\rho, \sigma \in \mathrm{D}(\mathcal{X})$ as follows:

$$1 - \frac{1}{2} \left\| \rho_0 - \rho_1 \right\|_1 \leq \mathrm{F}(\rho, \sigma) \leq \sqrt{1 - \frac{1}{4} \left\| \rho_0 - \rho_1 \right\|_1^2}. \tag{2.4}$$

Distance measures for quantum operations: Given two quantum operations $\Phi_0, \Phi_1 : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$, one defines the *trace distance* between the quantum operations as

$$\left\| \Phi_0 - \Phi_1 \right\|_1 = \max \left\{ \left\| \Phi_0(\rho) - \Phi_1(\rho) \right\| :_{\rho \in \mathrm{D}(\mathcal{X})} \right\}.$$

It holds that for Hermiticity preserving super-operators the trace norm is attained by a density operator. Unlike the case of trace distance between states, the trace distance between the quantum operations does not completely characterize the optimum distin-

guishing procedure. To illustrate this fact, consider the following scenario between Alice and Bob. Alice has two quantum operations $\Phi_0, \Phi_1 : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ and she applies one of the operations uniformly at random on a quantum state provided by Bob. The goal of the Bob is to prepare a state that determines the operation applied by Alice. One strategy of Bob is to prepare a state $\rho \in \mathrm{D}(\mathcal{X})$ and send it to Alice. It is clear that Bob's optimum success probability is

$$\frac{1}{2} + \frac{1}{4} \max \{ \| \Phi_0(\rho) - \Phi_1(\rho) \|_1 : \rho \in \mathrm{D}(\mathcal{X}) \} = \frac{1}{2} + \frac{1}{4} \| \Phi_0 - \Phi_1 \|_1 .$$

However, a better approach for Bob is to prepare a state $\sigma \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Z})$ for some choice of vector space $\mathcal{Z}$ and send the first subsystem to Alice. In this case, Bob's optimum success probability is

$$\frac{1}{2} + \frac{1}{4} \max \left\{ \left\| \left(\Phi_0 \otimes \mathbb{I}_{\mathrm{L}(\mathcal{Z})} \right)(\sigma) - \left(\Phi_1 \otimes \mathbb{I}_{\mathrm{L}(\mathcal{Z})} \right)(\sigma) \right\|_1 : \sigma \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Z}) \right\} \leq \frac{1}{2} + \frac{1}{4} \| \Phi_0 - \Phi_1 \|_\diamond .$$

Of course, the equality can be achieved for an appropriate choice of $\mathcal{Z}$ and $\sigma \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Z})$.

2.3 Summary of notations

This section briefly summarizes the notations defined in this chapter. We begin with linear algebra basics first.

- The complex Euclidean spaces are denoted by script uppercase letters $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$. The vectors are denoted by lowercase Roman letters u, v, and w. The dimension of a vector space $\mathcal{X}$ is denoted $\dim(\mathcal{X})$. A real vector space of dimension N is denoted $\mathbb{R}^N$.

- The vector space $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ denotes the space of operators of the form $A : \mathcal{X} \to \mathcal{Y}$. If $\mathcal{X} = \mathcal{Y}$, the operator space is denoted $\mathrm{L}(\mathcal{X})$. For an operator A, the operators $A^*, \overline{A}$, and $A^\top$ are the adjoint, entry-wise conjugate, and transpose of A.

- The set of Hermitian operators are denoted $\mathrm{Herm}(\mathcal{X})$. The set of symmetric operators are denoted $\mathrm{Herm}\left(\mathbb{R}^N\right)$ if $\mathbb{R}^N$ is the underlying space on which the symmetric operators act.

- The set of positive semidefinite operators are denoted $\mathrm{Pos}(\mathcal{X})$. The set of symmetric positive semidefinite operators are denoted $\mathrm{Pos}\left(\mathbb{R}^N\right)$ if $\mathbb{R}^N$ is the underlying space on which the positive semidefinite operators act. We use the standard notation $A \succeq 0$ to denote that A is positive semidefinite.

- The set of positive definite operators are denoted $\mathrm{Pd}(\mathcal{X})$. The set of symmetric positive definite operators are denoted $\mathrm{Pd}\left(\mathbb{R}^N\right)$ if $\mathbb{R}^N$ is the underlying space on which the positive definite operators act. We use the standard notation $A \succ 0$ to denote that A is positive semidefinite.
- The set of density operators are denoted $\mathrm{D}(\mathcal{X})$. The lowercase Greek symbols ρ, σ, and ξ are used to denote density operators.
- The set of linear isometries are denoted $\mathrm{U}(\mathcal{X}, \mathcal{Y})$ and the set of unitary operators are denoted $\mathrm{U}(\mathcal{X})$.
- For a super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$, the Choi-Jamiołkowski representation is denoted $J(\Phi)$.
- The Euclidean norm and 1-norm of a vector $u \in \mathcal{X}$ are denoted $\|u\|$ and $\|u\|_1$ respectively.
- The operator, Frobenius, and trace norms of an operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ are denoted $\|A\|_\infty$, $\|A\|_2$, and $\|A\|_1$ respectively.
- The super-operator trace norm and diamond norm of a super-operator $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ are denoted $\|\Phi\|_1$ and $\|\Phi\|_\diamond$ respectively.

The notations used in this thesis when discussing quantum information is as follows.

- Any physical system with associated complex Euclidean space $\mathcal{X}$ is identified by a quantum register X. The pure quantum states are denoted using Dirac's notation such as $|\psi\rangle$, $|\psi\rangle$, and $|\varphi\rangle$.
- The thesis focuses on projective measurement operators, which are usually denoted by Λ, Π, and Δ.
- The fidelity of two quantum states ρ and σ is denoted $\mathrm{F}(\rho, \sigma)$.

Chapter 3

Semidefinite Programming

The purpose of this chapter is to introduce semidefinite programming (SDP), a widely used optimization method with enormous applications in various disciplines in mathematics and theoretical computer science such as approximation algorithms, control theory, information theory, and quantum information. Semidefinite programs are either a maximization or a minimization of a linear function over the intersection of a collection of finitely many affine constraints and the set of positive semidefinite operators. From the viewpoint of formulation of the optimization problem, they are a class of conic programs that generalizes linear programming, quadratic programming, and second-order cone programming. From the algorithmic perspective, they generalize linear programming and second-order cone programming.

A major part of this thesis uses semidefinite programming techniques. The wide applications of semidefinite programs comes from the fact that they can be used as an algorithmic and an analytical tool. The results presented in Chapters 5, 6, and 7 can be viewed as applications of semidefinite programs. While in Chapter 5, we exploit a class of semidefinite programs as an algorithmic tool, the next two Chapters focus on using semidefinite programs analytically to prove certain *parallel repetition theorems*. This chapter itself is divides in three sections.

- In Section 3.1, we introduce semidefinite programs and briefly discuss two syntactically different yet equivalent forms of semidefinite programs: *standard form* and the *super-operator form*.
- In Section 3.2, we briefly discuss semidefinite programming duality theory and sufficient conditions for strong duality to hold. The semidefinite programs that are predominantly used in this thesis do satisfy the strong duality conditions (Theorems 8 and 9).

- In Section 3.3, we briefly summarize how semidefinite programs have been applied in this thesis.

For a comprehensive treatment on the subject, we refer to excellent surveys [Lov03, LR05, VB96], texts [Tun10, VB04], and the references therein.

3.1 Equivalent forms of semidefinite programs

The two forms that we discuss in this section are known as the standard form and the super-operator form. Let $\mathcal{X} = \mathbb{C}^N$ be a complex Euclidean space, $A \in \mathrm{Herm}(\mathcal{X})$, $b \in \mathbb{R}^M$, and $\mathcal{F} : \mathrm{L}(\mathcal{X}) \to \mathbb{C}^M$ be a linear function whose action on any operator in $\mathrm{Herm}(\mathcal{X})$ is a vector in $\mathbb{R}^M$. A semidefinite program in *standard form* associates the following two optimization problems to the triple $(\mathcal{F}, A, b)$:

Primal problem		Dual problem	
supremum:	$\langle A, X \rangle$	infimum:	$\langle b, y \rangle$
subject to:	$\mathcal{F}(X) = b,$	subject to:	$\mathcal{F}^*(y) = A + Y,$
	$X \in \mathrm{Pos}(\mathcal{X}).$		$Y \in \mathrm{Pos}(\mathcal{X}).$

The use of *supremum* and *infimum* is justified because there are instances of semidefinite programs where a finite optimum value exists but is not attained. The functions $\langle A, X \rangle$ and $\langle b, y \rangle$ are called the *primal objective function* and the *dual objective function*. The map $\mathcal{F}$ can be described by a collection of operators $\{F_1, F_2, \ldots, F_M\} \subset \mathrm{Herm}(\mathcal{X})$ such that

$$\mathcal{F}(X)[i] = \langle F_i, X \rangle .$$

The map $\mathcal{F}^*$ is the unique linear map that satisfies $\langle y, \mathcal{F}(X) \rangle = \langle \mathcal{F}^*(y), X \rangle$ for all $X \in \mathrm{L}(\mathcal{X})$ and $y \in \mathbb{C}^M$. It is an easy observation that

$$\mathcal{F}^*(y) = \sum_{i=1}^{M} y[i] F_i.$$

We can equivalently write the two optimization problems as follows:

Primal problem		Dual problem	
supremum:	$\langle A, X \rangle$	infimum:	$\langle b, y \rangle$
subject to:	$\forall i : \langle F_i, X \rangle = b_i,$	subject to:	$\sum_i y[i] F_i - A \in \mathrm{Pos}(\mathcal{X}).$
	$X \in \mathrm{Pos}(\mathcal{X}).$		

While writing semidefinite programs in the standard form, we may often omit the underlying space on which the operators act and use the notation $X \succeq 0$ instead of $X \in \mathrm{Pos}(\mathcal{X})$ for convenience. This will not create any ambiguity and the underlying space will always be understood from the context.

In literature the *standard form* is usually written in terms of symmetric operators instead of Hermitian operators. In other words, the operators $A, F_1, F_2, \ldots, F_M$ are symmetric operators. It is not hard to see that these two forms are equivalent. Any semidefinite program in the above form can be converted into a semidefinite program where the operators specifying the program are symmetric.

In various applications, the standard form is sufficient to concisely describe the program. However the semidefinite programming problem that arises in Chapter 5 is better described in a different form, called the super-operator form. Let $\mathcal{X}$ and $\mathcal{Y}$ be two vector spaces, $A \in \mathrm{Herm}(\mathcal{X})$, $B \in \mathrm{Herm}(\mathcal{Y})$, and $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ be a Hermiticity preserving super-operator. A semidefinite program in *super-operator form* associates the following two optimization problems to (Φ, A, B):

Primal problem		Dual problem	
supremum:	$\langle A, X \rangle$	infimum:	$\langle B, Y \rangle$
subject to:	$\Phi(X) \preceq B$,	subject to:	$\Phi^*(Y) \succeq A$,
	$X \in \mathrm{Pos}(\mathcal{X})$.		$Y \in \mathrm{Pos}(\mathcal{Y})$.

The adjoint super-operator $\Phi^* : \mathrm{L}(\mathcal{Y}) \to \mathrm{L}(\mathcal{X})$ is the unique Hermiticity preserving linear map that satisfies

$$\langle Y, \Phi(X) \rangle = \langle \Phi^*(Y), X \rangle$$

for all $X \in \mathrm{L}(\mathcal{X})$ and $Y \in \mathrm{L}(\mathcal{Y})$.

It is sometimes desirable to work with semidefinite programs in super-operator form. The two forms are equivalent to each other in the following sense. Any semidefinite program in standard form can be converted into a semidefinite program in super-operator form and vice-versa.

3.2 Semidefinite programming duality

The duality theory for semidefinite programs refers to the relation between the associated primal and dual problems. A simple observation leads us to the following theorem, referred to as the *weak duality theorem*.

Theorem 7. *The following hold.*

1. *Let the triple* $(\mathcal{F}, A, b)$ *describe a semidefinite program in the standard form. Then for any feasible solutions* X *and* (y, Y) *of the primal and the dual problems, respectively,*

$$\langle b, y \rangle \geq \langle A, X \rangle .$$

2. *Let the triple* (Φ, A, B) *describe semidefinite program in the super-operator form. Then for any feasible solutions* X *and* Y *of the primal and the dual problems, respectively,*

$$\langle B, Y \rangle \geq \langle A, X \rangle .$$

One follows the following convention in case the set of primal feasible solutions is empty or the set of dual feasible solutions is empty. If the set of primal feasible solutions is empty, then the optimum of the primal problem is assumed to be $-\infty$. Likewise if the set of dual feasible solutions is empty, then the optimum of the dual problem is assumed to be ∞.

One of the fundamental facts about linear programs is that both primal and dual problems attain their respective optimum values assuming that they are feasible. Moreover, the optimum values of primal and dual problems coincide with each other. This property is called *strong duality*. In contrast, semidefinite programs do not always satisfy strong duality. One can easily come up with examples of semidefinite programs where the primal or the dual problem do not attain their respective optimum values or there is a positive *duality gap*.

Although the presence of a positive duality gap in semidefinite programming is unavoidable, there exists a strong duality theorem for semidefinite programs that satisfy come conditions. The semidefinite programs that we often encounter in various applications generally satisfy the strong duality conditions.

For a semidefinite program in the standard form, let α and β denote the optimum values for primal and dual problems, respectively. One has the following strong duality theorem.

Theorem 8. *For a vector space* $\mathcal{X} = \mathbb{C}^N$*, let* $A \in \mathrm{Herm}(\mathcal{X})$*,* $b \in \mathbb{R}^M$*, and* $\mathcal{F} : \mathrm{L}(\mathcal{X}) \to \mathbb{C}^M$ *specify a semidefinite program in standard form. Then the following hold.*

1. *Let* α *be finite and suppose there exists an operator* $X' \in \mathrm{Pd}(\mathcal{X})$ *such that* $\mathcal{F}(X') = b$*. Then* $\alpha = \beta$ *and there exists* $(y, Y) \in \mathbb{R}^M \oplus \mathrm{Pos}(\mathcal{X})$ *such that*

$$\mathcal{F}^*(y) = A + Y \qquad \text{and} \qquad \langle b, y \rangle = \beta.$$

2. *Let* β *be finite and suppose there exists* $(y', Y') \in \mathbb{R}^M \oplus \mathrm{Pd}(\mathcal{X})$ *such that* $\mathcal{F}^*(y') = A + Y'$*. Then* $\alpha = \beta$ *and there exists* $X \in \mathrm{Pos}(\mathcal{X})$ *such that*

$$\mathcal{F}(X) = b \qquad \text{and} \qquad \langle A, X \rangle = \alpha.$$

The conditions that there must exist a primal feasible solution X' such that $X' \in \mathrm{Pd}(\mathcal{X})$ or there must exist a dual feasible solution (y', Y') such that $Y' \in \mathrm{Pd}(\mathcal{X})$ are termed as *Slater conditions*, and X' and (y', Y') are called *Slater points*. Item 1 of the theorem states that the existence of a Slater point for the primal problem implies that strong duality holds and that a dual feasible solution attains the optimum dual value. Item 2 of the theorem states that the existence of a Slater point for the dual problem implies that strong duality holds and that a primal feasible solution attains the primal optimum value. The existence of Slater points for both primal and dual problems guarantees that both problems attain the same optimum value.

A similar theorem holds for semidefinite programs in super-operator form as stated below [Wat08]. Let α and β be the optimum values for primal and dual problems, respectively.

Theorem 9. *For vector spaces $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$, let $A \in \mathrm{Herm}(\mathcal{X})$, $B \in \mathrm{Herm}(\mathcal{Y})$, and $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ specify a semidefinite program in super-operator form. Then the following hold.*

1. *Let α be finite and suppose there exists an operator $X' \in \mathrm{Pd}(\mathcal{X})$ such that $\Phi(X') \prec B$. Then $\alpha = \beta$ and there exists $Y \in \mathrm{Pos}(\mathcal{Y})$ such that*

$$\Phi^*(Y) \succeq A \qquad \text{and} \qquad \langle B, Y \rangle = \beta.$$

2. *Let β be finite and suppose there exists $Y' \in \mathrm{Pd}(\mathcal{Y})$ such that $\Phi^*(Y') \succ A$. Then $\alpha = \beta$ and there exists $X \in \mathrm{Pos}(\mathcal{X})$ such that*

$$\Phi(X) \succeq B \qquad \text{and} \qquad \langle A, X \rangle = \alpha.$$

Item 1 of the theorem states that the existence of a Slater point for the primal problem implies that strong duality holds and that a dual feasible solution attains the dual optimum value. Item 2 of the theorem states that the existence of a Slater point for the dual problem implies that strong duality holds and that a primal feasible solution attains the primal optimum value. The existence of Slater points for both primal and dual problem guarantees that both problems attain the same optimum value.

3.3 Semidefinite programs in this thesis

This section intends to briefly summarize how semidefinite programs are applied in this thesis. From an algorithmic perspective, we show that a certain class of semidefinite programs can be solved by an efficient parallel algorithm, which we then use to prove

a better complexity-theoretic upper bound on the model of quantum interactive proof systems. From an analytical perspective, we use semidefinite programs to establish *parallel repetition theorems* for certain classes of two-prover one-round quantum games.

3.3.1 Algorithmic application of semidefinite programs

Semidefinite programs have been widely used as an algorithmic means to efficiently solve optimization problems. The efficiency of the algorithm is guaranteed by a generic algorithm known as the *ellipsoid method*. From a theoretical viewpoint, the ellipsoid method is an extremely powerful method and provided that the semidefinite program under consideration satisfies some nice conditions, it guarantees a provable polynomial-time algorithm for finding a feasible solution with objective value ϵ-close to the optimum value.

The *semidefinite programming approximation problem* is the following computational problem as described in Watrous [Wat08].

- *Input.* A semidefinite program (Φ, A, B) over $\mathcal{X} = \mathbb{C}^N$ and $\mathcal{Y} = \mathbb{C}^M$, an accuracy parameter $\epsilon > 0$, and positive rational numbers r and R.
- *Output.* A number γ such that $|\gamma - \alpha| < \epsilon$, where α is the optimum value of the primal problem of the semidefinite program.

The ellipsoid method requires the following two conditions to be satisfied by the semidefinite program (Φ, A, B) [GLS93]. The conditions essentially define the parameters r and R mentioned in the computational problem above.

1. The set of primal feasible solutions is contained in a ball of radius R. In other words, for every primal feasible X, $\|X\|_2 \leq R$.
2. The set of primal feasible solutions contains a ball of radius r. In other words, there exists a primal feasible solution X such that $X + C$ is also primal feasible for every choice of C such that $\|C\|_2 \leq r$.

The running time of the ellipsoid method is a polynomial in the input parameters N, M, polylogarithmic in the input parameters $1/r, R, 1/\epsilon$, and polynomial in the maximum bit-length of the matrix entries of $(J(\Phi), A, B)$. Here $J(\Phi)$ is the Choi-Jamiołkowski representation of super-operator Φ. One can recover the action of Φ on any operator X from $J(\Phi)$ by using (2.2). It should be noted that the choice of using the primal problem to run the ellipsoid method is not necessary. If the dual problem satisfies items 1 and 2 above, then one can run the ellipsoid method on the dual problem as well. If strong duality

holds, then it will give an approximation for the optimum value of the other problem as well.

While ellipsoid method is a very powerful algorithmic technique and is provably a polynomial-time algorithm on Turing machine, there are faster algorithms for semidefinite programs, the interior-point methods, that are used in practice. At present the polynomial running time on interior-point methods is not established on a Turing machine. However, this thesis is concerned with a recent approach for solving a class of semidefinite programs based on the *matrix multiplicative weights update method*. The ordinary multiplicative weights update method is a framework for algorithmic design having its origins in various fields including combinatorial optimization [PST95], computational geometry [Cha00], game theory [BvN50, Rob51], and learning theory [FS99]. Its matrix version, as discussed in the PhD thesis of Kale [Kal07], is an iterative procedure that can be used to efficiently approximate optimum values of certain semidefinite programs [AK07, WK06]. However, the running time of the algorithm depends polynomially on $1/\epsilon$, where ϵ is the accuracy parameter.

A very rough idea of the algorithm presented in Arora and Kale [AK07] is a primal-dual approach based on matrix multiplicative weights update method and proceeds as follows. The algorithm guesses an objective value γ for the optimum value, and proceeds to construct either a primal feasible solution or a dual feasible solution with the desired objective value. To do this, the algorithm guesses a candidate solution for the primal problem and calls a subroutine. The function of the subroutine is to returns a certificate that shows that the candidate solution is either infeasible or does not attain the desired objective value or both. The specifics of the subroutine depends on the semidefinite program in hand and is not important right now. If the subroutine returns a certificate, then the algorithm updates the candidate solution using the certificate to generate a new candidate solution. If the subroutine does not return a certificate, then it can be showed that the candidate solution is close to being primal feasible and the objective value is approximately γ. Finally, if the subroutine returns a certificate for a sufficient number of iterations, then the certificates generated in all iterations can be used to generate a dual feasible solution with objective value close to γ.

An important aspect of the matrix multiplicative weights update method from the viewpoint of this thesis and the papers [JUW09, JW09] is its *parallelizability* for certain class of semidefinite programs. For the class of semidefinite programs of our interest, the method runs for a very few number of iterations, each iteration consisting of matrix computations that can be parallelized, and gives a certificate of primal or dual feasibility with a certain objective value. Although we haven't made the notion of "few iterations" precise, we emphasize that the fact that the method runs for few iterations for the semidefinite program of our interest cannot be generalized to any semidefinite program under some widely believed complexity-theoretic assumption.

3.3.2 Analytical application of semidefinite programs

Semidefinite programs have been used analytically to prove important theorems in various disciplines such as combinatorial optimization, discrepancy theory, metric embeddings, and quantum information (see the references [HLW06, GW07, Lov79, Lov00] for some applications on these topics). For more applications, we refer to the surveys and texts mentioned at the beginning of the chapter. In this thesis, we use semidefinite programs to establish parallel repetition theorems for certain classes of two-prover one-round games.

Imagine a protocol between two co-operating quantum players and a referee, where the referee picks a question for each player, sends the questions, receives an answer from each player, and decides to accept or reject based on the questions and answers. The measure of how well the players perform is determined by the maximum success probability achieved by them, which is called the *quantum value* of the game. The parallel repetition of such games asks the following question. What is the rate of decrease of success probability of players if the referee plays multiple instances of the protocol simultaneously and decides to accept if and only if all instances are accepted? Ideally if the original game G has quantum value p, then the n-fold repetition of the game should have quantum value p^n. However, this *perfect* bound does not hold in general.

For the class of two-prover one-round games considered in Chapter 6, we establish a *perfect* parallel repetition theorem via semidefinite programming. It turns out that there exists a semidefinite program that characterizes the quantum value of the game. For a game G, let us denote $\varepsilon_q(G)$ to be another measure of the game defined as follows. For any strategy $\mathcal{S}$, $\varepsilon_q(G, \mathcal{S})$ denotes the difference between the probability of success and the probability of the failure. The quantity $\varepsilon_q(G)$ is the maximum over all possible quantum strategies. Then one defines the game $G \oplus G$ to be the game where the verifier plays two instances of the original game and accepts if and only if the players either win or lose both instances of the game. To obtain a perfect bound on the two-fold repetition of such a game, which we denote $G \otimes G$, it suffices to show that $\varepsilon_q(G \oplus G) = (\varepsilon_q(G))^2$. This is obtained via duality of semidefinite programming.

For the class of two-prover one-round games considered in Chapter 7, we establish a *strong* parallel repetition theorem again via semidefinite programming duality. In this case, there exists a semidefinite programming relaxation that upper bounds the quantum value of the game. It also holds that from the an optimal primal solution of the semidefinite program, one can derive a quantum strategy with good enough approximation guarantee. We obtain a parallel repetition theorem for such games by showing that $\bar{\sigma}_q(G \otimes G) = (\bar{\sigma}_q(G))^2$, where $\bar{\sigma}_q(G)$ is the optimum of the semidefinite programming relaxation.

Chapter 4

Basic Computational Complexity Theory

This chapter gives a brief introduction to computational complexity theory. Computational complexity theory is concerned with the systematic study of the amount of computational resources required to solve problems on a computational model. The study entails defining computational models, and computational resources and the corresponding complexity measures.

The most basic computational model studied in computer science is deterministic Turing machine. Without going into the model in detail, which can be found in the literature, we remark that the choice is apt to study the computational difficulty of the problems. One of the major reasons for considering the model is its apparent simplicity that also avoids the pitfall of being too detailed and hence difficult to work with. There are host of variants of the basic model that have been widely studied in the literature such as circuit based models, nondeterministic, and probabilistic Turing machines.

The resources one normally considers mimic the real-life situation. One normally considers the amount of *time* and *space* required to solve a problem as primary computational resources. To understand the complexity of an algorithm, the standard practice researchers adopt is *asymptotic analysis*. To be more precise, given an input to the algorithm, we quantify the complexity measures by functions of algorithm's input size and study the asymptotic behavior of the functions. We now proceed on to give a brief overview on the organization of the chapter.

- In Section 4.1, we briefly describe the basics of computational complexity theory followed by a glossary of complexity classes relevant to the thesis.
- In Section 4.2, we give a detailed description of the quantum computational model and basic quantum complexity classes mentioned implicitly in this thesis.

- In Section 4.3, we focus on interactive proof systems and their multi-prover variants. These models form the bedrock for few of the most important breakthroughs in theory of computation in the last twenty five years.

The field itself have seen tremendous growth in the last three decades and for a more comprehensive treatment of the subject, we refer to excellent texts by Arora and Barak [AB09], and Goldreich [Gol08].

4.1 Classical computational complexity

We assume knowledge of deterministic Turing machine and their nondeterministic and probabilistic variant. These models have been comprehensively defined in texts cited above.

A function $p : \mathbb{N} \to \mathbb{N}$ (where $\mathbb{N} = \{1, 2, \dots\}$) is said to be a *polynomial-bounded function* if there exists a polynomial-time deterministic Turing machine that outputs $1^{p(n)}$ on input 1^n for all $n \in \mathbb{N}$. Assume that $\{0,1\}$ is the alphabet set and let $\{0,1\}^*$ denote the set of all possible strings over it. A computational problem is a function f of the form

$$f : \{0,1\}^* \to \{0,1\}^*.$$

The function f is said to be *polynomial-time computable* (*exponential-time computable* if there exists a polynomial-time (exponential-time) deterministic Turing machine that outputs $f(x)$ for every valid input x. The function f is said to be *probabilistic polynomial-time computable* if there exists a polynomial-time probabilistic Turing machine that outputs $f(x)$ for every valid input x with probability at least $\frac{2}{3}$. The function f is said to be *polynomial-space computable* if there exists a polynomial-space deterministic Turing machine that outputs $f(x)$ for every valid input x. Here the amount of resource used is measured with respect to $|x|$, the size of the input. Of special interest are *decision problems* or more generally *promise problems*, which are of the form $f : \{0,1\}^* \to \{0,1\}$. A decision problem L can be partitioned into two disjoint subsets L_{yes} and L_{no} such that their union is $\{0,1\}^*$, and

$$L_{\text{yes}} = \{x \in \{0,1\}^* : f(x) = 1\} \qquad \text{and} \qquad L_{\text{no}} = \{x \in \{0,1\}^* : f(x) = 0\}.$$

The set L_{yes} represents the 'yes' instances of the problem and the set L_{no} represents the 'no' instances of the problem. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is a partition of strings over $\{0,1\}^*$ into three disjoint subsets: the set A_{yes} representing the 'yes' instances of the problem, the set A_{no} representing the 'no' instances of the problem, and the set of strings which are neither in A_{yes} nor in A_{no}, representing the disallowed instances of the problem [ESY84, Gol05].

The notion of promise problems is central to probabilistic and quantum computational complexity theory. They notion has been instrumental in formalizing "gap problems" to capture inapproximability, property testing, and introducing complete problems for various complexity classes to be discussed in this chapter. Readers should note that some of the consequences of decision problems do not generalize to promise problems [Gol05]. However, given the applications of this notion, we define complexity classes in terms of promise problems instead of decision problems. For more insights on promise problems, we refer to the Ref. [Gol05].

The following complexity classes are relevant to this thesis.

- The complexity class P is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that for any A, there exists a polynomial-time deterministic Turing machine that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.

- The complexity class BPP is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that for any A, there exists a polynomial-time probabilistic Turing machine that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$ with probability at least $\frac{2}{3}$.

- The complexity class NP is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time deterministic Turing machine such that the following holds: (i) for every string $x \in A_{\text{yes}}$, there exists a string $y \in \{0,1\}^{p(x)}$ such that the machine accepts on input (x, y), and (ii) for every string $x \in A_{\text{no}}$, there does not exists a string $y \in \{0,1\}^{p(x)}$ such that the machine accepts on input (x, y). When $x \in A_{\text{yes}}$, the string y for which the machine accepts (x, y) is often referred to as *proof* (or *certificate* or *witness*).

- The complexity class MA is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time probabilistic Turing machine such that the following holds: (i) for every string $x \in A_{\text{yes}}$, there exists a string $y \in \{0,1\}^{p(x)}$ such that the machine accepts with probability at least $\frac{2}{3}$, and (ii) for every string $x \in A_{\text{no}}$, the probability that the machine rejects over all choices of $y \in \{0,1\}^{p(x)}$ is at least $\frac{2}{3}$.

- The complexity class PP is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a polynomial-time probabilistic Turing machine that accepts every $x \in A_{\text{yes}}$ with probability strictly greater than $\frac{1}{2}$ and accepts every $x \in A_{\text{no}}$ with probability at most $\frac{1}{2}$.

- The complexity class PSPACE is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a deterministic Turing machine that uses polynomial-space and accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.

- The complexity class EXP is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a exponential-time deterministic Turing machine that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
- The complexity class NEXP is the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that there exists a exponential function $e : \mathbb{N} \to \mathbb{N}$ and a exponential-time deterministic Turing machine such that the following holds: (i) for every string $x \in A_{\text{yes}}$, there exists a string $y \in \{0,1\}^{e(x)}$ such that the machine accepts on input (x, y), and (ii) for every string $x \in A_{\text{no}}$, there does not exists a string $y \in \{0,1\}^{e(x)}$ such that the machine accepts on input (x, y).

These complexity classes are related to each other by the following well-known relationships:

$$\text{P} \subseteq \text{NP} \subseteq \text{MA} \subseteq \text{PP} \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq \text{NEXP} \qquad \text{and} \qquad \text{P} \subseteq \text{BPP} \subseteq \text{MA}.$$

The containments among these classes are not known to be strict except for $\text{P} \neq \text{EXP}$ and $\text{NP} \neq \text{NEXP}$. Proving strict containments or equality among these classes are major open problems in the area of computational complexity theory. A few other complexity classes based on the model of interactive proof systems and their variants is deferred to section 4.3. The quantum variants of interactive proof systems that are relevant to this thesis are discussed in subsequent chapters.

4.2 Quantum computational model and BQP

This section is devoted to quantum computational complexity. Our objective is to first describe a quantum computational model and then define relevant quantum complexity classes. For a comprehensive overview of the subject, we refer to the survey on quantum computational complexity [Wat09b].

Quantum circuits: The most widely used quantum computational model is the *quantum circuit model*. The topological structure of a quantum circuit is same as a classical circuit with the exception that the gates in the quantum circuits are unitary operations. Quantum circuits refer to a directed acyclic graph, where the nodes are labeled as one of the *quantum gates* from a finite set. These gets are represented by unitary operators acting on of one, two, or three qubits.

In general, the quantum circuit computing a function $f : \{0,1\}^* \to \{0,1\}^m$ on input string x will have $|x|$ input qubits that represents the input string as a quantum state

in standard basis $|x\rangle$ and may use $p(|x|)$ qubits (initialized to $|0\ldots0\rangle$ as *ancilla* qubits for some function $p : \mathbb{N} \to \mathbb{N}$. The quantum circuit transforms the state $|x\rangle |0\ldots0\rangle$ to a $(|x|+p(|x|))$-qubit quantum state. The first m qubits are designated as output qubits and the rest of the qubits are designated as garbage qubits. Measuring the first m qubits in the standard basis yields the output of the circuit.

The nature of a quantum operation is continuous which means that one cannot hope to simulate any quantum operation with a finite set of quantum gates. However, there exist finite sets of quantum gates that are *approximately universal* meaning that any quantum operation can be approximated by a composition of these gates with very high precision. This is mentioned in the following theorem, also known as the Solovay-Kitaev Theorem.

Theorem 10. *Let $\mathcal{X} = \mathbb{C}^{2^n}$ and $\mathcal{Y} = \mathbb{C}^{2^m}$. Let $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ be an arbitrary quantum operation acting on n input qubits and producing m output qubits. Then for every $\epsilon > 0$ there exists a finite set of quantum gates S and a quantum circuit $\mathcal{Q}$ acting on $(n+p(n))$-qubits using gates from S that simulates a quantum operation $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ such that*

$$\|\Psi - \Phi\|_\diamond \leq \epsilon.$$

Moreover, the number of gates $\mathcal{Q}$ requires to simulate Φ depends polynomially on $\log(1/\epsilon)$ for fixed choice of n and m.

One such finite set mentioned in Theorem 10 consists of the following three unitary operation.

1. *Hadamard gate.* A Hadamard gate is a unitary operation on one qubit as described below:
$$H|a\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^a |1\rangle \qquad \text{for} \qquad a \in \{0,1\}.$$

2. *Phase gate.* A phase gate is a unitary operation on one qubit as described below:
$$P|a\rangle = i^a |a\rangle \qquad \text{for} \qquad a \in \{0,1\}.$$

3. *Toffoli gate.* A Toffoli gate is a unitary operation on three qubits as described below:
$$T|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c \oplus a \wedge b\rangle \qquad \text{for} \qquad a,b,c \in \{0,1\}.$$

Quantum complexity classes: Before proceeding on to define quantum complexity classes, we define polynomial-time generated families of quantum circuits. A family of quantum circuits $\{\mathcal{Q}_n : n \in \mathbb{N}\}$ is said to be *polynomial-time generated* if there exists a polynomial-time deterministic Turing machine that generates a description of the circuit

$\mathcal{Q}_n$ on input 1^n. The encoding of the circuit has to be efficient in the sense that the length of the encoding is at most $p(n)$ for some fixed polynomial $p : \mathbb{N} \to \mathbb{N}$.

By restricting our attention to polynomial-time generated family of quantum circuits, we impose a *uniformity condition* on the circuit family. This allows us to visualize the circuits as quantum algorithms as follows. On input x, the Turing machine first generates the description of the circuit and then runs the quantum circuit. We remark that the notion of polynomial-time generated families of quantum circuits is to capture the notion of *efficient quantum computation*, which is precisely captured by such families of polynomial-size.

More precisely, the quantum complexity class BQP is defined as the collection of promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ such that for every problem A, there exists a polynomial-time generated family of quantum circuits $\mathcal{Q} = \{\mathcal{Q}_n : n \in \mathbb{N}\}$, where $\mathcal{Q}_n$ takes n input qubits and produces 1 output qubit, such that the following two conditions are met.

1. For every $x \in A_{\text{yes}}$, the circuit $\mathcal{Q}_{|x|}$ accepts x with probability at least $2/3$. In other words, the output qubit produces 1 with probability at least $2/3$.
2. For every $x \in A_{\text{no}}$, the circuit $\mathcal{Q}_{|x|}$ accepts x with probability at most $1/3$. In other words, the output qubit produces 1 with probability at most $1/3$.

The probabilities $2/3$ and $1/3$ are arbitrary and one can without loss of generality assume that they can be replaced by two parameters a and b, respectively such that their gap is bounded from below by a fixed inverse polynomial. Indeed, one can decrease the two-sided error probability to a quantity exponentially close to 0. The class BQP trivially contains P and BPP and is known to be contained in PP.

4.3 Interactive proofs and their variants

In this section, we discuss the model of interactive proof systems and their multi-prover variants. Interactive proof systems associate the notion of verifying a claim with an interactive process by explicitly referring to two interactive players: the *prover* and the *verifier*. The prover possess unlimited computational power and the verifier is restricted to perform efficient computation. Roughly speaking, the notion of efficient computation in the classical setting is polynomial-time computation with an additional resource: *randomness*. There is an symmetry between the computational power of the players and the main reason for such an asymmetry is that the fundamental goal is to understand the complexity of the interactive verification procedure rather than the complexity of generating the proof that convinces the verifier.

Interactive proof systems: The study of interactive proof systems was initiated by Goldwasser, Micali and Rackoff [GMR85], and Babai [Bab85]. The model of interactive proof system is formalized below. Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem and $x \in A_{\text{yes}} \cup A_{\text{no}}$ be an input string. We begin by describing the action of the prover and the verifier followed by describing their interaction.

The verifier is specified by a bounded-error probabilistic polynomial time Turing machine that takes $q_v(|x|) + q_m(|x| + r(|x|))$ bits as inputs and produces $q_v(|x|) + q_m(|x|$ bits as output. The first $q_v(|x|)$ bits represent verifier's private bits, the next $q_m(|x|$ bits represent the message bits obtained from interacting with the prover, and the final $r(|x|)$ bits represent the outcome of random coin tosses performed by the verifier. At the end of the interaction and the verifier's final computation, the first bit designates the output; if it is 1, the verifier *accepts* and if it is 0, the verifier *rejects*.

The prover is defined along similar lines except that the computational requirements are dropped. Without loss of generality, one can assume that the prover is deterministic because her goal is to maximize the acceptance probability of the verifier. At every step of the interaction, the prover computes an arbitrary function of the form

$$f : \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)} \longrightarrow \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)}.$$

The first $q_m(|x|)$ bits represent the message bits from the verifier and the last $q_p(|x|)$ represent the prover's private bits. No restriction is placed on the class of functions being computed by the prover and the number of private bits.

Given two players described above, we always assume that they are compatible in the following sense: Both players agree upon the number of message bits communicated. A $2k-1$-message interactive proof system is described by k functions computed by the prover and k functions computed by the verifier:

$$P_i : \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)} \longrightarrow \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)} \qquad \text{for all} \qquad i \in [k]$$

and

$$V_i : \{0,1\}^{q_v(|x|)} \times \{0,1\}^{q_m(|x|)} \times \{0,1\}^{r(|x|)} \longrightarrow \{0,1\}^{q_v(|x|)} \times \{0,1\}^{q_m(|x|)} \times \{0,1\}^{r(|x|)}$$

for all $i \in [k]$. The prover is sometimes referred to as $2k-1$-message prover. Messages exchanged between the two players are represented by alternations between the prover's computation and the verifier's computation. The interaction begins by prover applying P_1 and sending the first q_m bits to the verifier and ends when prover computes P_k and sends the first q_m bits. After the end of interaction, the verifier computes V_k and checks whether the first bit is 0 (reject) or 1 (accept).

A $2k$-message interactive proof system proceeds along similar lines except that the

first message is sent by the verifier instead of the prover and the verifier computes $k+1$ functions. Finally, the complexity class based on interactive proof systems is defined as follows.

Definition 11. *For two polynomial-time computable functions $c, s : \mathbb{N} \to [0,1]$, a promise problem $A = (A_{yes}, A_{no})$ is said to have a k-message (c,s)-interactive proof system between a prover and a verifier as described above, if the following holds:*

1. *The number of messages exchanged between the prover and the verifier is k, the last message is from the prover to the verifier, and the number of bits in each message is bounded by a fixed polynomial $q_m : \mathbb{N} \to \mathbb{N}$.*
2. *(Completeness) For every $x \in A_{yes}$, there exists a k-message prover such that the verifier accepts with probability at least $c(|x|)$ after exchanging k messages.*
3. *(Soundness) For every $x \in A_{no}$, for all k-message prover, the verifier accepts with probability at most $s(|x|)$ after exchanging k messages.*

For any k, the complexity class $\mathrm{IP}(k,c,s)$ *consists of all promise problems $A = (A_{yes}, A_{no})$ having a k-message interactive proof system with completeness $c(|x|)$ and soundness $s(|x|)$. The class* $\mathrm{IP}(k)$ *denotes* $\mathrm{IP}(k, 2/3, 1/3)$ *and*

$$\mathrm{IP} = \bigcup_{k \in \mathrm{poly}} \mathrm{IP}(k).$$

The complexity classes NP and MA can be viewed as a special form of interactive proof systems, where the interaction between the prover and the verifier is limited to the prover sending the proof to the verifier:

$$\mathrm{NP} = \mathrm{IP}(1,1,0) \qquad \text{and} \qquad \mathrm{MA} = \mathrm{IP}(1, 2/3, 1/3).$$

In the definition above, the parameters c and s are arbitrary polynomial-time computable functions as long as the gap $c - s$ is at least inverse polynomial. It is known that any interactive protocol of the above form can be simulated by an interactive protocol with perfect completeness ($c = 1$) and soundness exponentially close to 0. One should note that there is an asymmetry between the completeness and the soundness conditions. While the completeness of an interactive protocol can be *perfect*, the soundness cannot be made 0 without decreasing the expressive power of the model. In complexity theory terms, the class of problems accepted by an interactive proof system with $s = 0$ is precisely NP. Moreover, in contrast to the non-interactive setting, such proof systems are extremely powerful and are characterized by the following well-known relationship [LFKN92, Sha92, She92]:

$$\mathrm{IP} = \mathrm{PSPACE}.$$

The proof of IP $\subseteq$ PSPACE is attributed to Feldman [Fel86]. The proof of the other containment follows from an idea known as *arithmetization*, which views Boolean formulas algebraically by representing them as polynomials. The idea of arithmetization has found further applications in the setting of proof systems.

Many variants of interactive proof systems have been studied in literature, including but not limited to public-coin interactive proof systems [Bab85, BM88], zero-knowledge interactive proof systems [GMR89], multi-prover interactive proof systems [BOGKW88, BFL91, FL92], and interactive proof systems with competing provers [FK97]. Of these models, the models of public-coin and multi-prover interactive proof systems are relevant to this thesis.

The model of public-coin interactive proof system is a special type of interactive proof system, where roughly speaking, the verifier's randomness is public. To be more precise, before the start of a k-round (or $2k$-message) protocol the verifier tosses unbiased random coins to generate k strings

$$s_1 \in \{0,1\}^{p_1(|x|)}, \quad s_2 \in \{0,1\}^{p_2(|x|)}, \quad \ldots, \quad \textit{and} \quad s_k \in \{0,1\}^{p_k(|x|)}$$

where $p_1, p_2, \ldots, p_k : \mathbb{N} \to \mathbb{N}$ are fixed polynomials. The subsequent computation of the verifier at every round is deterministic and the verifier's message to the prover at i-th round is s_i. We say a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ admits a k-message public-coin interactive proof system with completeness c and soundness s if the items 1-3 of Definition 11 hold. The complexity class AM(k) (often referred to as Arthur-Merlin proof systems or Arthur-Merlin games) consists of the collection of problems $A = (A_{\text{yes}}, A_{\text{no}})$ that admit a k-message public-coin interactive proof systems.

It is known that the restrictive model of public-coin interactive proof system are as powerful as the general model of private-coin interactive proof system [GS89]. Given that the model is weaker and allows the prover to see verifier's randomness, the following containment is surprising:

$$\text{IP}(k) \subseteq \text{AM}(k+2).$$

The key idea behind the proof of the containment is that the verifier in Arthur-Merlin proof system convinces himself by approximately counting the number of paths that leads the verifier in the interactive proof system model to accept. If the number of accepting paths is very large, then the verifier in Arthur-Merlin proof system accepts and otherwise rejects.

Arthur-Merlin proof systems are extremely robust against error probability and one can assume that the completeness probability of such proof system is 1 and soundness inverse exponentially close to 0. Moreover, if $k \geq 2$ is constant, then it is known that

$$\text{IP}(k) \subseteq \text{AM}(k+2) \subseteq \text{AM}(2).$$

It is common to denote AM(2) as AM, which is inconsistent with the notational definition of IP. Nevertheless we follow the notations commonly used in the literature. Constant message interactive proof systems are assumed to be weaker than IP in terms of expressive power under a complexity-theoretic assumption. In fact, under a strong derandomization hypothesis, it is believed that AM = NP. In other words, under the hypothesis, interaction and randomization does not help the verifier.

Multi-prover interactive proof systems: A variant of interactive proof systems relevant to the thesis is the model of multi-prover interactive proof systems [BOGKW88, BFL91, FRS94, LS91]. Roughly speaking, the model is a form of interactive protocol between multiple co-operating provers and a verifier where the provers are not allowed to communicate with each other once the protocol starts. Without this restriction on provers not communicating with each other, the model is equivalent to single-prover interactive proof system.

Below we formalize the model of multi-prover interactive proof system. The description of the provers and the verifier is pretty much the same except that below we talk in terms of number of rounds of communication instead of the number of messages exchanged. For readers who are not familiar with the terminology, a round of communication represents a message from the verifier to the prover followed by a message from the prover to the verifier. This is a notational ambiguity, however, we do this to be consistent with the terminology used in literature.

Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem and $x \in A_{\text{yes}} \cup A_{\text{no}}$ be an input string. We begin by describing the action of the provers and the verifier followed by describing their interaction. Suppose the number of provers is m.

The verifier is specified by a bounded-error probabilistic polynomial time Turing machine that takes $q_v(|x|) + q_m(|x| + r(|x|))$ bits as inputs and produces $q_v(|x|) + q_m(|x|$ bits as output. The first $q_v(|x|)$ bits represent verifier's private bits, the next $q_m(|x|$ bits represent the message bits obtained from interacting with the provers, and the final $r(|x|)$ bits represent the outcome of random coin tosses performed by the verifier. At the end of the interaction and the verifier's final computation, the first bit designates the output; if it is 1, the verifier *accepts* and if it is 0, the verifier *rejects*.

The provers are defined along similar lines except that the computational requirements are dropped. At every step of the interaction, the provers computes an arbitrary function of the form

$$f^{(j)} : \{0,1\}^{q_m^{(j)}(|x|)} \times \{0,1\}^{q_p^{(j)}(|x|)} \longrightarrow \{0,1\}^{q_m^{(j)}(|x|)} \times \{0,1\}^{q_p^{(j)}(|x|)} \qquad \text{for all} \qquad j \in [k].$$

Here the superscript j denotes the index of the prover. The first $q_m^{(j)}(|x|)$ bits represent the message bits from the verifier and the last $q_p^{(j)}(|x|)$ represent the prover's private bits.

No restriction is placed on the class of functions being computed by the prover and the number of private bits.

A k-round multi-prover interactive proof system is described by k functions computed by each prover and k functions computed by the verifier:

$$P_i^{(j)} : \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)} \longrightarrow \{0,1\}^{q_m(|x|)} \times \{0,1\}^{q_p(|x|)} \qquad \text{for all} \qquad (i,j) \in [k] \times [m]$$

and

$$V_i : \{0,1\}^{q_v(|x|)} \times \{0,1\}^{q_m(|x|)} \times \{0,1\}^{r(|x|)} \longrightarrow \{0,1\}^{q_v(|x|)} \times \{0,1\}^{q_m(|x|)} \times \{0,1\}^{r(|x|)}$$

for all $i \in [k+1]$. The provers and the verifier are said to be compatible if they agree upon k and if

$$q_m = \sum_{j=1}^{m} q_m^{(j)}.$$

More specifically, after verifier performs his computation, he partitions the message bits into j parts, the j-th part being of size $q_m^{(j)}$, and sends the j-th part to the corresponding prover. Messages exchanged between the provers and the verifier are represented by alternations between the provers' computation and the verifier's computation. The interaction begins by verifier applying V_1 and sending the corresponding messages to the provers and ends when provers compute their last function in the sequence and send the message bits. After the end of interaction, the verifier computes V_{k+1} and checks whether the first bit is 0 (reject) or 1 (accept).

Definition 12. *For two polynomial-time computable functions $c, s : \mathbb{N} \to [0,1]$, a promise problem $A = (A_{yes}, A_{no})$ is said to have a m-prover k-round (c,s)-interactive proof system between m provers each with unlimited computational power who are not allowed to communicate with each other, and a verifier that executes* BPP *strategy, if the following holds:*

1. *The number of messages exchanged between the provers and the verifier is $2k$, the last message is from the prover to the verifier, and the number of bits in each message is bounded by a fixed polynomial $p : \mathbb{N} \to \mathbb{N}$.*
2. *(Completeness) For every $x \in A_{yes}$, there exist m provers that the cause the verifier to accept with probability at least $c(|x|)$ after k rounds.*
3. *(Soundness) For every $x \in A_{no}$, for all m provers, the verifier accepts with probability at most $s(|x|)$ after k rounds.*

For any k, the complexity class MIP(m,k,c,s) *consists of all promise problems $A = (A_{yes}, A_{no})$ having a k-round m-provers interactive proof system with completeness prob-*

ability c and soundness probability s. The class $\mathrm{MIP}(m,k)$ *denotes* $\mathrm{MIP}(m,k,2/3,1/3)$ *and*

$$\mathrm{MIP} = \bigcup_{m,k \in \mathrm{poly}} \mathrm{MIP}(m,k).$$

Multi-prover interactive proof systems are extremely robust against error probabilities. It is known that multi-prover interactive proof systems can have perfect completeness and the soundness can be made exponentially close to 0. Moreover, such proof systems are extremely powerful as characterized by the following well-known relationship:

$$\mathrm{MIP} = \mathrm{NEXP}.$$

This was first established by Babai, Fortnow, and Lund [BFL91] and subsequently improved in a paper by Feige and Lovász [FL92] culminating in the characterization of NEXP by $\mathrm{MIP}(2,1)$, the model of two-prover one-round interactive proof systems. The above characterization is surprising because if we do not allow the verification procedure to be interactive and randomized, then the proof system (with multiple provers) can only accept any problem in NP. The characterization also implies that unlike in the single prover case, multi-prover interactive proof systems can be parallelized to one round without losing its expressive power. Moreover, the seemingly restrictive model is also extremely robust against error probability. That is, any promise problem in NEXP admits a two-prover one-round proof system with perfect completeness and soundness exponentially close to zero.

The original motivation of introducing the model of multi-prover interactive proofs was zero-knowledge proof systems and it was shown that NP has perfect zero-knowledge proofs [BOGKW88]. The focus on multi-prover interactive proof systems led to the development of probabilistically checkable proof (PCP) systems [ALMSS98, AS98] and their connection with hardness of approximation [Hås96, BGS98, Hås01]. It is not an understatement that the development of probabilistically checkable proofs has been one of the crowning achievements of computational complexity theory.

Given that two-prover one-round interactive proof systems have perfect completeness, one simple way of reducing the soundness is to repeat multiple instances of the protocol simultaneously. Although it increases the communication cost between the provers and the verifier, the new protocol has the property that it is still a one-round protocol. This kind of repetition is called *parallel repetition*. For a while it was thought that if the original soundness is s, then the soundness of the proof system where k instances of the protocol is repeated is s^k. This was proved to be untrue [For89, FL92] and subsequently a great body of work was devoted in coming up with a concrete bound on k-fold repetition of the protocol [CCL90, CCL92, FV02, Ver95, Ver96]. A concrete bound was first established by Raz [Raz98]. This was recently simplified and improved by Holenstein [Hol07] and

Rao [Rao08]. In the classical case, Feige and Verbitsky [FV02] showed that Raz's bound is nearly tight. A major part of this thesis concerns with parallel repetition of two-prover one-round proof systems, where the provers are allowed to process quantum information but the communication channel between the provers and the verifier is classical.

For the purpose of this thesis, it is often convenient to abstract two-prover one-round interactive proof systems as games played between two co-operating players and a referee. One can make the association between two-prover one-round games with the model of two-prover one-round interactive proof systems precise by saying that on every input x, the verifier plays a game with the provers such that the value of the game (maximum success probability of the provers) is at least $c(|x|)$ for all $x \in A_{\text{yes}}$ and is at most $s(|x|)$ for all $x \in A_{\text{no}}$ (refer to Definition 12). Of course, the game depends on the input string x.

Chapter 5

Expressive Power of QIP

This chapter is based on the Ref. [JJUW09] and is a joint work with Rahul Jain, Zhengfeng Ji, and John Watrous.

The chapter introduces the class QIP, the collection of promise problems that admit quantum interactive proof systems. Quantum interactive proof systems are quantum analogue of interactive proof systems wherein the prover and the verifier are allowed to exchange and process quantum information. It is immediate that QIP contains IP and hence PSPACE [LFKN92, Sha92, She92], since the additional ability of processing quantum information is no hindrance to the verifier's ability to behave classically and force the prover to behave classically. The main result of this chapter is the reverse containment:

$$\text{QIP} \subseteq \text{PSPACE}.$$

The containment implies that the any promise problem that admits a quantum interactive proof system also admits a classical interactive proof system. The proof relies on an alternate characterization of PSPACE in terms of Boolean circuits and several known facts on quantum interactive proof systems, which are briefly discussed in the subsequent section.

The organization of this chapter is as follows.

- In Section 5.1, we formally define the model of quantum interactive proof systems and mention the known facts on quantum interactive proof systems that allows us to simplify the model to a variant of quantum interactive proof systems, referred to as single-coin quantum interactive proof systems.
- In Section 5.2, we discuss an alternate characterization of PSPACE in terms of polynomial-space uniform families of Boolean circuits of polynomial depth and ma-

trix multiplicative weights update method, a meta-algorithm that forms the basis of our algorithm.

- In Section 5.3, we characterize the set of potential quantum strategies adopted by the prover in single-coin quantum interactive proof systems as a simple exponential-size semidefinite program.
- In Section 5.4, we present the algorithm for solving the semidefinite program that characterizes the single-coin quantum interactive proof systems followed by a detailed analysis of the algorithm and its implementation by polynomial-space uniform families of Boolean circuits of polynomial depth.
- Finally in Section 5.5, we discuss two recent results reported by Jain and Yao [JY11] and Gutoski and Wu [GW10] that are related to the main result and techniques used in this chapter.

5.1 Quantum interactive proof systems

This section introduces the model of quantum interactive proof systems. Similar to the classical case, quantum interactive proof systems associate the notion of efficient verification of a purported claim with an interactive process by explicitly referring to two interactive players: the *prover* and the *verifier*. The only difference between classical and quantum interactive proof systems is that in the latter both players are allowed to exchange and process quantum information. The study of quantum interactive proofs was initiated by Watrous [Wat03] (the conference version first appeared in 1999).

The model of quantum interactive proof system is formalized below. Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem and $x \in A_{\text{yes}} \cup A_{\text{no}}$ be an input string. We begin by describing the action of the prover and the verifier followed by describing their interaction.

The verifier is defined by polynomial-time generated families of quantum circuits of polynomial-size. More specifically, the verifier can be represented by a unitary operator acting on $q_v(|x|)+q_m(|x|)$ qubits. The first $q_v(|x|)$ is verifier's private qubits and the rest of the qubits represent the quantum message exchanged between the prover and the verifier. Of course, the unitary operator is an efficiently implementable operator in the sense that it can be implemented *exactly* by the finite set of gates described in Section 4.2. The first qubit of the verifier represents the output of the interaction with the prover. Measuring this qubit in standard basis reveals the verifier's output; if it is 1, the verifier *accepts* and if it is 0, the verifier *rejects*.

Similarly, the prover is represented by quantum circuits. However, we drop all the computational requirements such as polynomials-size quantum circuits or polynomial-time

generation of the circuits. The prover is represented by a unitary operator acting on $q_m(|x|) + q_p(|x|)$ qubits, the first $q_m(|x|)$ representing the qubits received form the verifier and the last $q_p(|x|)$ representing the prover's private qubits. One places no bound on number of prover's private qubits, however, by the singular value decomposition (Theorem 1), one can assume that $q_m(|x|) = q_p(|x|)$.

The interaction between the prover and the verifier is described below. Imagine a quantum interactive proof systems in which a prover and a verifier exchange $2k$ message. Alternately, the number of rounds of communication between the players is k. The verifier is described by $k+1$ quantum circuits

$$\{V_0(x), V_1(x), \ldots, V_k(x)\}$$

and the prover is described by k circuits

$$\{P_1(x), P_2(x), \ldots, P_k(x)\}$$

such that they operate on same number of message qubits. The interaction begins by verifier applying $V_0(x)$ on two registers of size $q_v(|x|)$ and $q_m(|x|)$, respectively, initialized to state $|0\ldots0\rangle \otimes |0\ldots0\rangle$. The interaction begins by verifier sending the qubits designates as message to the prover, who applies $P_1(x)$ on his private qubits (initialized in any state she desires) and the message qubits. The interaction proceeds by prover sending the message qubits back to the verifier who applies his next operation. This process is alternated until prover applies his final map $P_k(x)$ and sends the message qubits to the verifier. After the interaction ends, the verifier applies his final operation $V_k(x)$ and measures the first qubit in the standard basis, the output of which denotes whether the verifier accepts or rejects.

A $2k-1$ message quantum interactive proof system proceeds in a similar fashion except that the prover initiates the interaction by sending the first message. One assumes that the verifier's private qubits are initialized to $|0\ldots0\rangle$. Upon receiving the message, the verifier proceeds to apply his first operation, and the interaction proceeds as before until verifier applied his final operation and measures the first qubit in standard basis to decide whether to accept or reject.

In a k-message quantum interactive proof system, the prover is usually referred to as k-message quantum prover. Finally, the complexity class based on quantum interactive proof system is defined below.

Definition 13. *For two polynomial-time computable functions $c, s : \mathbb{N} \to [0,1]$, a promise problem $A = (A_{yes}, A_{no})$ is said to have a k-message (c,s)-quantum interactive proof system between a prover and a verifier as described above, if the following holds:*

1. *The number of messages exchanged between the prover and the verifier is k, the last*

message is from the prover to the verifier, and the number of qubits in each message is bounded by a fixed polynomial $q_m : \mathbb{N} \to \mathbb{N}$.

2. *(Completeness) For every $x \in A_{yes}$, there exists a k-message quantum prover such that the quantum verifier accepts with probability at least $c(|x|)$ after exchanging k messages.*
3. *(Soundness) For every $x \in A_{no}$, for all k-message quantum prover, the quantum verifier accepts with probability at most $s(|x|)$ after exchanging k messages.*

For any k, the complexity class $\mathrm{QIP}(k, c, s)$ *consists of all promise problems $A = (A_{yes}, A_{no})$ having a k-message quantum interactive proof system with completeness c and soundness s. The class* $\mathrm{QIP}(k)$ *denotes* $\mathrm{QIP}(k, 2/3, 1/3)$ *and*

$$\mathrm{QIP} = \bigcup_{k \in \mathrm{poly}} \mathrm{QIP}(k).$$

In the above definition, the parameters c and s can be arbitrary polynomially-bounded functions such that their difference $c - s$ is at least inverse polynomial. It is known that any quantum interactive protocol of the above form can be simulated by a quantum interactive protocol with perfect completeness and soundness exponentially close to zero. This and various other facts are summarized below.

As mentioned earlier, the study of quantum interactive proofs was initiated by Watrous [Wat03] (the conference version of the paper appeared in 1999), where he showed that any problem in PSPACE admits a three-message quantum interactive proof system. The result is extremely surprising in the wake of the classical case, where it is conjectured that PSPACE does not admit a constant message interactive proof system. The belief is based on a well known complexity-theoretic conjecture that states that such a result would imply that polynomial hierarchy collapses to the second level [Bab85, GS89]. Similar to the classical case, many variants of quantum interactive proof systems have been studied, including public-coin quantum interactive proof systems [MW05], zero-knowledge quantum interactive proof systems [HKSZ08, Kob08, Wat02, Wat09a], multi-prover quantum interactive proof systems [CHTW04a, KKMTV11, KKMV09, KM03], and quantum interactive proof systems with competing provers [GW05, GW07]. The focus of this section is on quantum interactive proof systems and the subsequent simplifications of the most general model to a model more suited for our purpose.

Subsequent to the containment PSPACE $\subseteq$ QIP(3), Kitaev and Watrous [KW00] established a series of results on quantum interactive proof systems that simplified the model considerably and exhibited some nice properties of it. They are summarized below.

1. Every quantum interactive proof system with completeness c and soundness s can be transformed into an equivalent quantum interactive proof system with perfect completeness and soundness bounded from above by $1 - (c - s)^2/2$. This transformation is

obtained at the cost of an extra round of communication between the prover and the verifier. In complexity theory terms,

$$\mathrm{QIP}(k, c, s) \subseteq \mathrm{QIP}(k + 2, 1, 1 - (c - s)^2/2).$$

2. Every quantum interactive proof system with completeness 1 and soundness bounded away from 1 can be transformed into a three-message quantum interactive proof system with perfect completeness and soundness bounded away from 1. The soundness depends on the original soundness and the number of messages exchanged between the prover and the verifier. In complexity theory terms,

$$\mathrm{QIP}(k, 1, 1 - \epsilon) \subseteq \mathrm{QIP}(3, 1, 1 - \epsilon^2/4k).$$

3. Multiple executions of the protocol of a three-message quantum interactive proof system simultaneously (in parallel) with perfect completeness reduces the soundness to a quantity that is exponentially close to zero. The exponential reduction is *perfect* in the following sense. If the soundness of one execution of the protocol is s, then the soundness of r simultaneous executions in s^r. This implies that

$$\mathrm{QIP}(3, 1, s) \subseteq \mathrm{QIP}(3, 1, s^r)$$

   for any polynomial $r : \mathbb{N} \to \mathbb{N}$. This follows from characterizing the success probability of a three-message protocol by the diamond norm of a super-operator Φ defined by verifier's operation, $\|\Phi\|_\diamond$, together with the multiplicative property of the norm under the tensor product operation. The result can be extended to k-message quantum interactive proof systems for any polynomially-bounded function $k : \mathbb{N} \to \mathbb{N}$ [Gut09].

4. The maximum success probability of the prover in a three-message quantum interactive proof system can be expressed as an exponential-size semidefinite program. The semidefinite program satisfies the conditions of ellipsoid method and hence can be efficiently approximated in the size of the operators defining it. This shows that

$$\mathrm{QIP} \subseteq \mathrm{EXP}.$$

The fact that the model of three-message quantum interactive proof systems are as powerful as the general model leaves us with four complexity classes based on the model:

$$\mathrm{QIP}(0) = \mathrm{BQP}, \quad \mathrm{QIP}(1) = \mathrm{QMA}, \quad \mathrm{QIP}(2), \quad \text{and} \quad \mathrm{QIP}(3) = \mathrm{QIP}.$$

The most mysterious of the four classes is QIP(2) and the present status of this class is that it lies in PSPACE [JUW09]. This was known prior to the main result of this chapter. The class QMA is the quantum analogue of the complexity classes NP and MA. Unlike

QIP, it is not known whether QIP(2) and QMA can have perfect completeness. However, both classes are extremely robust against error probabilities.

We briefly remark that Marriott and Watrous [MW05] demonstrated an alternate method of error reduction for QMA. Most of the error reduction techniques blow up the size of the messages of the prover and the verifier. The alternate error reduction technique for QMA, as described by Marriott and Watrous, achieves exponentially small two-sided error probabilities without increasing the size of the message sent by the prover. For details, we refer the interested readers to their paper. Using this fact, Marriott and Watrous gave an alternate proof of the containment

$$\mathrm{QMA} \subseteq \mathrm{PP}.$$

Coming back to quantum interactive proofs, Marriott and Watrous also demonstrated that it is always possible to transform a given quantum interactive proof system into one with a simpler form as described below in detail. They showed that any three-message quantum interactive proof system can be converted into a three-message quantum interactive proof system, where the verifier's message to the prover is a uniformly random bit. We refer to such proof systems as *single-coin quantum interactive proof systems*. In greater detail, such a three-message protocol is described below.

1. The verifier receives a register X from the prover and sends a bit $a \in \{0,1\}$, chosen uniformly at random to the prover.
2. Upon receiving a, the prover applies a quantum operation on a second register Y and sends Y to the verifier.
3. The verifier measures the pair (X,Y) with respect to two-outcome projective measurement $\{\Pi_0^a, \Pi_1^a\}$ that depends on a. Here the operator Π_1^a corresponds to outcome *accept*.

We remark that the three-message protocol described above does not follow the description of quantum interactive proof systems strictly. However, it is easy to describe the above protocol in a way that agrees with Definition 13 and the preceding discussion.

For the remainder of this chapter, we will denote $m = p(n)$ for some fixed polynomial $p : \mathbb{N} \to \mathbb{N}$ and assume without loss of generality, that the size of registers X and Y are m-qubits. Consequently, the associated complex Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$ are isomorphic to $\mathbb{C}^M$ for $M = 2^m$. Marriott and Watrous proved that every problem problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ in QIP has a quantum interactive proof system of the above form, where the completeness is perfect and the soundness is bounded from above by $1/2 + \varepsilon$ for any desired constant $\varepsilon > 0$. In fact ε can be made exponentially small, however, for our needs it suffices to consider it to be a constant. Note that if we allow the verifier to send only one uniformly random bit, then the prover can always succeed with probability $1/2$ by randomly guessing

the bit and then sending a state in (X, Y) that causes the verifier to accept with probability 1 for that particular choice. To reduce the soundness to something exponentially close to zero, one can execute multiple instances of the protocol simultaneously.

5.2 Some background information

In this section, we discuss two different topics that form the basis of the proof. The first topic is a complexity-theoretic result due to Borodin [Bor77] and the second topic is a meta-algorithm that has found applications in various areas in theoretical computer science.

5.2.1 Boolean circuits and PSPACE

In this section, we discuss an alternate characterization of PSPACE in terms of Boolean circuits. To prove that QIP is contained in PSPACE, we will demonstrate that the SDP characterization of single-coin quantum interactive proofs can be solved by such circuits. This is done in detail in Section 5.4.

The complexity class NC is defined as the class of all functions (including promise problems) computable by families of *logarithmic-space uniform* families of Boolean circuits of *polylogarithmic depth*. Such circuits can be represented as a directed acyclic graph with a polynomial number of vertices. The restriction of polylogarithmic depth is an abstraction of massive parallelizability. The restriction of logarithmic-space uniformity means that there exists a logarithmic-space deterministic Turing machine that generates the description of the circuit. Consequently, the circuit is polynomial in size with the uniformity condition allowing the circuit to be viewed as as algorithm The problems in class NC are instances of efficiently computable problems and therefore are contained in P.

Also, let us denote NC(poly) to be the class that consists of all functions computable by *polynomial-space uniform* families of Boolean circuits having *polynomial-depth*. The NC(poly) circuits satisfy the following property. Let $f : \{0,1\}^* \to \{0,1\}^*$ is a function computable in NC(poly) and $g : \{0,1\}^* \to \{0,1\}^*$ is a function computable in NC. Then the composition $g \circ f$ (if it is well defined) is also computable in NC(poly). This follows from the most straightforward way of composing the families of circuits that compute f and g, along with the observation that the size of $f(x)$ is at most exponential in $|x|$.

The NC(poly) circuits can be potentially exponential in size and hence can be trivially simulated in exponential-time. They do not necessarily represent efficient computation. However, the restriction of polynomial depth is a significant restriction in their computational power. This is best exemplified by the classic result of Borodin [Bor77]. A corollary

of Borodin's result is that when restricted to promise problems, we have

$$\text{NC(poly)} = \text{PSPACE}.$$

The main idea behind the containment NC(poly) $\subseteq$ PSPACE is that in polynomial-space it is possible to do a space efficient depth-first traversal of a directed acyclic graph of polynomial-depth aka NC(poly) circuit.

The other containment is not hard to follow. One of the ways of showing the containment is to show that a PSPACE-complete problem can be solved by NC(poly) circuits. We precisely do this below using the following problem.

> *Input.* An exponential-size graph described succinctly by a polynomial-time generated family of polynomial-size circuits and two designated nodes s and t.
>
> *Ouput.* A Boolean value representing whether there exists a path between s and t. In other words, the value is 1 if and only if there exists a path between s and t.

When we say that the graph is described succinctly by a circuit, we mean that the circuit takes two inputs s and t and outputs 1 if and only if there is an edge between s and t. One should also note that a "scaled-down" version of the problem is complete for L, the class of problems solvable on a deterministic Turing machine that uses logarithmic-space. The fact that the above problem is PSPACE-complete has been shown by Lozano and Balcázar [LB90].

Now we describe an NC(poly) circuit for the above problem. The algorithm for the problem is a two-step procedure as follows.

1. Compute the adjacency matrix of the exponential-size graph from its circuit description. Let the adjacency matrix be denoted A.
2. Compute the transitive-closure of the graph T_A using the identity

   $$T_A = \mathbb{I} + A + A^2 + \cdots + A^N$$

   where N is the number of vertices in the graph. Here the addition and multiplication are over Boolean variables. Output the (s, t)-entry of T_A.

Given the circuit description of the problem, the first step can be done exactly in NC(poly) by computing every entry of the adjacency matrix using the circuit describing the graph in parallel. The second step can be done in NC (where the input is the size of the graph). This is straightforward to see because it involves Boolean matrix addition and multiplication for which NC algorithm exists. The matrix computation A^i can be done using repeated

squaring trick, which increases the depth by $\lceil \log_2 N \rceil$. The final computation involves outputting the (s,t)-entry of T_A, which by the property of transitive-closure of the graph shows an existence of a path between s and t. We see that composing the two circuits gives us the desired NC(poly) for the above problem.

5.2.2 Matrix multiplicative weights update method

In this section, we discuss the matrix variant of multiplicative weights update method. The proof of our main result is heavily based on this method. Before we proceed, we describe the basic multiplicative weights update method for some intuition.

Multiplicative weights update method: Let us consider a scenario where we have n experts, each making a prediction on the outcome of a game. They win or lose the bet depending on their prediction. Our goal is to devise an algorithm whose average loss over long run is close to the average loss of the best expert. The algorithm proceeds by maintaining a certain distribution over the experts, where the weight associated with each expert is tied with the expert's performance in the previous rounds. At every round, the weights on the experts are updated by a multiplicative rule and hence the name, multiplicative weights update method.

The algorithm works in a more general scenario where the number of possible outcomes is non-binary. To account for this general setting, we assume that a cost is associated with choosing an expert. For now we assume that the cost associated with n experts at some round t is given to the algorithm in form of a vector $c^{(t)}$. The cost is revealed only after the algorithm picks an expert. In more concrete settings, the cost is usually formulated in terms of the problem in hand. The only assumption we make about the cost is that

$$0 \leq c^{(t)} \leq \bar{e} \tag{5.1}$$

where $\bar{e}$ is the vector of all 1's. The assumption is due to technical reasons. Often the assumption does not hold and we briefly mention how to handle this case after describing the algorithm.

Suppose the algorithm picks the experts from a probability distribution $p^{(t)}$. Then the expected cost incurred by the algorithm is given by $\left\langle c^{(t)}, p^{(t)} \right\rangle$. The total expected cost of the algorithm and the cost of always choosing the best expert after T rounds are

$$\sum_{t=0}^{T-1} \left\langle c^{(t)}, p^{(t)} \right\rangle \qquad \text{and} \qquad \min_i \left\{ \sum_{t=0}^{T-1} c^{(t)}[i] \right\}$$

respectively. The algorithm is described below.

Let $\varepsilon \leq 1/2$ and the initial weight for an expert i is $w^{(0)}[i] := 1$. Repeat for every t from 0 to $T-1$:

1. Choose expert i according to the distribution $p^{(t)}$, where

$$p^{(t)} = \left\{ \frac{w^{(t)}[1]}{w^{(t)}_{sum}}, \ldots, \frac{w^{(t)}[n]}{w^{(t)}_{sum}} \right\} \qquad \text{and} \qquad w^{(t)}_{sum} = \sum_{i=1}^{n} w^{(t)}[i].$$

2. Given the cost vector $c^{(t)}$, update the weights of the expert by the following rule:

$$w^{(t+1)}[i] = w^{(t)}[i](1-\varepsilon)^{c^{(t)}[i]}.$$

The choice of the uniform weight at the start of the algorithm represents our complete lack of knowledge about the experts. The algorithm monitors the performance of experts and skews the distribution in favor of the experts who performed well in the previous iterations. The skewing of the distribution is done by using the update rule described in step 2.

It holds that after T rounds, the average cost incurred by the algorithm and the average cost of any expert i follows the following relationship:

$$\frac{1}{T}\sum_{t=0}^{T-1} \left\langle c^{(t)}, p^{(t)} \right\rangle \;\leq\; \frac{1+\varepsilon}{T}\sum_{t=0}^{T-1} c^{(t)}[i] \;+\; \frac{\log n}{\varepsilon T}.$$

The proof can be found in the PhD thesis of Kale [Kal07]. When designing algorithms for some problem using the multiplicative weights update method, often (5.1) does not hold. However, there exists a parameter χ such that the cost vector satisfies

$$0 \leq c^{(t)} \leq \chi \bar{e}.$$

The parameter χ depends on the problem in hand. In such a scenario, we redefine the cost by dividing the original cost by χ. The performance of the method depends on χ as follows:

$$\frac{1}{T}\sum_{t=0}^{T-1} \left\langle c^{(t)}, p^{(t)} \right\rangle \;\leq\; \frac{1+\varepsilon}{T}\sum_{t=0}^{T-1} c^{(t)}[i] \;+\; \frac{\chi \log n}{\varepsilon T}.$$

Matrix multiplicative weights update method: The matrix variant of multiplicative weights update method is a generalization of the algorithm described above to Hermitian operators in the most natural way. We discuss two formalism in this case: the *loss matrix formalism* and the *gain matrix formalism*. Roughly speaking, the loss matrix formalism

relates the average loss of the matrix multiplicative weights update method with the minimum average loss over all experts. On the other hand, the gain matrix formalism relates the average gain of the matrix multiplicative weights update method with the maximum average gain over all experts. The notions of loss matrices and gain matrices is made precise below.

We first describe the loss matrix formalism. Fix a complex Euclidean space $\mathcal{X} = \mathbb{C}^n$. One associates every rank-one density operator $uu^* \in \mathrm{D}(\mathcal{X})$ with an expert. Given a *loss matrix* $C \in \mathrm{Herm}(\mathcal{X})$, the loss of an expert uu^* is given by $\langle C, uu^* \rangle$. Similar to the basic setting, we assume that the loss matrix satisfies the relationship

$$0 \preceq C \preceq \mathbb{I}_{\mathcal{X}}.$$

This is the only assumption we make on the algorithm. The most natural way of representing a probability distribution over the experts is by using density operators. More specifically, given an ensemble $(p[u], uu^*)$, one uses the density operator

$$\rho = \sum_u p[u] \; uu^*$$

to represent the distribution over the experts. Conversely, given a density operator ρ, the spectral decomposition gives a canonical distribution over the experts. With these associations in mind, we describe the matrix multiplicative weights update method below.

Let $\varepsilon \leq 1/2$ and initialize the weight operator $W^{(0)} := \mathbb{I}_{\mathcal{X}}$. Repeat for every t from 0 to $T - 1$:

1. Use the density operator $\rho^{(t)}$ defined as

$$\rho^{(t)} := \frac{W^{(t)}}{\mathrm{Tr}\left(W^{(0)}\right)}.$$

2. Observe the cost operator $C^{(t)}$ and update the weights of the expert by the following rule:

$$W^{(t+1)} = \exp\left(-\varepsilon \sum_{j=0}^{t} C^{(j)}\right).$$

It holds that after T rounds, the average loss incurred by the algorithm and the average loss of any expert uu^* follows the relationship:

$$\frac{\exp(-\varepsilon)}{T} \sum_{t=0}^{T-1} \left\langle C^{(t)}, \rho^{(t)} \right\rangle \;\leq\; \frac{1}{T} \sum_{t=0}^{T-1} \left\langle C^{(t)}, uu^* \right\rangle \;+\; \frac{\log n}{\varepsilon T}. \tag{5.2}$$

The minimum loss over all experts in this setting is

$$\min_u \left\{ \sum_{t=0}^{T-1} \left\langle C^{(t)}, uu^* \right\rangle \right\} = \lambda_n \left(\sum_{t=0}^{T-1} C^{(t)} \right).$$

The above follows simply from the variational characterization of eigenvalues of Hermitian matrices.

Now we turn to describe the gain matrix formalism. The setting and the assumption is same except that we talk about gain matrices instead of loss matrices. The algorithm is essentially the same except that the update rule is

$$W^{(t+1)} = \exp\left(\varepsilon \sum_{j=0}^{t} C^{(j)} \right).$$

It holds that after T rounds, the average gain of the algorithm and the average gain of any expert uu^* follows the relationship:

$$\frac{\exp(\varepsilon)}{T} \sum_{t=0}^{T-1} \left\langle C^{(t)}, \rho^{(t)} \right\rangle \;\geq\; \frac{1}{T} \sum_{t=0}^{T-1} \left\langle C^{(t)}, uu^* \right\rangle \;-\; \frac{\log n}{\varepsilon T}. \tag{5.3}$$

Again from the variational characterization of eigenvalues of Hermitian matrices, the maximum gain over all experts in this setting is

$$\max_u \left\{ \sum_{t=0}^{T-1} \left\langle C^{(t)}, uu^* \right\rangle \right\} = \lambda_1 \left(\sum_{t=0}^{T-1} C^{(t)} \right).$$

In the algorithm described below, we use both the loss and gain matrix formalism.

5.3 SDP for single-coin quantum interactive proofs

In this section, we demonstrate the semidefinite programming characterization of the single-coin quantum interactive proof systems. Recall that a single-coin quantum interactive proof system proceeds by prover sending a register X upon which the verifier sends a random bit a back to the prover. The prover finally sends another register Y. The verifier measures the two registers and decides to accept or reject depending upon the outcome of the measurement operation.

Given two m-qubit registers, X and Y whose content depends on the bit $a \in \{0, 1\}$ chosen uniformly at random, any strategy adopted by the prover can be exactly characterized

by density operators

$$\{\rho_a : a \in \{0,1\}\} \subset \mathrm{D}\left(\mathcal{X} \otimes \mathcal{Y}\right)$$

that satisfy the following equation

$$\mathrm{Tr}_{\mathcal{Y}}(\rho_0) = \mathrm{Tr}_{\mathcal{Y}}(\rho_1). \tag{5.4}$$

The necessity of the condition follows from the fact that the content of X is independent of the bit a send to the prover. The sufficiency of the condition follows from Theorem 5, which states that if the prover holds a purification of the quantum state in X, then the prover can generate all possible states $\xi \in \mathrm{D}\left(\mathcal{X} \otimes \mathcal{Y}\right)$ by applying a suitable quantum operation on Y such that $\mathrm{Tr}_{\mathcal{Y}}(\xi)$ agrees with the quantum state in X.

Moreover, given that the measurement operators corresponding to outcome *accept* are Π_1^0 and Π_1^1, it immediately follows that the maximum success probability of the prover is given by the expression

$$\frac{1}{2}\left\langle \Pi_1^0, \rho_0 \right\rangle + \frac{1}{2}\left\langle \Pi_1^1, \rho_1 \right\rangle \tag{5.5}$$

subject to the constraint that $\rho_0, \rho_1 \in \mathrm{D}\left(\mathcal{X} \otimes \mathcal{Y}\right)$ satisfy (5.4). This is clearly an instance of a semidefinite program, where one wishes to maximize (5.5) subject to the constraint that two density operators ρ_0 and ρ_1 satisfy (5.4).

We first modify the semidefinite program to a form that is well-suited for our approach. Let

$$P_0 = \Pi_1^0 + \alpha\Pi_0^0 \qquad \text{and} \qquad P_1 = \Pi_1^1 + \alpha\Pi_0^1$$

for a positive number α. We will choose $\alpha = 4$ later on. Any smaller choice of α will work, however, the choice of $\alpha = 4$ is sufficient for the correct operation of the algorithm. Since the elements of the sets $\{\Pi_1^0, \Pi_0^0\}$ and $\{\Pi_1^1, \Pi_0^1\}$ are projection operators, and $\Pi_1^0 + \Pi_0^0 = \mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}}$ and $\Pi_1^1 + \Pi_0^1 = \mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}}$, it follows that P_0 and P_1 are positive definite operators, with inverses given by

$$P_0^{-1} = \Pi_1^0 + \frac{1}{\alpha}\Pi_0^0 \qquad \text{and} \qquad P_1^{-1} = \Pi_1^1 + \frac{1}{\alpha}\Pi_0^1.$$

A simple calculation shows that

$$\frac{1}{2}\left\langle \Pi_1^0, \rho_0 \right\rangle + \frac{1}{2}\left\langle \Pi_1^1, \rho_1 \right\rangle \leq \frac{1}{2}\left\langle P_0^{-2}, \rho_0 \right\rangle + \frac{1}{2}\left\langle P_1^{-2}, \rho_1 \right\rangle \leq \frac{1}{2}\left\langle \Pi_1^0, \rho_0 \right\rangle + \frac{1}{2}\left\langle \Pi_1^1, \rho_1 \right\rangle + \frac{1}{\alpha^2}.$$

Moreover, since P_a is a positive definite operator for both choices of a, it holds that $P_a\rho_a P_a$ range over all positive semidefinite operators as ρ_a does. Hence, by defining operators Q_0 and Q_1 as

$$Q_0 = P_0^{-1}\rho_0 P_0^{-1} \qquad \text{and} \qquad Q_1 = P_1^{-1}\rho_1 P_1^{-1}$$

the semidefinite program described by (5.4) and (5.5) is equivalent to the semidefinite

program, where one maximizes the quantity

$$\mathrm{Tr}\left(\frac{1}{2}Q_0 + \frac{1}{2}Q_1\right)$$

over all positive semidefinite operators Q_0 and Q_1 that satisfy the positive semidefinite inequalities $\mathrm{Tr}_{\mathcal{Y}}(P_0 Q_0 P_0) \preceq \xi$ and $\mathrm{Tr}_{\mathcal{Y}}(P_1 Q_1 P_1) \preceq \xi$ for some choice of $\xi \in \mathrm{D}(\mathcal{X})$. Let P and Q be the operators

$$P = \begin{pmatrix} (\Pi_1^0 + \alpha\Pi_0^0) & 0 \\ 0 & (\Pi_1^1 + \alpha\Pi_0^1) \end{pmatrix} \in \mathrm{Pos}(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}) \tag{5.6}$$

and

$$Q = \begin{pmatrix} \frac{1}{2}Q_0 & 0 \\ 0 & \frac{1}{2}Q_1 \end{pmatrix} \in \mathrm{Pos}(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}).$$

Here the complex Euclidean space $\mathcal{Z} = \mathbb{C}^2$ can be thought of as the space corresponding to verifier's coin toss. With this simple modification, the semidefinite program for single-coin quantum interactive proof system can be written as follows. We denote the resulting semidefinite program as $\mathrm{SDP}(P)$.

Primal problem		Dual problem	
maximize:	$\mathrm{Tr}(Q)$	minimize:	μ
subject to:	$\mathrm{Tr}_{\mathcal{Y}}(PQP) \preceq \frac{1}{2}\mathbb{I}_{\mathcal{Z}} \otimes \xi$,	subject to:	$\frac{1}{2}\mathrm{Tr}_{\mathcal{Z}}(R) \preceq \mu\mathbb{I}_{\mathcal{X}}$,
	$Q \in \mathrm{Pos}(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y})$,		$P(R \otimes \mathbb{I}_{\mathcal{Y}})P \succeq \mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}}$,
	$\xi \in \mathrm{D}(\mathcal{X})$.		$R \in \mathrm{Pos}(\mathcal{Z} \otimes \mathcal{X})$.

The primal and dual problems of $\mathrm{SDP}(P)$ are not written in the forms described in Chapter 3. However, it is not hard to see that one can write them in one of the forms. We utilize the above form because it is more suited to the algorithm and its analysis presented in Section 5.4. Before we proceed, let us convince ourselves that the "dual problem" above is indeed the dual to the primal problem.

Let $\Psi : \mathrm{L}(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y} \oplus \mathcal{X}) \to \mathrm{L}(\mathcal{Z} \otimes \mathcal{X} \oplus \mathbb{C}^1)$ be the super-operator whose action is described as

$$\begin{pmatrix} Q & * \\ * & \xi \end{pmatrix} \xrightarrow{\Psi} \begin{pmatrix} \mathrm{Tr}_{\mathcal{Y}}(PQP) - \frac{1}{2}\mathbb{I}_{\mathcal{Z}} \otimes \xi & 0 \\ 0 & \mathrm{Tr}(\xi) \end{pmatrix}.$$

The adjoint super-operator $\Psi^* : \mathrm{L}(\mathcal{Z} \otimes \mathcal{X} \oplus \mathbb{C}^1) \to \mathrm{L}(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y} \oplus \mathcal{X})$ is therefore the super-operator whose action is described as follows:

$$\begin{pmatrix} R & * \\ * & \mu \end{pmatrix} \xrightarrow{\Psi^*} \begin{pmatrix} P(R \otimes \mathbb{I}_{\mathcal{Y}})P & 0 \\ 0 & \mu\mathbb{I}_{\mathcal{X}} - \frac{1}{2}\mathrm{Tr}_{\mathcal{Z}}(R) \end{pmatrix}.$$

Writing the primal objective function in the appropriate form, we observe that the dual constraint is

$$\begin{pmatrix} P\left(R \otimes \mathbb{I}_{\mathcal{Y}}\right) P & 0 \\ 0 & \mu \mathbb{I}_{\mathcal{X}} - \frac{1}{2} \operatorname{Tr}_{\mathcal{Z}}(R) \end{pmatrix} \succeq \begin{pmatrix} \mathbb{I}_{\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}} & 0 \\ 0 & 0 \end{pmatrix}$$

and the dual objective function is

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \in \operatorname{Pos}\left(\mathcal{Z} \otimes \mathcal{X} \oplus \mathbb{C}^1\right).$$

It follows from above that the dual problem is minimizing the scalar quantity μ subject to the constraints

$$P(R \otimes \mathbb{I}_{\mathcal{Y}})P \succeq \mathbb{I}_{\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}} \qquad \text{and} \qquad \frac{1}{2} \operatorname{Tr}_{\mathcal{Z}}(R) \preceq \mu \mathbb{I}_{\mathcal{X}}.$$

This is exactly the dual problem stated above. It is immediate that the for any positive semidefinite operator R, the positive real number μ satisfying $\operatorname{Tr}_{\mathcal{Z}}(R) \preceq \mu \mathbb{I}_{\mathcal{X}}$ is the operator norm of $\operatorname{Tr}_{\mathcal{Z}}(R)$. We also remark that the usage of terms "maximize" and "minimize" is justified for the above primal-dual pair because Slater points exist for both primal and dual problems:

$$(\hat{Q}, \hat{\xi}) = \left(\frac{\mathbb{I}_{\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}}}{4 \dim(\mathcal{X} \otimes \mathcal{Y})}, \frac{\mathbb{I}_{\mathcal{X}}}{\dim(\mathcal{X})} \right) \quad \text{and} \quad (\hat{\mu}, \hat{R}) = \left(\alpha, \alpha \mathbb{I}_{\mathcal{Z} \otimes \mathcal{X}}\right).$$

Since the objective value of the problems are bounded by 1, by Theorem 9, the optimum values of the primal and dual problems are equal and attained.

5.4 NC(poly) simulation of QIP

Given that NC(poly) characterizes PSPACE, it is natural to consider showing that one can approximately solve SDP(P) in NC(poly). It also helps our cause that various matrix operations such as matrix addition, matrix multiplication, matrix inversion, tensor product, and scalar multiplication are known to be computable in NC. In addition to them, computations such as matrix exponentiation, matrix square root, and spectral decomposition can be approximated with high precision within NC.

The assumption that a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ belongs to QIP implies that there exists a single-coin quantum interactive proof system for the problem described in Section‘5.1 with the following completeness and soundness conditions.

1. *Completeness.* For every $x \in A_{\text{yes}}$, there exists a strategy for the prover that causes the verifier to accept with probability 1.

2. *Soundness.*For every $x \in A_{\text{no}}$, regardless of whatever strategy adopted by the prover, the probability that the verifier accepts is at most $1/2 + \varepsilon$ for $\varepsilon = 1/64$.

As noted before, one can assume ε to be exponentially close to 0. However, the choice of ε is sufficient for correct functioning of the algorithm. Recall that, the optimum value of SDP(P) is at least the maximum success probability of the prover and at most $1/\alpha^2$ plus the maximum success probability of the prover. For the choice of $\alpha = 4$, the completeness and soundness conditions impose the following promise on the optimum value of SDP(P).

1. If $x \in A_{\text{yes}}$, then the optimum value of the semidefinite program SDP(P) is at least 1. In other words, there exists a primal feasible solution with objective value 1.
2. If $x \in A_{\text{no}}$, then the optimum value of the semidefinite program SDP(P) is at most $1/2+\varepsilon+1/\alpha^2 < 5/8$. In other words, there exists a dual feasible solution with objective value at most 5/8.

With this promise in mind, we show that a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$in QIP is also contained in NC(poly) as follows. We consider a two-stage computation to distinguish between the two cases as follows.

1. The first stage of the computation computes explicit matrix representations of the projection operators Π_0^0, Π_1^0, Π_0^1, and Π_1^1 from given input string x. This computation can be done exactly in NC(poly).
2. The second stage of the computation runs the algorithm described in the next subsection on the semidefinite program SDP(P) described by the four projection operators mentioned in item 1.

We first demonstrate how the first step can be done in NC(poly). Notice that the verifier's circuit corresponding to both choices of $a \in \{0, 1\}$ is composed of polynomial number of Hadamard, Phase, and Toffoli gates followed by a measurement operation. Following Marriott and Watrous [MW05], the measurement operation for both choices of $a \in \{0, 1\}$ are described by diagonal projection operators. Given that the registers X and Y are m-qubit registers, each of the gates can be written explicitly as a $2^{2m} \times 2^{2m}$ matrix, the entries of each matrix being from the set $\{-1, 0, 1, \iota\}$ modulo the normalization factor of $\sqrt{2}$ for each Hadamard gate. The normalization factor for each Hadamard gate can be accounted for by counting the number of such gates. If r is the number of Hadamard gates, the projection operator Π_b^a for $a, b \in \{0, 1\}$ can be written as

$$\Pi_b^a = \frac{1}{2^r} \left(U_1^a \dots U_q^a\right)^* \Lambda_b^a \left(U_1^a \dots U_q^a\right),$$

where $\{\Lambda_0^a, \Lambda_1^a\}$ are the diagonal projection operators corresponding to the verifier's message $a \in \{0,1\}$ and the matrix entries of each U_j^a is from the set $\{-1, 0, 1, \iota\}$. Computing Π_b^a for each $a, b \in \{0,1\}$ can be done in NC(poly) by first computing each entry of the matrices $U_1^a, \ldots, U_q^a$ and Λ_b^a independently by an NC(poly) circuit and then composing it by an NC circuit that performs matrix multiplication.

The second stage of the computation is to run the algorithm described in the following subsection on SDP(P). The goal of the algorithm is to *accept* if the optimum value of SDP(P) is at least 1 and *reject* if the optimum value of SDP(P) is at most 5/8. For the whole computation to be in NC(poly), this stage of computation has to be implemented in NC, which is described in the remainder of this section.

5.4.1 A parallel semidefinite programming algorithm

This section presents the parallel algorithm for SDP(P). Let

$$\Pi_0^0,\ \Pi_1^0,\ \Pi_0^1,\ \Pi_1^1 \in \mathrm{Pos}\left(\mathcal{X} \otimes \mathcal{Y}\right)$$

be the four projection operators generated by the NC(poly) circuit as described above. The projection operators satisfy

$$\Pi_0^0 + \Pi_1^0 = \Pi_0^1 + \Pi_1^1 = \mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}},$$

for vector spaces $\mathcal{X}$ and $\mathcal{Y}$ both having dimension $M = 2^m$.

For $\alpha = 4$ and P as defined in (5.6), if the optimum of the semidefinite program SDP(P) is at least 1, then the algorithm outputs *accept* and if the optimum is at most 5/8, then the algorithm outputs *reject*. We remark that the algorithm does not try to find an approximation to the optimum value of the semidefinite program. Instead, it either generates a primal feasible solution with objective value greater than 5/8 or a dual feasible solution with objective value strictly less than 1, and accepts or rejects based on the solution generated.

Now we begin describing the desired algorithm for the semidefinite program SDP(P). Let

$$\alpha = 4, \quad \gamma = \frac{4}{3}, \quad \varepsilon = \frac{1}{64}, \quad \delta = \frac{\varepsilon}{\alpha^2}, \quad \text{and} \quad T = \left\lceil \frac{8\gamma \log(M)}{\varepsilon^2 \delta} \right\rceil,$$

and

$$P = \begin{pmatrix} (\Pi_1^0 + \alpha\Pi_0^0) & 0 \\ 0 & (\Pi_1^1 + \alpha\Pi_0^1) \end{pmatrix}, \quad \rho^{(0)} = \frac{1}{2M^2}\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}}, \quad \text{and} \quad \sigma^{(0)} = \frac{1}{M}\mathbb{I}_{\mathcal{X}}.$$

The algorithm is as follows.

1. Compute P, $\rho^{(0)}$, $\sigma^{(0)}$, and T.
2. For every t in $\{0, 1, \ldots, T-1\}$, repeat the following steps:

   (a) Compute

   $$Z^{(t)} \leftarrow \frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)$$

   and compute the projection $\Delta^{(t)}$ onto the negative eigenspace of $Z^{(t)}$. Compute

   $$\beta^{(t)} = \left\langle \Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}, P\rho^{(t)}P \right\rangle.$$

   If $\beta^{(t)} \leq \varepsilon$ then output *accept* and stop.

   (b) Compute

   $$\begin{aligned} X^{(t+1)} &\leftarrow \exp\left(-\sum_{j=0}^{t} \frac{\varepsilon\delta}{\beta^{(j)}} P\left(\Delta^{(j)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right) \\ Y^{(t+1)} &\leftarrow \exp\left(\sum_{j=0}^{t} \frac{\varepsilon\delta}{\beta^{(j)}} \left(\mathrm{Tr}_{\mathcal{Z}}\left(\Delta^{(j)}\right)\right)\right) \end{aligned}$$

   and

   $$\rho^{(t+1)} \leftarrow \frac{X^{(t+1)}}{\mathrm{Tr}\left(X^{(t+1)}\right)} \qquad \text{and} \qquad \sigma^{(t+1)} \leftarrow \frac{Y^{(t+1)}}{\mathrm{Tr}\left(Y^{(t+1)}\right)}$$

3. If the algorithm has not stopped during any iteration of step 2 then output *reject* and stop.

Before presenting the analysis of the algorithm, we briefly discuss the intuition behind the algorithm itself. The algorithm is inspired by the Arora-Kale framework [AK07] for solving semidefinite programs by matrix multiplicative weights update method. A rough sketch of the algorithm is as follows. It is an iterative procedure that tries to construct a suitable primal feasible solution that will demonstrate that the optimum value of the semidefinite program is at least 1. At every step of the iteration, the algorithm checks whether the candidates for primal feasibility are *approximately feasible* using a suitable condition. If the condition is satisfied, then the algorithm halts and *accepts*. Otherwise, the algorithm updates the candidate solutions to obtain better candidate solutions and repeat the process again. If the condition for approximate primal feasibility is not satisfied for a certain number of iterations, the algorithm halts and *rejects*. Each of the three steps of the algorithm described above serves the following purposes.

1. The first step of the algorithm computes the positive semidefinite operator P that specify the objective function and the primal constraints, and various parameters on

which the algorithm operates. It also computes the initial candidate solutions for primal feasibility.

2. The second step of the algorithm is an iterative procedure, which checks if the candidate solutions can be used to obtain a primal feasible solution that guarantees that the optimum value of SDP(P) is at least 1. This is step 2(a) of the algorithm. The next step (step 2(b)) is where the current candidate solutions are updated to new candidate solutions.
3. The third step of the algorithm merely halts and outputs *reject* thereby implying that it has failed to generate a primal feasible solution with the desired objective value. In this case, the algorithm guarantees the existence of a dual feasible solution with objective value strictly less than 1.

In greater detail, the functioning of the algorithm is as follows. We emphasize that the algorithm does not attempt to approximate the optimum value of SDP(P), but merely attempts to determine whether its optimum value is at least or at most $1/\gamma = 3/4$. Given the task at hand, it suffices to show that the optimum value is more or less than $1/\gamma$ by exhibiting a primal or a dual feasible solution with objective value $1/\gamma$, respectively. From the promise on the optimum value of SDP(P), it is fairly straightforward to observe that if there exists a primal feasible solution with objective value $1/\gamma$, then the optimum value is at least 1, and likewise if there exists a dual feasible solution with objective value $1/\gamma$, then the optimum value is at most $5/8$. Note that the algorithm can be run for different choices of $1/\gamma$ to obtain an increasingly accurate approximation of the optimum value. However, for our purposes it is not required to get an accurate approximation of the optimum value of SDP(P).

There is a certain asymmetry in the algorithm. At every iteration, the algorithm *always* tries to generate a primal feasible solution with objective value $1/\gamma$ and checks if the candidate solutions are primal feasible or not. If they are not, the algorithm generates a certificate that guarantees that the candidate solutions are not feasible. If the algorithm cannot generate the desired primal feasible solution for a certain number of iterations, then it uses the certificates to construct a dual feasible solution with objective value close enough to $1/\gamma$. The use of certificates to generate a dual feasible solution is one of the most important aspects of the algorithm.

The variables $\rho^{(t)}$ and $\sigma^{(t)}$ generated in step 2(b) in successive iterations within the algorithm are operators in correspondence with the primal candidate solutions $Q^{(t)}$ and $\xi^{(t)}$, respectively. The initial choices for $\rho^{(0)}$ and $\sigma^{(0)}$ are identity operators of the appropriate dimension, scaled down to have unit trace. They represent the complete lack of information at the start of the algorithm and may not form a primal feasible solution. In this respect, our algorithm can be thought of as an instance of *infeasible start methods* for solving semidefinite programs. At every step, the algorithm makes sure that $\rho^{(t)}$ and $\sigma^{(t)}$ are

density operators. This allows us to implicitly define the candidate solutions for primal feasibility as

$$Q^{(t)} := \frac{\rho^{(t)}}{\gamma} \qquad \text{and} \qquad \xi^{(t)} := \sigma^{(t)}.$$

From the association of $\rho^{(t)}$ and $Q^{(t)}$, it follows that the objective value corresponding to the operator $Q^{(t)}$ is

$$\mathrm{Tr}\left(Q^{(t)}\right) = \frac{3}{4}.$$

Given that the primal candidate solutions $Q^{(t)}$ and $\xi^{(t)}$ always attain the desired objective value, the objective of the algorithm reduces to finding out if they are *close* to a primal feasible solution. It is a simple observation that the candidate solutions are primal feasible if the projection on the negative eigenspace of the operator $Z^{(t)}$, or alternately,

$$\mathbb{I}_{\mathcal{Z}} \otimes \xi^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(PQ^{(t)}P\right)$$

is the zero operator. However, as we see later in the analysis, it suffices to show that the candidate solutions for primal problem yield a setting that is *close* to satisfying the primal constraints above. The notion of closeness is described by the real parameter $\beta^{(t)}$, whose specification is well-suited to the analysis of the algorithm. At a very informal level, a very small value of $\beta^{(t)}$ implies that the weights on the negative eigenspace of $Z^{(t)}$ are very small relative to the current choices of primal candidate solutions $Q^{(t)}$ and $\xi^{(t)}$. It is not difficult to show, as is done in the formal analysis that follows in the next subsection, that if $\rho^{(t)}$ and $\sigma^{(t)}$ provide a setting of $Q^{(t)}$ and $\xi^{(t)}$ for which $\beta^{(t)}$ is small (step 2(a)), then these operators can be appropriately massaged to yield a truly feasible setting without sacrificing the objective value by too much. In such a scenario, the algorithm generates a certificate that the optimum value of SDP(P) is at least 1.

On the other hand, if it so happens that a given choice of $\rho^{(t)}$ and $\sigma^{(t)}$ fails to yield a setting for $Q^{(t)}$ and $\xi^{(t)}$ that is close to being primal feasible, then the algorithm can generate a certificate of this fact. This certificate is unsurprisingly the projection operator on the negative eigenspace of $Z^{(t)}$, which is denoted $\Delta^{(t)}$ in the algorithm. The projection operator serves two distinct purposes in the algorithm as described in the following two paragraphs.

One of the purposes of the operator $\Delta^{(t)}$ is to update the current candidate solution to a new (and possibly better) candidate solution, as described in step 2(b) of the algorithm. This step is where the matrix multiplicative weights update method takes place. The new candidate solutions $\rho^{(t+1)}$ and $\sigma^{(t+1)}$ are generated from the previously computed values of $\Delta^{(j)}$ and $\beta^{(j)}$, for $j = 0, \ldots, t$. Informally speaking, one can think that the effect of the matrix exponentials in these generations is to suppress the weights of $\rho^{(t+1)}$ and boast the weights of $\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)}$ corresponding to the negative eigenspaces of $Z^{(t)}$. Of course, this is only

an informal explanation: technical properties of the matrix exponentials are necessary for a formal demonstration that the algorithm operates correctly. We also note that the two updates in step 2(b) of the algorithm is associated with the loss and gain matrix formalism as describes in Section 5.2.2.

The other purpose of these operators is to implicitly construct a feasible solution R for the dual problem. As noted above, unlike the primal variables Q and ξ, the feasibility of the dual candidate solutions can only be guaranteed after certain number of iterations. It holds that if for a logarithmic number of successive iterations, the algorithm fails to find a setting of $(Q^{(t)}, \xi^{(t)})$ that is close to being primal feasible, a dual-feasible setting for R achieving a dual objective value close to $1/\gamma$ can be found. It is critical for the parallel simulation of the algorithm that the number of iterations T is logarithmic in the size of the operators.

5.4.2 Analysis of the algorithm (assuming exact computations)

Before we proceed with the analysis of the algorithm, we require the following observations. For any operator $A \in \mathrm{L}(\mathcal{X})$, the exponential of A is defined as

$$\exp(A) = \mathbb{I}_{\mathcal{X}} + \sum_{i=1}^{\infty} \frac{A^i}{i!}$$

Also, for any two operators $A, B \in \mathrm{Herm}(\mathcal{X})$, the *Golden-Thompson inequality* states the following:

$$\mathrm{Tr}\left[\exp(A+B)\right] \leq \mathrm{Tr}\left[\exp(A)\exp(B)\right].$$

A proof of the inequality can be found in section IX.3 of Bhatia [Bha97]. We also require the following two simple lemmas.

Lemma 14. *Let $R \in \mathrm{Pos}(\mathcal{X})$ be an operator satisfying $R \preceq \mathbb{I}_{\mathcal{X}}$. Then for every real number η, it holds that*

$$\exp(\eta R) \preceq \mathbb{I}_{\mathcal{X}} + \eta \exp(\eta) R.$$

Proof. It is sufficient to prove the inequality for a scalar $\lambda \in [0,1]$, for then the operator inequality follows by considering a spectral decomposition of it and the commutativity of the operator and its exponential. If $\lambda = 0$ the inequality is immediate, so assume $0 < \lambda \leq 1$.

Consider the function $f(\lambda) = \exp(\eta\lambda)$. By the Mean Value Theorem, there exists a value $\lambda_0 \in (0, \lambda)$ such that

$$\frac{f(\lambda) - f(0)}{\lambda} = f'(\lambda_0) \Longrightarrow \frac{\exp(\eta\lambda) - 1}{\lambda} = \eta \exp(\eta\lambda_0) \leq \eta \exp(\eta),$$

from which the inequality follows.

□

Lemma 15. *For any positive semidefinite operator $R \in \mathrm{Pos}\left(\mathcal{Z} \otimes \mathcal{X}\right)$ such that $\mathcal{Z} = \mathbb{C}^2$. Then*

$$R \preceq 2\mathbb{I}_{\mathcal{Z}} \otimes \mathrm{Tr}_{\mathcal{Z}}(R).$$

Proof. Let

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \text{and} \quad Y = \begin{pmatrix} 0 & -\iota \\ \iota & 0 \end{pmatrix}$$

be the two-dimensional Pauli operators. It holds for any $R \in \mathrm{L}\left(\mathcal{Z} \otimes \mathcal{X}\right)$

$$\mathbb{I}_{\mathcal{Z}} \otimes \mathrm{Tr}_{\mathcal{Z}}(R) = \frac{1}{2}\left[R + (X \otimes \mathbb{I}_{\mathcal{X}})R(X \otimes \mathbb{I}_{\mathcal{X}}) + (Y \otimes \mathbb{I}_{\mathcal{X}})R(Y \otimes \mathbb{I}_{\mathcal{X}}) + (Z \otimes \mathbb{I}_{\mathcal{X}})R(Z \otimes \mathbb{I}_{\mathcal{X}})\right]$$

and hence $R \preceq 2\mathbb{I}_{\mathcal{Z}} \otimes \mathrm{Tr}_{\mathcal{Z}}(R)$.

□

We now show that the algorithm described at the beginning of the section performs as desired. Recall that, if $x \in A_{\mathrm{yes}}$ then the optimum value of $\mathrm{SDP}(P)$ is at least 1 and if $x \in A_{\mathrm{no}}$ then the optimum value of $\mathrm{SDP}(P)$ is at most $5/8$. In what follows below, we precisely establish the two facts. The analysis presented below is done under the assumption that every step of the algorithm can be performed exactly. However, some of the computations performed by the algorithm cannot be done exactly and this is formally discussed in the following subsection.

The case when algorithm accepts: Assume first that the algorithm accepts. Let the algorithm accepts at any iteration t. We write ρ, σ, Δ, and β instead of $\rho^{(t)}$, $\sigma^{(t)}$, $\Delta^{(t)}$, and $\beta^{(t)}$ for the sake of clarity. Since Δ is the projection onto the negative eigenspace of $\gamma\mathbb{I}_{\mathcal{Z}} \otimes \sigma - \mathrm{Tr}_{\mathcal{Y}}(P\rho P)$, we have that

$$-\Delta\,\mathrm{Tr}_{\mathcal{Y}}(P\rho P)\Delta \preceq \Delta\left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma - \mathrm{Tr}_{\mathcal{Y}}(P\rho P)\right)\Delta \preceq \frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma - \mathrm{Tr}_{\mathcal{Y}}(P\rho P).$$

We conclude that

$$\mathrm{Tr}_{\mathcal{Y}}(P\rho P) \preceq \frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma + \Delta\,\mathrm{Tr}_{\mathcal{Y}}(P\rho P)\Delta$$

and therefore using Lemma 15,

$$\mathrm{Tr}_{\mathcal{Y}}(P\rho P) \preceq \mathbb{I}_{\mathcal{Z}} \otimes \left[\frac{\gamma}{2}\sigma + 2\ \left(\mathrm{Tr}_{\mathcal{Z}}\left(\Delta\,\mathrm{Tr}_{\mathcal{Y}}(P\rho P)\Delta\right)\right)\right].$$

Given that

$$\beta = \langle \Delta \otimes \mathbb{I}_{\mathcal{Y}}, P\rho P \rangle = \langle \Delta, \mathrm{Tr}_{\mathcal{Y}}(P\rho P) \rangle$$

it follows that

$$\mathrm{Tr}_{\mathcal{Y}} \left[P \left(\frac{\rho}{\gamma + 4\beta} \right) P \right] \preceq \frac{1}{2} \mathbb{I}_{\mathcal{Z}} \otimes \left(\frac{\gamma\sigma + 4 \ (\mathrm{Tr}_{\mathcal{Z}} \left(\Delta \, \mathrm{Tr}_{\mathcal{Y}}(P\rho P) \Delta \right))}{\gamma + 4\beta} \right). \tag{5.7}$$

Defining $Q \in \mathrm{Pos}\left(\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}\right)$ and $\xi \in \mathrm{D}\left(\mathcal{X}\right)$ as

$$Q = \frac{\rho}{\gamma + 4\beta} \qquad \text{and} \qquad \xi = \frac{\gamma\sigma + 4 \ \mathrm{Tr}_{\mathcal{Z}} \left(\Delta \, \mathrm{Tr}_{\mathcal{Y}}(P\rho P) \Delta \right)}{\gamma + 4\beta}$$

we observe that (5.7) can be alternately written as $\mathrm{Tr}_{\mathcal{Y}}\left(PQP\right) \preceq \mathbb{I}_{\mathcal{Z}} \otimes \xi$, which implies that (Q, ξ) form a primal feasible solution. Given that the algorithm has accepted, it holds that $\beta \leq \varepsilon$, and therefore the primal objective value obtained is

$$\mathrm{Tr}\left(Q\right) = \frac{1}{\gamma + 4\beta} \geq \frac{1}{\gamma + 4\varepsilon} \geq \frac{3}{4}\left(1 - 3\varepsilon\right) > \frac{5}{8}.$$

The case when algorithm rejects: Assume that the algorithm rejects. In this case, $\beta^{(t)} > \varepsilon$ for every choice of t. Let

$$R := \frac{1 + 4\varepsilon}{T} \sum_{t=0}^{T-1} \frac{\Delta^{(t)}}{\beta^{(t)}}. \tag{5.8}$$

Below we show that $\mu = 7/8$ and R form a dual feasible solution. This is done by establishing the following two operator inequalities:

$$P(R \otimes \mathbb{I}_{\mathcal{Y}})P \succeq \mathbb{I}_{\mathcal{Z} \otimes \mathcal{X} \otimes \mathcal{Y}} \qquad \text{and} \qquad \mathrm{Tr}_{\mathcal{Z}}(R)/2 \preceq \mu \mathbb{I}_{\mathcal{X}}.$$

As we will see shortly, the first operator inequality follows from the *loss matrix formalism* and the second operator inequality follows from the *gain matrix formalism* discussed in Section 5.2.2. Note that the update that generates the operators $X^{(0)}, \ldots, X^{(T-1)}$ resemble the update in the loss matrix formalism with the associated loss matrix at iteration t dfined as

$$C^{(t)} = \frac{\delta}{\beta^{(t)}} \left(P \left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}} \right) P \right). \tag{5.9}$$

Since the algorithm rejects, it is easy to observe that

$$\left\| C^{(t)} \right\|_{\infty} = \left\| \frac{\delta}{\beta^{(t)}} P \left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}} \right) P \right\|_{\infty} < \frac{\delta \left\| P^2 \right\|_{\infty}}{\varepsilon} < \frac{\delta \alpha^2}{\varepsilon} = 1$$

for every iteration $t \in \{0, \ldots, T-1\}$, and consequently it satisfies the assumption in the matrix multiplicative weights update method. With this association in mind, let us proceed with the first part of the analysis. Observe that for every choice of $t = 0, \ldots, T-1$, we have

$$\begin{aligned}
\operatorname{Tr}\left(X^{(t+1)}\right) &= \operatorname{Tr}\left[\exp\left(-\sum_{j=0}^{t}\frac{\varepsilon\delta}{\beta^{(j)}}P\left(\Delta^{(j)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right] \\
&\leq \operatorname{Tr}\left[\exp\left(-\sum_{j=0}^{t-1}\frac{\varepsilon\delta}{\beta^{(j)}}P\left(\Delta^{(j)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\exp\left(-\frac{\varepsilon\delta}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right] \\
&= \operatorname{Tr}\left[X^{(t)}\exp\left(-\frac{\varepsilon\delta}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right]
\end{aligned}$$

where the inequality follows from Golden-Thompson inequality. Lemma 14 implies that

$$\exp\left(-\frac{\varepsilon\delta}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right) \preceq \mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}} - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P$$

and consequently

$$\begin{aligned}
\operatorname{Tr}\left(X^{(t+1)}\right) &\leq \operatorname{Tr}\left(X^{(t)}\left(\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}} - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right) \\
&= \operatorname{Tr}\left(X^{(t)}\right) - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}\left\langle X^{(t)}, P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right\rangle \\
&= \left[\operatorname{Tr}\left(X^{(t)}\right)\right]\left(1 - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}\left\langle \rho^{(t)}, P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right\rangle\right).
\end{aligned}$$

Since

$$\beta^{(t)} = \left\langle \rho^{(t)}, P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right\rangle,$$

it follows from above and the inequality $1 - z \leq \exp(-z)$ for all $z \in \mathbb{R}$ that

$$\operatorname{Tr}\left(X^{(t+1)}\right) \leq \left[\operatorname{Tr}\left(X^{(t)}\right)\right]\exp\left(-\varepsilon\delta\exp(-\varepsilon)\right).$$

Applying the above inequality recursively yields

$$\operatorname{Tr}\left(X^{(T)}\right) \leq \left[\operatorname{Tr}\left(X^{(0)}\right)\right]\exp\left(-T\varepsilon\delta\exp(-\varepsilon)\right) = 2M^2\exp\left(-T\varepsilon\delta\exp(-\varepsilon)\right).$$

Now we turn to lower bound the quantity $\operatorname{Tr}\left(X^{(T)}\right)$, which in turn will help us establish that R is dual feasible. For any Hermitian operator $H \in \operatorname{Herm}\left(\mathbb{C}^k\right)$, it holds that

$$\operatorname{Tr}\left(\exp(H)\right) \geq \exp\left(\lambda_k(H)\right)$$

and therefore

$$\begin{aligned}\mathrm{Tr}\left(X^{(T)}\right) &= \mathrm{Tr}\left(\exp\left(-\sum_{t=0}^{T-1}\frac{\varepsilon\delta}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right)\\ &\geq \exp\left(-\varepsilon\delta\lambda_{2M^2}\left(\sum_{t=0}^{T-1}\frac{1}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right).\end{aligned}$$

Combining the lower bound and the upper bound on $\mathrm{Tr}\left(X^{(T)}\right)$ obtained above, we observe that

$$\begin{aligned}\exp\left(-\varepsilon\delta\lambda_{2M^2}\left(\sum_{t=0}^{T-1}\frac{1}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\right) &\leq \mathrm{Tr}\left(X^{(T)}\right)\\ &\leq 2M^2\exp(-T\varepsilon\delta\exp(-\varepsilon)).\end{aligned}$$

The above inequality yields the following

$$\lambda_{2M^2}\left(\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{\beta^{(t)}}P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right)\geq\exp(-\varepsilon)-\frac{\log(2M^2)}{T\varepsilon\delta}. \tag{5.10}$$

The above inequality does not resemble (5.2), however, we note that for our choice of loss matrix (refer to (5.9)) it holds that the expected loss at every iteration t is δ. Substituting this equality in (5.2) and rearranging term gives us (5.9). To show that R as defined in (5.8) is dual feasible, we substitute the value of T in (5.10). We obtain that

$$\lambda_{2M^2}(P(R\otimes\mathbb{I}_{\mathcal{Y}})P)\geq(1+4\varepsilon)\left(\exp(-\varepsilon)-\frac{\log(2M^2)}{T\varepsilon\delta}\right)\geq 1.$$

It follows from above that

$$P(R\otimes\mathbb{I}_{\mathcal{Y}})P\geq\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}}$$

implying that R forms a dual feasible solution of SDP(P). It remains to prove that

$$\mathrm{Tr}_{\mathcal{Z}}(R)/2\preceq\mu\mathbb{I}_{\mathcal{X}}$$

for some choice of μ. The analysis of this part is very similar to the analysis described above except that we use the gain matrix formalism instead of the loss matrix formalism below. The update that generates the operators $Y^{(0)},\ldots,Y^{(T-1)}$ resemble the update in the gain matrix formalism with the associated gain matrix at iteration t defined as

$$C^{(t)}:=\frac{\delta}{\beta^{(t)}}\left(\mathrm{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right)\right). \tag{5.11}$$

For the above choice of gain matrix, we observe that

$$\left\|C^{(t)}\right\|_\infty = \left\|\frac{\delta}{\beta^{(t)}}\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right)\right\|_\infty < \frac{2\delta}{\varepsilon} = \frac{2}{\alpha^2} < 1$$

for every choice of $t \in \{0,\ldots,T-1\}$ and hence the assumption of matrix multiplicative weights update method is satisfied. By applying Golden-Thompson inequality again, we have for every choice of $t = 0,\ldots,T-1$

$$\begin{aligned}
\operatorname{Tr}\left(Y^{(t+1)}\right) &= \operatorname{Tr}\left[\exp\left(\varepsilon\delta\sum_{j=0}^{t}\frac{\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(j)}\right)}{\beta^{(j)}}\right)\right] \\
&\leq \operatorname{Tr}\left[\exp\left(\varepsilon\delta\sum_{j=0}^{t-1}\frac{\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(j)}\right)}{\beta^{(j)}}\right)\exp\left(\varepsilon\delta\frac{\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right)}{\beta^{(t)}}\right)\right] \\
&= \operatorname{Tr}\left[Y^{(t)}\exp\left(\varepsilon\delta\frac{\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right)}{\beta^{(t)}}\right)\right]
\end{aligned}$$

Using Lemma 14 once again, we obtain that

$$\exp\left(\frac{\varepsilon\delta}{\beta^{(t)}}\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right)\right) \preceq \mathbb{I}_{\mathcal{X}} + \frac{\varepsilon\delta\exp(\varepsilon)}{\beta^{(t)}}\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right).$$

It follows that

$$\operatorname{Tr}\left(Y^{(t+1)}\right) \leq \left[\operatorname{Tr}\left(Y^{(t)}\right)\right]\left(1 + \frac{\varepsilon\delta\exp(\varepsilon)}{\beta^{(t)}}\left\langle\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right),\sigma^{(t)}\right\rangle\right).$$

Since $\Delta^{(t)}$ is projection onto the negative eigenspace of $\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}}\otimes\sigma^{(t)} - \operatorname{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)$, we have the following positive semidefinite inequality

$$\Delta^{(t)}\left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}}\otimes\sigma^{(t)} - \operatorname{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\right)\Delta^{(t)} \preceq 0$$

and consequently

$$\left\langle\operatorname{Tr}_{\mathcal{Z}}\left(\Delta^{(t)}\right),\sigma^{(t)}\right\rangle = \left\langle\Delta^{(t)},\mathbb{I}_{\mathcal{Z}}\otimes\sigma^{(t)}\right\rangle \leq \frac{2}{\gamma}\left\langle\Delta^{(t)},\operatorname{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\right\rangle = \frac{2\beta^{(t)}}{\gamma}.$$

Therefore,

$$\operatorname{Tr}\left(Y^{(t+1)}\right) \leq \left[\operatorname{Tr}\left(Y^{(t)}\right)\right]\left(1 + \frac{2\varepsilon\delta\exp(\varepsilon)}{\gamma}\right) \leq \left[\operatorname{Tr}\left(Y^{(t)}\right)\right]\exp\left(\frac{2\varepsilon\delta\exp(\varepsilon)}{\gamma}\right).$$

By applying the above inequality recursively and observing that $Y^{(0)} = \mathbb{I}_{\mathcal{Y}}$, we immediately obtain that

$$\mathrm{Tr}\left(Y^{(T)}\right) \leq M \exp\left(\frac{2T\varepsilon\delta\exp(\varepsilon)}{\gamma}\right).$$

On the other hand, we have

$$\mathrm{Tr}\left(Y^{(T)}\right) = \mathrm{Tr}\left[\exp\left(\varepsilon\delta\,\mathrm{Tr}_{\mathcal{Z}}\left(\sum_{t=0}^{T-1}\frac{\Delta^{(t)}}{\beta^{(t)}}\right)\right)\right] \geq \exp\left[\varepsilon\delta\lambda_1\left(\mathrm{Tr}_{\mathcal{Z}}\left(\sum_{t=0}^{T-1}\frac{\Delta^{(t)}}{\beta^{(t)}}\right)\right)\right].$$

The lower and upper bounds on $\mathrm{Tr}\left(Y^{(T)}\right)$ yield the following inequality:

$$\lambda_1\left(\mathrm{Tr}_{\mathcal{Z}}\left(\sum_{t=0}^{T-1}\frac{\Delta^{(t)}}{\beta^{(t)}}\right)\right) \leq \frac{2T\exp(\varepsilon)}{\gamma} + \frac{\log(M)}{\varepsilon\delta}. \tag{5.12}$$

The above inequality does not resemble (5.3), however, we note that for our choice of loss matrix (refer to (5.11)) it holds that the expected loss at every iteration t is at most $2\delta/\gamma$. Substituting this equality in (5.3) and rearranging term gives us (5.12).

Define the scalar quantity μ as

$$\mu := \frac{1+8\varepsilon}{\gamma}.$$

Using (5.12), we obtain

$$\begin{aligned}\lambda_1\left(\mathrm{Tr}_{\mathcal{Z}}(R)\right) &= \left(\frac{1+4\varepsilon}{T}\right)\lambda_1\left(\mathrm{Tr}_{\mathcal{Z}}\left(\sum_{t=0}^{T-1}\frac{\Delta^{(t)}}{\beta^{(t)}}\right)\right)\\ &\leq (1+4\varepsilon)\left(\frac{2\exp(\varepsilon)}{\gamma} + \frac{\log(M)}{T\varepsilon\delta}\right) \leq \frac{2(1+8\varepsilon)}{\gamma} = 2\mu.\end{aligned}$$

It follows that $\frac{1}{2}\mathrm{Tr}_{\mathcal{Z}}(R) \preceq \mu\mathbb{I}_{\mathcal{X}}$. Substituting the value of ε and γ, we observe that $\mu < 7/8$. This completes the construction of an appropriate dual feasible solution under the assumption that the algorithm rejects.

To summarize, we have shown that if the optimum value of the semidefinite program is at least 1, then the algorithm constructs (Q, ξ) that form a primal feasible solution with objective value strictly greater than 5/8 and accepts. Similarly, if the optimum value is smaller than 5/8, then the algorithm constructs (μ, R) that form a dual feasible solution with objective value less than 7/8 and rejects. It is clear that the algorithm operates as desired.

5.4.3 Precision issues

In the previous subsection, we presented the analysis assuming that every computation done by the algorithm is exact. However, some of the matrix computations required in the algorithm cannot be done exactly in NC or for that matter any discrete model of computation. Nevertheless, we briefly sketch below how the analysis in the previous subsection can be appropriately modified to accommodate for the approximate computations, provided we make certain assumptions about the quality of the approximation. While simple matrix computations such as matrix addition, matrix multiplication, tensor product, and matrix inverse can be done exactly, other computations such as singular value decomposition, square root of a positive semidefinite matrix, and matrix exponentiation cannot be computed with zero precision error because of the unavoidable fact that the output of such computations may involve irrational numbers.

The algorithm presented in the previous section refers to two such computations that has to be approximated: computation of the negative eigenspace projections in step 2(a) and matrix exponentiations in step 2(b). Assuming that the approximate computations can be performed with sufficient accuracy (refer to the subsection following this one), the analysis presented in the previous subsection can be used without any major complications to show that the algorithm still operates as desired.

Keeping the objective of the algorithm in mind, we make the following assumptions on the approximation of computation of negative eigenspace projection and matrix exponentiation. We remark that the one can even get a much better approximation than what we assume below. Let $\eta > 0$ be the accuracy parameter of the approximate computations. Assume that

$$2\eta < \varepsilon.$$

For every choice of $t \in \{0, \ldots, T-1\}$, let $\Delta^{(t)}$ be a positive semidefinite operator satisfying

$$\left\| \sqrt{\Delta^{(t)}} - \Lambda^{(t)} \right\|_\infty < \frac{\eta^2}{50M} \tag{5.13}$$

for $\Lambda^{(t)}$ being the true projection operator onto the negative eigenspace of $Z^{(t)}$ as defined in step 2(a) of the algorithm. Since $\Lambda^{(t)}$ is a projection operator, it holds that

$$\left\| \sqrt{\Delta^{(t)}} \right\|_\infty < 1 + \frac{\eta^2}{50M}.$$

Here $\sqrt{\Delta^{(t)}}$ is the positive semidefinite operator that will be computed by the algorithm as an approximation to the true projection $\Lambda^{(t)}$. That the operator $\Lambda^{(t)}$ can be approximated to the accuracy described in (5.13) will be justified in the next subsection.

For every choice of $t \in \{0, \ldots, T-1\}$, let $\rho^{(t)}$ and $\sigma^{(t)}$ be positive semidefinite operators

that satisfy

$$\left\| \rho^{(t)} - \frac{X^{(t)}}{\mathrm{Tr}\left(X^{(t)}\right)} \right\|_\infty < \frac{\eta^2}{50M^2} \qquad \text{and} \qquad \left\| \sigma^{(t)} - \frac{Y^{(t)}}{\mathrm{Tr}\left(Y^{(t)}\right)} \right\|_\infty < \frac{\eta^2}{10M} \tag{5.14}$$

where $X^{(t)}$ and $Y^{(t)}$ are defined in step 2(b) of the algorithm. In addition to above, we assume that both $\rho^{(t)}$ and $\sigma^{(t)}$ are density operators. We do not lose any generality by assuming it. That these operators can be approximated to the accuracy described in (5.14) will be justified in the next subsection.

Given that the negative eigenspace projection and matrix exponentiation can be approximated within the desired accuracy for any iteration t, the closeness parameter $\beta^{(t)}$, defined in step 2(a) of the algorithm, can be computed exactly. More specifically, we define $\beta^{(t)}$ not in terms of the true projection operator $\Lambda^{(t)}$; instead we define it in terms of $\Delta^{(t)}$, the approximation to the true projection operator as follows:

$$\beta^{(t)} = \left\langle \Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}, P\rho^{(t)}P \right\rangle. \tag{5.15}$$

Keeping in mind the assumptions made on the quality of approximation, we proceed with the brief sketch of the modified analysis of the algorithm.

The case when algorithm accepts: Assume that the algorithm accepts at step t. To begin, we note that for any three Hermitian operators $A, B, C \in \mathrm{Herm}\left(\mathcal{W}\right)$, the following norm inequality holds:

$$\left\| CAC - BAB \right\|_\infty \leq \left\| B \right\|_\infty + \left\| C \right\|_\infty \cdot \left\| A \right\|_\infty \cdot \left\| B - C \right\|_\infty.$$

Letting $A = \frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)$, $B = \sqrt{\Delta^{(t)}}$,and $C = \Lambda^{(t)}$, we have from (5.13),

$$\left\| B \right\|_\infty + \left\| C \right\|_\infty \leq 2 + \frac{\eta^2}{50M} \quad \text{and} \quad \left\| B - C \right\|_\infty \leq \frac{\eta^2}{50M} \quad \text{and} \quad \left\| A \right\|_\infty \leq \gamma + \alpha^2 < 20.$$

This implies that the quantity

$$\left\| \Lambda^{(t)} \left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right) \right) \Lambda^{(t)} - \sqrt{\Delta^{(t)}} \left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}(P\rho^{(t)}P) \right) \sqrt{\Delta^{(t)}} \right\|_\infty$$

is strictly smaller than η^2/M and therefore

$$\begin{aligned} &\sqrt{\Delta^{(t)}} \left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right) \right) \sqrt{\Delta^{(t)}} \\ &\qquad \preceq \Lambda^{(t)} \left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}(P\rho^{(t)}P) \right) \Lambda^{(t)} + \frac{\eta^2}{M}\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}}. \end{aligned} \tag{5.16}$$

It follows from above that

$$-\sqrt{\Delta^{(t)}}\left(\mathrm{Tr}_{\mathcal{Y}}(P\rho^{(t)}P)\right)\sqrt{\Delta^{(t)}} \preceq \frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}}\otimes\sigma - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right) + \frac{\eta^2}{M}\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}}.$$

The above inequality follows because $\Lambda^{(t)}$ is the true projection operator onto the negative eigenspace of

$$\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}}\otimes\sigma - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right).$$

Therefore, using Lemma 15 on the operator $\sqrt{\Delta^{(t)}}\left(\mathrm{Tr}_{\mathcal{Y}}(P\rho^{(t)}P)\right)\sqrt{\Delta^{(t)}} \in \mathrm{Pos}\left(\mathcal{Z}\otimes\mathcal{X}\right)$, we have that

$$\mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right) \preceq \frac{1}{2}\mathbb{I}_{\mathcal{Z}}\otimes\left(\gamma\sigma^{(t)} + 4\,\mathrm{Tr}_{\mathcal{Z}}\left(\sqrt{\Delta^{(t)}}\,\mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\sqrt{\Delta^{(t)}}\right) + \frac{2\eta^2}{M}\mathbb{I}_{\mathcal{X}}\right).$$

Using the fact that

$$\beta^{(t)} = \left\langle\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}, P\rho^{(t)}P\right\rangle = \left\langle\Delta^{(t)}, \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\right\rangle = \mathrm{Tr}\left[\mathrm{Tr}_{\mathcal{Z}}\sqrt{\Delta^{(t)}}\left(\mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\sqrt{\Delta^{(t)}}\right)\right]$$

we obtain that

$$Q = \frac{\rho^{(t)}}{\gamma + 4\beta^{(t)} + 2\eta^2}$$

and

$$\xi = \frac{\gamma\sigma^{(t)} + 4\,\mathrm{Tr}_{\mathcal{Z}}\sqrt{\Delta^{(t)}}\left(\mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)\sqrt{\Delta^{(t)}}\right) + \frac{2\eta^2}{M}\mathbb{I}_{\mathcal{X}}}{\gamma + 4\beta^{(t)} + 2\eta^2}$$

satisfy all the primal constraints of SDP(P). That is, $\mathrm{Tr}_{\mathcal{Y}}(PQP) \preceq \frac{1}{2}\mathbb{I}_{\mathcal{Z}}\otimes\xi$ for $\xi\in\mathrm{D}\left(\mathcal{X}\right)$ and $Q\in\mathrm{Pos}\left(\mathcal{Z}\otimes\mathcal{X}\otimes\mathcal{Y}\right)$. Given that the algorithm accepts, it holds that $\beta^{(t)}\leq\varepsilon$ and hence the objective value of the primal solution is at least

$$\frac{1}{\gamma + 2\varepsilon + \eta^2} > \frac{1}{\gamma + 4\varepsilon + \frac{\varepsilon^2}{2}} > \frac{5}{8}.$$

Therefore, under the assumption that (5.13) holds, the algorithm works as desired in the accepting case.

The case when algorithm rejects: Assume that the algorithm rejects. Going through the analysis as presented in the previous subsection, we have by Golden Thompson inequality,

$$\mathrm{Tr}\left(X^{(t+1)}\right) \leq \mathrm{Tr}\left(X^{(t)}\right) - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}\left\langle X^{(t)}, P\left(\Delta^{(t)}\otimes\mathbb{I}_{\mathcal{Y}}\right)P\right\rangle.$$

At this point, we substitute the approximation $\rho^{(t)}$ stored by the algorithm for $\frac{X^{(t)}}{\mathrm{Tr}\left(X^{(t)}\right)}$ in the above equation using the first norm inequality in (5.14). We observe that

$$\begin{aligned}\left|\left\langle \rho^{(t)}, P\left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right\rangle - \frac{1}{\mathrm{Tr}\left(X^{(t)}\right)}\left\langle X^{(t)}, P\left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right\rangle\right| &< \frac{\eta^2 \left\| P\left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right\|_{\infty}}{50} \\ &< \eta^2.\end{aligned}$$

As a consequence of the above inequality, we have

$$\mathrm{Tr}\left(X^{(t+1)}\right) \leq \left[\mathrm{Tr}\left(X^{(t)}\right)\right]\left(1 - \frac{\varepsilon\delta\exp(-\varepsilon)}{\beta^{(t)}}\left(\left\langle \rho^{(t)}, P\left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right\rangle - \eta^2\right)\right).$$

Since the algorithm rejects, it holds that $\varepsilon < \beta^{(t)}$ for every choice of $t = 0, \ldots, T-1$. By our assumption on η, we have the following chain of inequalities: $2\eta < \varepsilon < \beta^{(t)}$, and hence $2\eta^2 < \varepsilon\beta^{(t)}$. Moreover, from (5.15) we have that $\beta^{(t)} = \left\langle \rho^{(t)}, P\left(\Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}\right) P\right\rangle$. Using both the observations in the previous equation, we obtain

$$\begin{aligned}\mathrm{Tr}\left(X^{(t+1)}\right) &\leq \left[\mathrm{Tr}\left(X^{(t)}\right)\right]\left(1 - \varepsilon\delta\exp(-\varepsilon)\left(1 - \frac{\varepsilon}{2}\right)\right) \\ &\leq \left[\mathrm{Tr}\left(X^{(t)}\right)\right]\exp\left(-\varepsilon\delta\left(1 - \frac{\varepsilon}{2}\right)\exp(-\varepsilon)\right).\end{aligned}$$

Applying the inequality recursively, we obtain

$$\mathrm{Tr}\left(X^{(T)}\right) \leq 2M^2\exp\left(-T\varepsilon\delta\left(1 - \frac{\varepsilon}{2}\right)\exp(-\varepsilon)\right).$$

Going through the analysis as before, we have that

$$\lambda_{M^2}(P(R \otimes \mathbb{I}_{\mathcal{Y}})P) \geq (1 + 4\varepsilon)\left(\left(1 - \frac{\varepsilon}{2}\right)\exp(-\varepsilon) - \frac{\log(2M^2)}{T\varepsilon\delta}\right) \geq 1.$$

We now proceed to show that the scalar quantity μ is large enough even after incorporating the precision issues. Along similar lines, we have

$$\mathrm{Tr}\left(Y^{(t+1)}\right) \leq \left(\mathrm{Tr}\left(Y^{(t)}\right) + \frac{\varepsilon\delta\exp(\varepsilon)}{\beta^{(t)}}\left\langle \Delta^{(t)}, Y^{(t)}\right\rangle\right)$$

At this point, we substitute the approximation $\sigma^{(t)}$ stored by the algorithm for $Y^{(t)}$ in the above equation by using the second norm inequality in (5.14). As before, it immediately follows that

$$\left|\left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle - \frac{1}{\mathrm{Tr}\left(Y^{(t)}\right)}\left\langle \Delta^{(t)}, Y^{(t)}\right\rangle\right| < \frac{\eta^2 \left\|\Delta^{(t)}\right\|_{\infty}}{10} < \eta^2.$$

It follows from above that

$$\frac{1}{\mathrm{Tr}\left(Y^{(t)}\right)}\left\langle \Delta^{(t)}, Y^{(t)}\right\rangle < \left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle + \eta^2 < \left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle + \frac{\varepsilon^2}{4}$$

and hence

$$\mathrm{Tr}\left(Y^{(t+1)}\right) \leq \left[\mathrm{Tr}\left(Y^{(t)}\right)\right]\left(1 + \frac{\varepsilon\delta\exp(\varepsilon)}{\beta^{(t)}}\left(\left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle + \frac{\varepsilon^2}{4}\right)\right).$$

Recall that $\Lambda^{(t)}$ is the true projection on the negative eigenspace of

$$\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}\left(P\rho^{(t)}P\right)$$

and consequently the operator inequality holds: $\Lambda^{(t)}\left(\frac{\gamma}{2}\mathbb{I}_{\mathcal{Z}} \otimes \sigma^{(t)} - \mathrm{Tr}_{\mathcal{Y}}(P\rho^{(t)}P)\right)\Lambda^{(t)} \preceq 0$. From (5.16) it follows that

$$\frac{\gamma}{2}\left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle \leq \left\langle \Delta^{(t)} \otimes \mathbb{I}_{\mathcal{Y}}, P\rho^{(t)}P\right\rangle + \frac{\eta^2}{M}\mathrm{Tr}\left(\mathbb{I}_{\mathcal{Z}\otimes\mathcal{X}}\right)$$

and hence

$$\left\langle \Delta^{(t)}, \sigma^{(t)}\right\rangle < \frac{2\left(\beta^{(t)} + 2\eta^2\right)}{\gamma} < \frac{2\beta^{(t)} + \varepsilon^2}{\gamma}.$$

Proceeding through the analysis presented in the previous subsection and simplifying, we obtain

$$\begin{aligned}\mathrm{Tr}\left(Y^{(t+1)}\right) &< \left[\mathrm{Tr}\left(Y^{(t)}\right)\right]\left(1 + \frac{\varepsilon\delta\exp(\varepsilon)}{\beta^{(t)}}\left(\frac{2\beta^{(t)} + \varepsilon^2}{\gamma} + \frac{\varepsilon^2}{4}\right)\right) \\ &\leq \left[\mathrm{Tr}\left(Y^{(t)}\right)\right]\exp\left(\frac{2\varepsilon\delta\left(1 + \frac{2\varepsilon}{3}\right)\exp(\varepsilon)}{\gamma}\right).\end{aligned}$$

Using the above inequality recursively yields

$$\mathrm{Tr}\left(Y^{(T)}\right) \leq M\exp\left(\frac{2T\varepsilon\delta\left(1 + \frac{2\varepsilon}{3}\right)\exp(\varepsilon)}{\gamma}\right).$$

Following the analysis of the previous subsection, we obtain that the largest eigenvalue of the operator $\mathrm{Tr}_{\mathcal{Z}}(R)$ is still at most 2μ as shown below.

$$\lambda_1\left(\mathrm{Tr}_{\mathcal{Z}}(R)\right) < (1 + 4\varepsilon)\left(\frac{2\left(1 + \frac{2\varepsilon}{3}\right)\exp(\varepsilon)}{\gamma} + \frac{\log(M)}{T\varepsilon\delta}\right) < \frac{2(1 + 8\varepsilon)}{\gamma} = 2\mu.$$

Hence as before

$$(\mu, R) = \left(\frac{1+8\varepsilon}{\gamma}, \frac{1+4\varepsilon}{T} \sum_{t=0}^{T-1} \frac{\Delta^{(t)}}{\beta^{(t)}} \right)$$

still form a dual feasible solution with dual objective value strictly less than 1. Therefore, under the accuracy assumptions of the approximations , the algorithm works as desired in the rejecting case too.

5.4.4 An NC implementation of the algorithm

The final part of the proof of the containment QIP $\subseteq$ PSPACE is to show that the algorithm presented earlier can be implemented in NC. We recall that NC is the class of all functions (including predicates that represent promise problems) computable by logarithmic-space uniform family of Boolean circuits of polylogarithmic depth.

Every entry of a matrix or the scalar quantities generated or used by the algorithm will have rational real and imaginary parts, which can be denoted $(a/b) + \iota(c/d)$ and stored as a quadruple (a, b, c, d) in binary notation. The *length* of any complex number with rational real and imaginary parts is understood to be the total number of bits needed to specify the quadruple.

It should be noted that the algorithm does the following matrix computations: matrix addition, matrix multiplication, tensor product operation, iterated matrix addition and multiplication, negative eigenspace projection, and matrix exponentiation. Out of these matrix operations, except for negative eigenspace projection and matrix exponentiation, every other matrix computations can be computed exactly in NC. We refer the reader to the survey of von zur Gathen [Gat93] for more details.

As was mentioned in the previous subsection, the negative eigenspace projection in step 2(a) and matrix exponentiation in step 2(b) of the algorithm cannot be computed exactly. Based on the discussion in the previous subsection, however, it suffices to show that both the computational problems can be approximated (within the accuracy prescribed in the previous subsection) in NC. Below we describe how these computations can be approximated in NC.

Negative eigenspace projection: Given a Hermitian operator $H \in \mathrm{Herm}\,(\mathbb{C}^n)$, the negative eigenspace projection problem asks for computing a positive semidefinite operator $\Delta \in \mathrm{Pos}\,(\mathbb{C}^n)$ such that

$$\|\Delta - \Lambda\|_\infty < \epsilon$$

where Λ is the projection operator onto the negative eigenspace of H. That this computation can be performed in NC can be shown by considering a two step computation as

described below.

1. Compute the positive and negative eigenvalues of H, each of them within very high precision.
2. Compute the projection onto the space spanned by the eigenvectors of H with negative eigenvalues.

The first step of the above procedure can be computed as follows. It is a well known fact that the characteristic polynomial of an operator can be computed exactly in NC [Csa76]. This can be done by computing each coefficient of the polynomial using the Newton identities. First of all, writing the characteristic polynomial of the operator H as a monic polynomial

$$\chi(x) = \sum_{i=0}^{n} (-1)^{n-i}\ a_{n-i}\ x^i$$

we observe that $a_1 = \mathrm{Tr}(M)$. Second of all, the Newton identities claim that the coefficients a_k can be computed as

$$a_k = \frac{1}{k} \sum_{i=1}^{k} (-1)^{i-1}\ a_{k-i}\, \mathrm{Tr}\left(A^i\right).$$

Given that the entries of M have rational real and imaginary parts, one can write the characteristic polynomial as an integer polynomial. It is known that roots of integer polynomials can be approximated within very high precision in NC provided they are real [BFKT86, BP98]. For instance, the algorithm of Bini and Pan [BP98] to approximate the roots of integer polynomials is a two step process roughly sketched as follows. The first step of the algorithm computes an interleaving set such that some eigenvalues can be computed within the desired accuracy and the rest of the eigenvalues are isolated in the interleaved set. The second step of the algorithm approximates the isolated eigenvalues using Newton-Raphson method. We briefly remark that one can compute complex roots of an integer polynomial in NC [Nef96]. For our purposes, it is important to differentiate between positive and negative roots without any error. However, this is not an issue as distinct non-zero roots of integer polynomials can neither be too close to one another nor to zero [Mah61, Mah64, Bug04]. For two distinct roots q and r of a polynomial $p(x) = a_0 + a_1 x + \cdots + a_n x^n$, it holds that

$$|q - r| > (n+1)^{-(n+1)} \left(\max_i \{|a_i|\}\right)^{-n+1}.$$

This bound is sufficiently large enough and the root finding algorithm can approximate the roots within this separation in NC. The fact that these roots are not close to zero can be observed by considering the polynomial $xp(x)$, which has a zero root. Using these facts,

we can approximate the roots of the characteristic polynomial and can distinguish between negative and positive real roots without any error.

That the second step of the computation of the negative eigenspace projection problem can be done in NC follows from the fact that in NC one can compute the eigenspace corresponding to any eigenvalue of H. For an eigenvalue λ of H, let $\widetilde{\lambda}$ be a sufficiently close approximation of λ. Consider the non-singular operator

$$F = \widetilde{\lambda}\mathbb{I}_{\mathbb{C}^n} - H.$$

It holds that F^{-1} divided by a sufficiently close approximation of the largest singular value of F^{-1} is a very good approximation of the projection onto the eigenspace of M corresponding to eigenvalue λ. This can be seen as follows. For $\widetilde{\lambda}$ being a sufficiently close approximation of λ, the largest singular value of F^{-1} is $\frac{1}{\left|\widetilde{\lambda}-\lambda\right|}$. Let $\kappa > 0$ be a sufficiently close approximation of this quantity. Then the action of $(\kappa F)^{-1}$ on

$$H = \lambda\Pi + \sum_{j:\lambda_j \neq \lambda} \lambda_j v_j v_j^*$$

is the operator

$$\left(\frac{1}{\kappa\left(\widetilde{\lambda}-\lambda\right)}\lambda\Pi + \frac{1}{\kappa\left(\widetilde{\lambda}-\lambda\right)}\sum_{j:\lambda_j\neq\lambda}\left(\widetilde{\lambda}-\lambda_j\right)\lambda_j v_j v_j^*\right).$$

We can choose κ to be a sufficiently close approximation of $\widetilde{\lambda}-\lambda$ such that

$$\frac{1}{\kappa\left|\left(\widetilde{\lambda}-\lambda\right)\right|}\left\|\sum_{j:\lambda_j\neq\lambda}\left(\widetilde{\lambda}-\lambda_j\right)\lambda_j v_j v_j^*\right\|_\infty < \epsilon$$

for a choice of $\epsilon > 0$ sufficient for our need. In such a case, $(\kappa F)^{-1}$ serves as an approximation of the projection onto the eigenspace of H corresponding to eigenvalue λ. In case $\widetilde{\lambda} = \lambda$, we perturb $\widetilde{\lambda}$ slightly to make F non-singular and proceed as before. By summing the approximate projections onto the eigenspaces corresponding to the negative eigenvalues (which can be distinguished from positive eigenvalues without any error), we can obtain the desired approximation Δ^2 of Λ.

Matrix exponentiation: Given a Hermitian operator $H \in \mathrm{Herm}\left(\mathbb{C}^n\right)$ with the promise that $\|H\|_\infty < k$, the matrix exponentiation problem asks for computing a positive semidefinite operator $X \in \mathrm{Pos}\left(\mathbb{C}^n\right)$ such that

$$\|X - \exp(H)\|_\infty < \epsilon.$$

The fact that matrix exponentials can be approximated in NC follows by truncating the series

$$\exp(H) = \mathbb{I}_{\mathbb{C}^n} + H + \frac{H^2}{2} + \frac{H^3}{6} + \cdots \tag{5.17}$$

to a number of terms linear in $k + \log(1/\epsilon)$. The fact that only linear number of terms in $k + \log(1/\epsilon)$ is required follows from a very simple observation on matrix exponentials and the above series. More specifically, for any $x \in \mathbb{R}$ it holds that

$$\exp(x) - \sum_{i=0}^{r-1} \frac{x^i}{i!} \;<\; \frac{x^r}{r!}\left(1 - \frac{r+1}{r+1-x}\right)$$

and consequently, given the bound on the operator norm of H, by taking the number of terms r to be linear in $k + \log(1/\epsilon)$, we can ensure that $\|X - \exp(H)\|_\infty < \epsilon$. To ensure that the resulting approximation X is positive semidefinite, one can take an odd number of terms in (5.17).

Given that every matrix computation can be either computed exactly or approximated within the accuracy required by the algorithm by NC computations, it follows that every iteration of the algorithm can be done in NC. To accommodate for the accuracy, we require that the entries of the matrices processed by the algorithm are specified by $O(M)$ bits, where M is the dimension of vector spaces $\mathcal{X}$ and $\mathcal{Y}$. As the number of iterations required by the algorithm is at most logarithmic in M, the whole algorithm can be implemented in NC by composing the circuits corresponding to each iteration of the algorithm in the most natural way. Given that the first stage of the simulation can be done exactly in NC(poly) as discussed at the start of this section, and the second stage can be done in NC, it follows that the overall computation is in NC(poly). Hence the promise problem $A \in$ QIP can be decided by an NC(poly) circuit, thereby showing that $A \in$ NC(poly) = PSPACE, as desired.

5.5 Subsequent work

There have been two recent developments reported after the main result in this chapter appeared in STOC 2010 [JJUW09]. The first result, due to Jain and Yao [JY11], concerns a class of semidefinite programs that admit an efficient parallel algorithm. The second result, due to Gutoski and Wu [GW10], concerns further applications of matrix multiplicative weights update method in quantum computational complexity theory. The second result

is a generalization of Wu's alternate proof of our main result [Wu10]. We briefly discuss both the results below.

Jain and Yao demonstrated that a general class of semidefinite programs, which they refer to as *positive semidefinite programs*, can be approximately solved in NC. Their result generalizes a result of Luby and Nisan [LN93] on positive instances of linear programs (a subsequent paper by Young [You01] showed an NC algorithm for such linear programs with some advantage over Luby-Nisan algorithm). In more detail, they consider the following semidefinite program:

Primal problem		Dual problem	
minimize:	$\langle A, X \rangle$	maximize:	$\langle b, y \rangle$
subject to:	$\langle F_i, X \rangle \geq b_i, \forall i \in [m]$	subject to:	$\sum_{i=1}^{m} y[i] F_i \preceq A,$
	$X \in \mathrm{Pos}(\mathcal{X})$.		$y \in \mathbb{R}_+^m.$

Here $\mathcal{X}$ is a complex Euclidean space of dimension n, $\mathbb{R}_+^m$ denote the non-negative orthant, and the Hermitian operators $A, F_1, \ldots, F_m$ are positive semidefinite operators. Jain and Yao [JY11] showed that there exists an algorithm that outputs an $(1+\epsilon)$-approximation to the optimum value of above semidefinite program such that the algorithm uses a number of iteration T that depends polynomially on $\log(n), \log(m)$ and $1/\epsilon$. Every iteration of their algorithm uses matrix computations that can be computed exactly or approximately in NC. Since the number of iteration is small, the whole algorithm can be parallelized. Note that it is highly unlikely that all semidefinite programs can be approximated to high accuracy in NC — for even if this were true just for linear programs it would imply NC = P [DLR79, Ser91, Meg92]. Moreover, it is not known whether the semidefinite program for single-coin quantum interactive proof system can be rewritten in the form described in Ref. [JY11].

The containment of QIP in PSPACE naturally raises the following question: Can the matrix multiplicative weights update method be used to demonstrate such a containment for other quantum complexity classes? One natural candidate to consider is QRG(2), the class of problems that admit a competing prover strategy of the following form. The class QRG(2) consists of all promise problems that admit two-message competing-prover quantum interactive proof system, where the verifier simultaneously asks both provers (*yes*-prover and *no*-prover) a question, receives their answers, and decides to accept or reject based on the questions and the answers received. The maximum success probability of the *yes*-prover in a one-round quantum refereed game can be written as a semidefinite program [GW07]. The classical analogue of this class is known to coincide with PSPACE [FK97]. Recently, Gutoski and Wu [GW10] showed that the class QRG(2) indeed coincides with PSPACE.

Their result is more general than what we have stated above. They consider a more general class of competing prover quantum interactive proofs, wherein the verifier interacts and exchanges several messages with the yes-prover and then interacts and exchanges several messages with the no-prover. They call such a proof system *doubly quantum interactive proof systems* and show that such proof system also coincide with PSPACE. It also follows from their result that the classical variant of this model is also equal to PSPACE. A consequence of their result is that public-coin variant of classical or quantum refereed games also coincide with PSPACE. This follows easily from the observation that in the public-coin variant, the quantum (classical) refereed game can be transformed into a doubly quantum (classical) interactive proof system.

One should not confuse doubly quantum interactive proof systems with polynomial-round quantum refereed games, wherein the verifier asks questions to both the provers simultaneously. While Gutoski and Wu show that doubly quantum interactive proofs are contained in PSPACE, it is known that quantum refereed games characterize EXP [FK97, GW07]. Their result on doubly quantum interactive proofs subsumes the main result of this chapter.

Chapter 6

Quantum XOR games and parallel repetition

This chapter is based on the Ref. [CSUU08] and is a joint work with Richard Cleve, William Slofstra, and Falk Unger.

The chapter introduces a class of two-prover one-round games wherein the verifier asks questions to each prover who each respond with a bit, and the verifier's verdict is based on the parity of the bits received. Such games are known as XOR games. The main result of this chapter is that when the the provers are allowed to process quantum information but the communication channel between the provers and the verifier is classical, a *perfect parallel repetition* holds in the following sense. The optimal success probability of the provers for simultaneously playing a collection of XOR games is exactly the product of the individual optimal success probabilities. This property is remarkable in view of the fact that, in the classical case it does not hold [BCLKP02]. The parallel repetition theorem is proved in two steps. The first step is to analyze parities of XOR games using semidefinite programming techniques and the second step is to relate the parities of XOR games with parallel repetition of XOR games via Fourier analysis.

The organization of the chapter is as follows.

- In Section 6.1, we introduce classical and quantum XOR games and discuss various known results about them.
- In Section 6.2, we briefly discuss Tsirelson's vector characterization of quantum XOR games and relate it to a simple semidefinite program that captures the set of potential strategies adopted by the quantum provers.
- In Section 6.3, we mention the two main theorems proved in this chapter and briefly describe the techniques used to prove them.

- In Section 6.4, we establish the main results of the chapter. We also show an unpublished proof due to Watrous [Wat04] that demonstrates that perfect parallel repetition theorem does not hold even for binary games (games with answer sets $\{0, 1\}$).
- In Section 6.5, we relate the success probability of the quantum provers in an XOR game with the corresponding Feige-Lovász relaxations [FL92], a fact noted earlier in Refs. [FG95, FKO07].
- In Section 6.6, we briefly discuss works related to the main result and techniques used in this chapter.

6.1 Quantum XOR games

An XOR game is operationally defined as follows. For a predicate $f : S \times T \to \{0, 1\}$ and a probability distribution π on $S \times T$, an XOR game $G = (f, \pi)$ is a one-round game between two provers and a verifier that proceeds as follows.

1. The verifier selects a pair of questions $(s, t) \in S \times T$ according to the probability distribution π. One assumes without loss of generality that the sets S and T are disjoint.
2. The verifier sends one question to each prover: s to prover Alice and t to prover Bob. Alice and Bob are forbidden from communicating with each other once the game starts.
3. Alice sends $a \in \{0, 1\}$ and Bob sends $b \in \{0, 1\}$ to the verifier.
4. The verifier accepts the response of the provers if and only if $a \oplus b = f(s, t)$.

In the classical version, the provers are allowed to have unlimited computing power, but are restricted to possessing classical information. In the quantum version, the provers have the ability to process quantum information and therefore, for instance, can share entanglement. In both versions, the communication channel between the provers and the verifier is classical. For an XOR game G, one defines its *classical value* $\omega_c(G)$ as the maximum possible success probability achievable by a classical strategy. Similarly, the *quantum value* $\omega_q(G)$ is defined as the maximum possible success probability achievable by a quantum strategy. A detailed description of what constitutes as a valid quantum strategy is given in the following section.

There is an association between two-prover one-round games and multi-prover interactive proof systems. For example, XOR games can be associated with two-prover one-round interactive proof systems as follows. For two polynomial-time computable functions $c, s : \mathbb{N} \to [0, 1]$, one defines an *XOR proof system* for a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ as

follows. On input string x, the provers and the verifier play an XOR game $G_x = (f_x, \pi_x)$. The question sets of both the provers S_x and T_x consist of strings of length polynomial in $|x|$, the probability distribution π_x can be sampled efficiently, and the function f_x can be computed in polynomial-time. The proof system satisfies the following two properties.

1. *Completeness.* For every $x \in A_{\text{yes}}$, there exists a strategy of the provers that causes the verifier to accept with probability at least $c(|x|)$. In other words, the maximum acceptance probability over prover's strategies is at least $c(|x|)$.

2. *Soundness.* For every $x \in A_{\text{no}}$, regardless of what strategy the provers adopt, the verifier accepts with probability at most $s(|x|)$. In other words, the maximum acceptance probability over prover's strategies is at most $s(|x|)$.

Such proof systems are denoted $\oplus\text{MIP}_{c,s}(2,1)$ in the classical case and $\oplus\text{MIP}^*_{c,s}(2,1)$ in the quantum case. The asterisk (*) denotes the presence of quantum entanglement between the provers. In the sections to follow, we will always refer to XOR games rather than XOR proof systems, their relationship being made explicit above.

Cleve, Høyer, Toner, and Watrous [CHTW04a] pointed out that earlier results of Bellare, Goldreich and Sudan [BGS98], and Håstad [Hås01] imply that, in the case of classical provers, these proof systems have sufficient expressive power to recognize every problem in NEXP (with soundness $s = 11/16 + \epsilon$ and completeness $c = 12/16 - \epsilon$, for arbitrary small $\epsilon > 0$). Thus, although these proof systems appear to be very restrictive, they can recognize any problem that an unrestricted multi-prover interactive proof system can recognize with bounded gap between completeness and soundness probabilities.

Although not noted in the paper, the results mentioned in their paper [CHTW04a] also imply that the quantum XOR proof systems lie in EXP (this was first noted in a presentation at CCC 2004 [CHTW04b]). This containtment was subsequently improved by Wehner [Weh06] who showed that quantum XOR proof systems are contained in QIP(2). Since QIP = PSPACE, Wehner's result implies that

$$\oplus\text{MIP}^*_{c,s}(2,1) \subseteq \text{PSPACE}$$

for any $c, s > 0$ such that $c - s$ is at least some inverse polynomial. Thus, quantum XOR proof systems are strictly less powerful than their classical counterpart under the widely believed assumption that PSPACE $\neq$ NEXP. This is the best complexity-theoretic upper bound known on $\oplus\text{MIP}^*_{c,s}(2,1)$ to date. On the lower bound front, Cleve, Gavinsky and Jain [CGJ09] showed that any problem in NP admits a quantum XOR proof systems with completeness $1 - \epsilon$ and soundness $1/2 + \eta$ for any $\epsilon, \eta > 0$. It is also not hard to observe that

$$\text{SZK} \subseteq \oplus\text{MIP}^*_{c,s}(2,1)$$

where the completeness $c = 1 - \epsilon$ and soundness $s = 1/2 + \eta$ for any arbitrary small $\epsilon, \eta > 0$. Here SZK is the class of problems that admit a statistical zero-knowledge proof system [SV03].

Returning to XOR games, quantum physicists have, in a sense, been studying them since the 1960s, when John Bell introduced his celebrated results that are now known as Bell inequality violations [Bel64]. An example is the *CHSH* game, named after the authors of the paper [CHSH69]. In this game, $S = T = \{0, 1\}$, π is the uniform distribution on $S \times T$, and $f(s, t) = s \wedge t$. It is well known that, for the *CHSH* game, the best possible classical strategy succeeds with probability $3/4$, whereas the best possible quantum strategy succeeds with probability $\cos^2\left(\frac{\pi}{8}\right) = \frac{\sqrt{2}+1}{2\sqrt{2}} \approx 0.85$ [CHSH69, Tsi80].

6.2 Tsirelson's vector characterization for quantum XOR games

In this section, we briefly discuss Tsirelson's vector characterization of quantum XOR games, which states that any valid quantum strategy for an XOR game can be characterized by a vector system. To this end, we first proceed to describe a quantum strategy for an XOR game. A quantum strategy for an XOR game proceeds as follows. The provers Alice and Bob share a state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$. One does not lose any generality by assuming that $\mathcal{X} = \mathcal{Y} = \mathbb{C}^N$, but for the sake of clarity we will denote Alice's and Bob's part by $\mathcal{X}$ and $\mathcal{Y}$. Upon receiving questions s and t, each prover applies a two-outcome measurement operation on the respective part of their bipartite shared state:

$$\left\{\Pi_s^0, \Pi_s^1\right\} \qquad \text{and} \qquad \left\{\Delta_t^0, \Delta_t^1\right\}.$$

The measurement operators satisfy

$$\sum_{a \in \{0,1\}} \Pi_s^a = \mathbb{I}_{\mathcal{X}} \qquad \text{and} \qquad \sum_{b \in \{0,1\}} \Delta_t^b = \mathbb{I}_{\mathcal{Y}}.$$

Alice and Bob send the outcome of the measurement operation to the verifier, who checks whether the parity of the bits received is indeed $f(s, t)$ or not. Denoting the strategy by $\mathcal{S}$, the success probability of the provers that follow the strategy $\mathcal{S}$ is given by the expression

$$\omega_q(G, \mathcal{S}) = \sum_{s,t} \sum_{a,b} \pi(s, t) V(a, b|s, t) \left\langle \psi \right| \Pi_s^a \otimes \Delta_t^b \left| \psi \right\rangle \tag{6.1}$$

where $V(a, b|s, t) = 1$ if and only if $a \oplus b = f(s, t)$. We point out two straightforward facts about general quantum strategies. First of all, since we are interested in maximizing (6.1),

one can without loss of generality assume that the state shared by the provers is a pure state. This follows easily from a standard convexity argument. Second of all, we can assume that for any optimal strategy the measurement operators are projection operators, and hence $\langle \Pi_s^0, \Pi_s^1 \rangle = 0$ and $\langle \Delta_t^0, \Delta_t^1 \rangle = 0$ [CHTW04a].

We do not place any restriction on the amount of entanglement the provers possess for playing the game. However, for the case of XOR games, a finite dimensional state suffices for the provers to perform optimally. Of course, the dimension of the state depends on the number of questions asked to the provers. This fact follows from the vector characterization of XOR games, which we describe below. To begin, fix an optimal strategy

$$|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y} \qquad \text{and} \qquad \left\{ \left\{ \Pi_s^0, \Pi_s^1 \right\} : s \in S \right\} \qquad \text{and} \qquad \left\{ \left\{ \Delta_t^0, \Delta_t^1 \right\} : t \in T \right\}$$

of the provers. Here the measurement operators $\{\Pi_s^0, \Pi_s^1\}$ and $\{\Delta_t^0, \Delta_t^1\}$ are projective measurement for every choice of $(s,t) \in S \times T$. Now for every choice of question pair (s,t), let

$$X_s = \Pi_s^0 - \Pi_s^1 = \sum_{a \in \{0,1\}} (-1)^a \Pi_s^a \qquad \text{and} \qquad Y_t = \Delta_t^0 - \Delta_t^1 = \sum_{b \in \{0,1\}} (-1)^b \Delta_t^b$$

be Hermitian unitary operators acting on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Noting that

$$V(a,b|s,t) = \frac{1 + (-1)^{a \oplus b} (-1)^{f(s,t)}}{2}$$

one can write the quantum value of an XOR game as

$$\omega_q(G) = \frac{1}{2} + \frac{1}{2} \sum_{s,t} \pi(s,t) (-1)^{f(s,t)} \langle \psi | X_s \otimes Y_t | \psi \rangle = \frac{1 + \varepsilon_q(G)}{2}.$$

The quantity $\varepsilon_q(G)$ is called the *quantum bias* of the game. To prove the desired result, it will be convenient for us to work with the quantum bias instead of the quantum value of the game. We remark that one can similarly define the *classical bias* of an XOR game G as

$$\varepsilon_c(G) = 2\omega_c(G) - 1.$$

One nice property about an XOR game is that the quantum bias can be characterized by a system of unit vectors in the following way.

Theorem 16 (**[Tsi85, CHTW04a]**). *Let S and T be finite sets, and let $|\psi\rangle$ be a pure quantum state with support on a bipartite vector space $\mathcal{X} \otimes \mathcal{Y}$ such that $\dim(\mathcal{X}) = \dim(\mathcal{Y}) = n$. For each $s \in S$ and $t \in T$, let X_s and Y_t be Hermitian unitary operators on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then there exists real unit vectors x_s and y_t in $\mathbb{R}^{2n^2}$ such that*

$$\langle \psi | X_s \otimes Y_t | \psi \rangle = \langle x_s, y_t \rangle$$

for all $s \in S$ and $t \in T$.

Conversely, suppose that S and T are finite sets, and x_s and y_t are unit vectors in $\mathbb{R}^N$ for each $s \in S$ and $t \in T$. Let $\mathcal{X}$ and $\mathcal{Y}$ be vector space of dimension $2^{\lceil N/2 \rceil}$ and $|\psi\rangle$ be a maximally entangled state on $\mathcal{X} \otimes \mathcal{Y}$. Then there exists Hermitian unitary operators X_s and Y_t on $\mathcal{X}$ and $\mathcal{Y}$, respectively, such that

$$\langle x_s, y_t \rangle = \langle \psi | X_s \otimes Y_t | \psi \rangle$$

for all $s \in S$ and $t \in T$.

Theorem 16 implies that any quantum strategy adopted by Alice and Bob can be characterized by some choice of unit vectors $\{x_s : s \in S\}$ and $\{y_t : t \in T\}$, the association of the unit vectors with a quantum strategy being made explicit above:

$$\varepsilon_q(G) = \max_{\{x_s\},\{y_t\}} \quad \sum_{s,t} \pi(s,t)(-1)^{f(s,t)} \ \langle x_s, y_t \rangle \,. \tag{6.2}$$

The proof of the theorem is not difficult to follow. Given two Hermitian unitary operators $X_s \in \mathrm{Herm}\,(\mathcal{X})$ and $Y_t \in \mathrm{Herm}\,(\mathcal{Y})$, and a bipartite quantum state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$, the corresponding vectors x_s and y_t are defined as

$$x_s := (X_s \otimes \mathbb{I}_{\mathcal{Y}}) |\psi\rangle \qquad \text{and} \qquad y_t = (\mathbb{I}_{\mathcal{X}} \otimes Y_t) |\psi\rangle \,.$$

The vectors may not be real vectors but nevertheless can be embedded into a real space of twice the dimension. The conversion of unit vectors into a quantum strategy follows from noting that for two n-dimensional real vectors, there exist a set of Hermitian unitary operators $\{M_1, \ldots, M_n\} \subset \mathrm{Herm}\,(\mathcal{Z})$ such that

$$M_i^2 = \mathbb{I}_{\mathcal{Z}} \qquad \text{and} \qquad M_i M_j = M_j M_i$$

for distinct i and j. We associate the vectors x_s and y_t with the Hermitian unitary operators X_s and Y_t as

$$X_s = \sum_{i=1}^{n} x_s[i] M_i \qquad \text{and} \qquad Y_t = \sum_{i=1}^{n} y_t[i] M_i^{\top}$$

and the state to be the maximally entangled state in $\mathcal{Z} \otimes \mathcal{Z}$. This construction works and the detail is left to the readers. Now we proceed to describe the semidefinite programming characterization of the quantum bias of the game.

Let A be the matrix with matrix entries $A[s,t] = \pi(s,t)(-1)^{f(s,t)}$. We call A the *cost matrix* of an XOR game. Assume the question sets to be $S = [N]$ and $T = [M]$. Then A is an N-by-M real matrix such that the absolute values of its entries of A sum to 1.

Conversely, any real matrix A such that the absolute value of its entries sum up to 1 represents an XOR game. Let

$$C = \frac{1}{2}\begin{pmatrix} 0 & A \\ A^\top & 0 \end{pmatrix}.$$

be the *symmetric cost matrix* of the game. Then the quantum bias of G can be expressed as the following semidefinite program:

$$\begin{aligned} \varepsilon_q(G) = \text{maximum:} \quad & \langle C, X \rangle \\ \text{subject to:} \quad & \forall\ k \in [N+M], \quad \langle E_{kk}, X \rangle = X[k,k] = 1, \\ & X \in \mathrm{Pos}\left(\mathbb{R}^{N+M}\right). \end{aligned}$$

The equivalence between the quantum bias and the above semidefinite programming problem follows straightforwardly from item 5 of Theorem 3 on page 20.

6.3 Proof structure

For any two XOR games $G_1 = (f_1, \pi_1)$ and $G_2 = (f_2, \pi_2)$, define their *sum* (modulo 2) as the following XOR game

$$G_1 \oplus G_2 = (f_1 \oplus f_2, \pi_1 \times \pi_2).$$

In this game, the verifier begins by choosing questions $(s_1, t_1) \in S_1 \times T_1$ according to the probability distribution π_1 and $(s_2, t_2) \in S_2 \times T_2$ according to the probability distribution π_2. The verifier sends (s_1, s_2) to Alice and (t_1, t_2) to Bob. Alice and Bob are expected to reply with a bit each and they win if and only if their respective outputs, a and b, satisfy $a \oplus b = f_1(s_1, t_1) \oplus f_2(s_2, t_2)$.

The most straightforward way for Alice and Bob (who may or may not share entanglement) to play $G_1 \oplus G_2$ is to optimally play G_1 and G_2 separately, producing outputs (a_1, b_1) for G_1 and (a_2, b_2) for G_2, and then to output $a = a_1 \oplus a_2$ and $b = b_1 \oplus b_2$ respectively. It is straightforward to calculate that the above method for playing $G_1 \oplus G_2$ succeeds with probability

$$\omega(G_1)\omega(G_2) + (1 - \omega(G_1))(1 - \omega(G_2)). \tag{6.3}$$

We dropped the subscript referring to the classical and quantum case in the above equation to stress on the fact that the above lower bound holds for both cases. One can ask the question whether the lower bound is tight, or, in other words, an optimal strategy for the game $G_1 \oplus G_2$ is to play both games independently and optimally as mentioned above.

Unsurprisingly, the answer to the question is *no* for *classical* strategies. To see why this is so, note that, using this approach for the XOR game $CHSH \oplus CHSH$, produces a success probability of 5/8. A better strategy is for Alice to output $a = s_1 \wedge s_2$ and Bob to

output $b = t_1 \wedge t_2$. It is straightforward to verify that this latter strategy succeeds with probability 3/4. However, we show that the in the quantum case, an optimal strategy is indeed the strategy described above.

Theorem 17 (**Additivity Theorem**). *For any two XOR games G_1 and G_2, an optimal quantum strategy for playing $G_1 \oplus G_2$ is for Alice and Bob to optimally play G_1 and G_2 separately, producing outputs a_1, b_1 for G_1 and a_2, b_2 for G_2, and then to output $a = a_1 \oplus a_2$ and $b = b_1 \oplus b_2$. In other words,*

$$\varepsilon_q(G_1 \oplus G_2) = \varepsilon_q(G_1) \cdot \varepsilon_q(G_2).$$

A straightforward consequence of the above theorem is that if $G_1, G_2, \ldots, G_k$ are k XOR games, then

$$\varepsilon_q\left(\bigoplus_{i=1}^{k} G_i\right) = \prod_{i=1}^{k} \varepsilon_q(G_i).$$

The proof of Theorem 17 uses semidefinite programming techniques, as described in the subsequent section. The additivity theorem will be used to prove a perfect parallel repetition theorem for quantum XOR games.

For any sequence of XOR games $G_1 = (f_1, \pi_1)$, ..., $G_n = (f_n, \pi_n)$, define their *conjunction*, denoted by

$$\bigotimes_{j=1}^{n} G_j$$

as follows. The verifier chooses questions $((s_1, t_1), \ldots, (s_n, t_n)) \in (S_1 \times T_1) \times \cdots \times (S_n \times T_n)$ according to the product distribution $\pi_1 \times \cdots \times \pi_n$, and sends $(s_1, \ldots, s_n)$ to Alice and $(t_1, \ldots, t_n)$ to Bob. Alice and Bob output bits $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$, respectively, and win if and only if their outputs simultaneously satisfy the following n conditions:

$$a_k \oplus b_k = f_k(s_k, t_k)$$

for all $k \in [n]$. One way for Alice and Bob to play conjunction of n games $G_1, G_2, \ldots, G_n$ is to independently play each game optimally. This succeeds with probability

$$\prod_{j=1}^{n} \omega(G_j).$$

Again the dropping of the subscript in the above equation refer to the fact that the above lower bound above holds for both cases. One can ask the question whether this lower bound is tight, or, in other words, an optimal strategy for the conjunction of games $G_1, \ldots, G_n$ is to play each game independently and optimally as mentioned above.

Again the answer is *no* for classical strategies. It was shown by Barrett, Collins, Hardy, Kent and Popescu [BCLKP02] that for the CHSH game

$$\omega_c(CHSH \otimes CHSH) = 10/16 > 9/16 = \omega_c(CHSH)\omega_c(CHSH).$$

However, using the additivity theorem, we show that the answer is *yes* for quantum strategies.

Theorem 18 (**Parallel Repetition Theorem**). *For any XOR games $G_1, \ldots, G_n$, we have that*

$$\omega_q\left(\bigotimes_{j=1}^{n} G_j\right) = \prod_{j=1}^{n} \omega_q(G_j).$$

This theorem can be viewed as the quantum version of Raz's parallel repetition theorem [Raz98] for the restricted class of XOR games. We call it a *perfect* parallel repetition theorem because the probabilities are multiplicative in the exact sense. The proof of Theorem 18 is based on Theorem 17 combined with Fourier analysis techniques for Boolean functions.

Prior to the result presented in this chapter, there was no known parallel repetition theorem along the lines of Raz [Raz98] for general quantum games (where the players share entanglement). Subsequent to this result, a parallel repetition theorem for quantum unique games have been proved [KRT08, Upa07], which we discuss in the chapter following this one.

The prefect parallel repetition theorem fails to hold for general binary games (where the answer sets of both the provers is $\{0, 1\}$). In particular, Watrous [Wat04] showed that there exists a binary game for which $\omega_q(G) = \omega_q(G \otimes G) = 2/3$, as in the classical case. The game considered by Watrous was originally defined by Feige and Lovász [FL92]. We describe the result in the section following this one.

For any XOR game G, the semidefinite programming relaxations of the classical value of G due to Feige and Lovász [FL92] have value equal to the quantum value of G. Although this was also noted previously [FG95, FKO07], for completeness, we show an explicit proof of this in Section 6.5. It is important to note that, *for general games*, the relationship between their quantum values and the Feige-Lovász relaxations of their classical values are not understood. As far as we know, neither quantity bounds the other for general games. However, using the fact that they are equivalent for XOR games combined with our Theorem 18, we can deduce that, whenever $G_1, \ldots, G_n$ are XOR games, the quantum value of $\otimes_{j=1}^{n} G_j$ coincides with the optimum values of the associated Feige-Lovász relaxations. (Note that this does not reduce our Theorem 18 to the results mentioned in Feige and Lovász since that would entail circular reasoning.)

6.4 Proof of the main result

In this section, we prove the main result of the chapter. The first subsection proves the additivity theorem, which is followed by a subsection on the proof of the perfect parallel repetition theorem for quantum XOR games. In the final subsection, we show Watrous's proof on the existence of a binary game that does not obey the perfect parallel repetition theorem in the quantum setting.

6.4.1 Proof of additivity theorem

We first establish the additivity theorem. The proof of the additivity theorem goes via showing that the optimum value of the semidefinite program corresponding to the XOR game $G_1 \oplus G_2$ is the product of the optimum values of the semidefinite program corresponding to games G_1 and G_2. The proof below is somewhat different from the one that appeared in the Ref. [CSUU08] but still uses semidefinite programming duality. We first recall that the quantum bias of game G with question sets $S = [N]$ and $T = [M]$ is associated with the following semidefinite program:

$$\underline{\text{Primal problem (P)}}$$

$$\begin{aligned} \varepsilon_q(G) = \text{maximize:} \quad & \langle C, X \rangle \\ \text{subject to:} \quad & \forall\ k \in [N+M], \quad X[k,k] = 1, \\ & X \in \text{Pos}\left(\mathbb{R}^{N+M}\right). \end{aligned}$$

The use of the term "maximize" is justified in the above semidefinite program as demonstrated below. The dual of the semidefinite program is

$$\underline{\text{Dual problem (D)}}$$

$$\begin{aligned} \varepsilon_q(G) = \text{minimize:} \quad & \sum_{k=1}^{N} y[k] + \sum_{l=1}^{M} z[l] \\ \text{subject to:} \quad & Z = \begin{pmatrix} \Lambda(y) & -\frac{1}{2}A \\ -\frac{1}{2}A^\top & \Lambda(z) \end{pmatrix} \in \text{Pos}\left(\mathbb{R}^{N+M}\right), \\ & y \in \mathbb{R}^N \quad \text{and} \quad z \in \mathbb{R}^M. \end{aligned}$$

Here $\Lambda : \mathbb{R}^{N+M} \to \text{Herm}\left(\mathbb{R}^{N+M}\right)$ is an operator that maps any vector into a diagonal matrix. Note that we have Slater points for both primal ($\mathbb{I}$) and dual problem ($(2\bar{e}, 2\mathbb{I})$)and

therefore strong duality holds (refer to Theorem 8). Here $\bar{e}$ is the all one vector. We first show that every optimal solution of the dual problem satisfies a nice property stated in the following lemma. The proof of the lemma stated below is similar to the proof of Lemma 35 in Chapter 7.

Lemma 19. *Let (y, z) be any optimal solution of the dual. Then it holds that*

$$\sum_{k=1}^{N} y[k] = \sum_{l=1}^{M} z[l].$$

Proof. Let (y, z) be an optimal dual solution. Since (y, z) forms a feasible dual solution, we have that

$$\sum_{k=1}^{N} y[k] > 0 \qquad \text{and} \qquad \sum_{l=1}^{M} z[l] > 0.$$

The matrix Z is positive semidefinite and hence it can be written as

$$Z = \begin{pmatrix} \Lambda(y) & -\frac{1}{2}A \\ -\frac{1}{2}A^\top & \Lambda(z) \end{pmatrix} = \begin{pmatrix} B \\ D \end{pmatrix} \begin{pmatrix} B \\ D \end{pmatrix}^\top = \begin{pmatrix} BB^\top & BD^\top \\ DB^\top & DD^\top \end{pmatrix} \succeq 0.$$

Hence (abusing notation) we have $\Lambda(y) = BB^\top$, $\Lambda(z) = DD^\top$, and $BD^\top = -\frac{A}{2}$. It is also immediate from above that the diagonal entries of $BB^\top$ and $DD^\top$ are the vectors y and z respectively. Without loss of generality, assume that

$$p := \sum_{k=1}^{N} y[k] > \sum_{l=1}^{M} z[l] =: q.$$

and let $\lambda > 0$. It is clear that $p, q > 0$. Define a new matrix Z' as

$$Z' = \begin{pmatrix} \lambda B \\ (1/\lambda)D \end{pmatrix} \begin{pmatrix} \lambda B \\ (1/\lambda)D \end{pmatrix}^\top = \begin{pmatrix} \lambda^2 BB^\top & BD^\top \\ DB^\top & (1/\lambda^2)DD^\top \end{pmatrix} \succeq 0.$$

It is clear that Z' is dual feasible for the off-diagonal block of the matrix is $BD^\top = -\frac{A}{2}$ and $Z' \in \text{Pos}\left(\mathbb{R}^{N+M}\right)$. The objective value of the new solution is $\lambda^2 p + (1/\lambda^2)q$, which is strictly less that $p + q$ for $\lambda \in \left(1, \sqrt{p/q}\right)$, contradicting our assumption that $p \neq q$ for optimal dual solution.

□

The lemma implies that for an optimal dual solution, the contribution from dual feasible vectors y and z towards the objective value is equal. Now we proceed to establish the

additivity theorem. Let G_1 and G_2 be two XOR games with cost matrices A_1 and A_2, respectively. Assume that for game G_i, the sets of questions are $S_i = [N_i]$ and $T_i = [M_i]$. The cost matrix of the game $G = G_1 \oplus G_2$ is $A_1 \otimes A_2$ and therefore its symmetric cost matrix is

$$C = \frac{1}{2} \begin{pmatrix} 0 & A_1 \otimes A_2 \\ (A_1 \otimes A_2)^\top & 0 \end{pmatrix}.$$

The primal problem of the SDP for game $G_1 \oplus G_2$ is

Primal problem (D)

$$\begin{aligned} \text{maximize:} \quad & \langle C, X \rangle \\ \text{subject to:} \quad & \forall k_1 \in [N_1 + M_1] \text{ and } k_2 \in [N_2 + M_2], \quad X[(k_1 k_2), (k_1, k_2)] = 1, \\ & X \in \operatorname{Pos}\left(\mathbb{R}^{N_1 N_2 + M_1 M_2}\right). \end{aligned}$$

Similarly, the dual problem can be written as

Dual problem (D)

$$\begin{aligned} \text{minimize:} \quad & \sum_{k_1 \in [N_1]} \sum_{k_2 \in [N_2]} y[(k_1, k_2)] + \sum_{l_1 \in [M_1]} \sum_{l_2 \in [M_2]} z[(l_1, l_2)] \\ \text{subject to:} \quad & Z = \begin{pmatrix} \Lambda(y) & -\frac{1}{2} A_1 \otimes A_2 \\ -\frac{1}{2}(A_1 \otimes A_2)^\top & \Lambda(z) \end{pmatrix} \in \operatorname{Pos}\left(\mathbb{R}^{N_1 N_2 + M_1 M_2}\right), \\ & y \in \mathbb{R}^{N_1 N_2} \quad \text{and} \quad z \in \mathbb{R}^{M_1 M_2}. \end{aligned}$$

As before, we define $\Lambda : \mathbb{R}^{N_1 N_2 + M_1 M_2} \to \operatorname{Herm}\left(\mathbb{R}^{N_1 N_2 + M_1 M_2}\right)$ to be the linear operator that maps a vector to a diagonal matrix. For both choices of $i \in \{1, 2\}$, let

$$\bar{X}_i = \begin{pmatrix} U_i & V_i \\ V_i^\top & W_i \end{pmatrix}$$

be an optimal solution of the primal problem of the SDP for game G_i. Then

$$\bar{X} = \begin{pmatrix} U_1 \otimes U_2 & V_1 \otimes V_2 \\ (V_1 \otimes V_2)^\top & W_1 \otimes W_2 \end{pmatrix}$$

is a feasible solution of the primal problem of the SDP for game $G_1 \oplus G_2$. The feasibility follows from the fact that if the diagonal entries of the matrices U_1, U_2, V_1, and V_2 are 1, then the diagonal entries of the matrices $U_1 \otimes U_2$ and $V_1 \otimes V_2$ are also 1. The positive semidefinite property of $\bar{X}$ follows from the fact that it is a symmetric restriction of $\bar{X}_1 \otimes \bar{X}_2$ (item 6

of Theorem 3). It is a simple calculation to show that $\bar{X}$ achieves an objective value equal to $\langle C_1, \bar{X}_1 \rangle \cdot \langle C_2, \bar{X}_2 \rangle$. Hence

$$\varepsilon_q(G_1 \oplus G_2) \geq \varepsilon_q(G_1)\varepsilon_q(G_2).$$

For the opposite inequality, let $(\bar{y}_i, \bar{z}_i)$ form an optimal dual solution for G_i for both choices of $i \in \{1, 2\}$. By Lemma 19, we have that

$$\sum_{k=1}^{N_i} \bar{y}_i[k] = \sum_{l=1}^{M_i} \bar{z}_i[l].$$

It will be convenient for us to denote the matrix

$$Z_i = \begin{pmatrix} \Lambda_i(\bar{y}_i) & -\frac{1}{2}A_i \\ -\frac{1}{2}A_i^\top & \Lambda_i(\bar{z}_i) \end{pmatrix} \qquad \text{as} \qquad \begin{pmatrix} B_i & -\frac{1}{2}A_i \\ -\frac{1}{2}A_i^\top & D_i \end{pmatrix}$$

from which it follows that

$$\mathrm{Tr}(B_i) = \sum_{k=1}^{N_i} \bar{y}_i[k] \qquad \text{and} \qquad \mathrm{Tr}(D_i) = \sum_{l=1}^{M_i} \bar{z}_i[l]. \tag{6.4}$$

Therefore, by Lemma 19

$$\mathrm{Tr}(B_i) = \mathrm{Tr}(D_i) = \frac{\varepsilon_q(G_i)}{2} \tag{6.5}$$

for both choices of $i \in \{1, 2\}$. Let $Z_i' = Z_i + 2C_i$. We now proceed to show that Z_i is positive semidefinite if and only if Z_i' is positive semidefinite. This follows easily from exploiting the block structure of the matrices Z_i and Z_i', and item 4 of Theorem 3. First of all note that for both choices of $i \in \{1, 2\}$, it holds that

$$\begin{pmatrix} u \\ v \end{pmatrix}^\top \begin{pmatrix} B_i & -\frac{1}{2}A_i \\ -\frac{1}{2}A_i^\top & D_i \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u \\ -v \end{pmatrix}^\top \begin{pmatrix} B_i & \frac{1}{2}A_i \\ \frac{1}{2}A_i^\top & D_i \end{pmatrix} \begin{pmatrix} u \\ -v \end{pmatrix}$$

and therefore

$$Z_i = \begin{pmatrix} B_i & -\frac{1}{2}A_i \\ -\frac{1}{2}A_i^\top & D_i \end{pmatrix} \succeq 0 \qquad \text{if and only if} \qquad Z_i' = \begin{pmatrix} B_i & \frac{1}{2}A_i \\ \frac{1}{2}A_i^\top & D_i \end{pmatrix} \succeq 0.$$

The above implies that the tensor product of Z_i and Z_j' is positive semidefinite. In particular,

$$Z_1 \otimes Z_2' \succeq 0 \qquad \text{and} \qquad Z_1' \otimes Z_2 \succeq 0.$$

and hence the following matrix

$$\bar{Z} := \begin{pmatrix} 2B_1 \otimes B_2 & -\frac{1}{2}A_1 \otimes A_2 \\ -\frac{1}{2}(A_1 \otimes A_2)^\top & 2D_1 \otimes D_2 \end{pmatrix}$$

is positive semidefinite. This follows easily from the fact that $\bar{Z}$ is a symmetric restriction of the matrix $Z_1 \otimes Z_2' + Z_1' \otimes Z_2$. A straightforward implication of this fact is that setting

$$y[(k_1, k_2)] = 2\bar{y}_1[k_1] \cdot \bar{y}_2[k_2] \qquad \text{and} \qquad z[(l_1, l_2)] = 2\bar{z}_1[l_1] \cdot \bar{z}_2[l_2]$$

yields a dual feasible solution to the SDP for $G_1 \oplus G_2$. Hence by (6.4) and (6.5), we have

$$\begin{aligned} \varepsilon_q(G_1 \oplus G_2) &\leq 2\left(\sum_{k=1}^{N_1} y_1[k]\right)\left(\sum_{k=1}^{N_2} y_2[k]\right) + 2\left(\sum_{l=1}^{M_1} z_1[l]\right)\left(\sum_{l=1}^{M_2} z_2[l]\right) \\ &= 2 \cdot \frac{\varepsilon_q(G_1)}{2} \cdot \frac{\varepsilon_q(G_2)}{2} + 2 \cdot \frac{\varepsilon_q(G_1)}{2} \cdot \frac{\varepsilon_q(G_2)}{2} = \varepsilon_q(G_1)\varepsilon_q(G_2). \end{aligned}$$

This finishes the proof of the additivity theorem.

6.4.2 Proof of perfect parallel repetition theorem

This section is devoted to the proof of the perfect parallel repetition theorem of quantum XOR games. To prove the main result, we begin with the following simple probabilistic lemma.

Lemma 20. *For any sequence of binary random variables $X_1, X_2, \ldots, X_n$,*

$$\frac{1}{2^n} \sum_{M \subseteq [n]} \mathbb{E}\left[(-1)^{\oplus_{j \in M} X_j}\right] = \mathbb{P}[X_1 \ldots X_n = 0 \ldots 0].$$

Proof. First of all, note that

$$\prod_{j=1}^{n} (1 + (-1)^{X_j}) \neq 0 \tag{6.6}$$

if and only if $X_j = 0$ for all $j \in [n]$. By the linearity of expectation,

$$\begin{aligned} \frac{1}{2^n} \sum_{M \subseteq [n]} \mathbb{E}\left[(-1)^{\oplus_{j \in M} X_j}\right] &= \mathbb{E}\left[\frac{1}{2^n} \sum_{M \subseteq [n]} (-1)^{\oplus_{j \in M} X_j}\right] \\ &= \mathbb{E}\left[\prod_{j=1}^{n} \left(\frac{1 + (-1)^{X_j}}{2}\right)\right] = \mathbb{P}\left[X_1 \ldots X_n = 0 \ldots 0\right] \end{aligned}$$

where the last equality follows from Eq 6.6.

□

We introduce the following terminology. For any strategy $\mathcal{S}$ (classical or quantum) for any game G, define $\omega(\mathcal{S}, G)$ as the success probability of strategy $\mathcal{S}$ on game G. Similarly, define the corresponding bias as $\varepsilon(\mathcal{S}, G) = 2\omega(\mathcal{S}, G) - 1$. We drop the subscript to emphasize that the claims below holds for both classical and quantum case. Now let $\mathcal{S}$ be any protocol for the game

$$\bigotimes_{j=1}^{n} G_j$$

For each $M \subseteq [n]$, define the protocol $\mathcal{S}_M$ for the sum of XOR games $\{G_i : i \in M\}$ as follows.

1. Run the protocol $\mathcal{S}$, yielding $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ for Alice and Bob, respectively.
2. Alice and Bob output
$$\bigoplus_{j \in M} a_j \qquad \text{and} \qquad \bigoplus_{j \in M} b_j$$
respecively.

Lemma 21. *For any strategy $\mathcal{S}$ for the conjunction of XOR games $G_1, G_2, \ldots, G_n$, we have*

$$\frac{1}{2^n} \sum_{M \subseteq [n]} \varepsilon \left(\mathcal{S}_M, \bigoplus_{j \in M} G_j \right) = \omega \left(\mathcal{S}, \bigotimes_{j=1}^{n} G_j \right).$$

Proof. For all $j \in [n]$, define $X_j = a_j \oplus b_j \oplus f_j(s_j, t_j)$. Then, for all $M \subseteq [n]$, we have

$$\mathbb{E}\left[(-1)^{\bigoplus_{j \in M} X_j}\right] = \varepsilon \left(\mathcal{S}_M, \bigoplus_{j \in M} G_j \right) \qquad \text{and} \qquad \mathbb{P}[X_1 \ldots X_n = 0 \ldots 0] = \omega \left(\mathcal{S}, \bigotimes_{j=1}^{n} G_j \right).$$

The result now follows from Lemma 20.

□

As a corollary to the above lemma, we have

$$\omega_c \left(\bigotimes_{j=1}^{n} G_j \right) \leq \frac{1}{2^n} \sum_{M \subseteq [n]} \varepsilon_c \left(\bigoplus_{j \in M} G_j \right) \quad \text{and} \quad \omega_q \left(\bigotimes_{j=1}^{n} G_j \right) \leq \frac{1}{2^n} \sum_{M \subseteq [n]} \varepsilon_q \left(\bigoplus_{j \in M} G_j \right).$$

Now, to complete the proof of Theorem 18, using Theorem 17, we have

$$\begin{aligned}\frac{1}{2^n}\sum_{M\subseteq[n]}\varepsilon_q\left(\bigoplus_{j\in M}G_j\right) &= \frac{1}{2^n}\sum_{M\subseteq[n]}\prod_{j\in M}\varepsilon_q(G_j)\\ &= \prod_{j=1}^{n}\left(\frac{1+\varepsilon_q(G_j)}{2}\right)=\prod_{j=1}^{n}\omega_q(G_j).\end{aligned}$$

Combining this with above, we conclude that

$$\omega_q\left(\bigotimes_{j=1}^{n}G_j\right)=\prod_{j=1}^{n}\omega_q(G_j)$$

which completes the proof of Theorem 18.

6.4.3 A counterexample to perfect parallel repetition

In this subsection, we give an unpublished proof due to Watrous [Wat04] that there exists a binary game G (that is not an XOR game) for which $\omega_q(G)=\omega_q(G\otimes G)=\frac{2}{3}$. The game used was originally proposed by Feige and Lovász [FL92], who showed that $\omega_c(G)=\omega_c(G\otimes G)=\frac{2}{3}$.

The game has binary questions ($S=T=\{0,1\}$) and binary answers ($A=B=\{0,1\}$). The operation of the game is as follows. The verifier selects a pair of questions (s,t) uniformly from $\{(0,0),(0,1),(1,0)\}$ and sends s and t to Alice and Bob, respectively. The verifier accepts the answers, a from Alice and b from Bob, if and only if $s\vee a\neq t\vee b$.

Consider a quantum strategy for this game, where $|\psi\rangle\in\mathcal{X}\otimes\mathcal{Y}$ is the shared entanglement, Alice's behavior is determined by the Hermitian unitary operators A_0 and A_1, and Bob's behavior is determined by the Hermitian unitary operators B_0 and B_1. On input (s,t), Alice computes a by measuring with respect to A_s, and Bob computes b by measuring with respect to B_t. It is straightforward to deduce that the bias of this strategy is

$$\frac{1}{3}\langle\psi|\left(-A_0\otimes B_0+A_0\otimes\mathbb{I}_{\mathcal{Y}}+\mathbb{I}_{\mathcal{X}}\otimes B_0.\right)|\psi\rangle$$

Once A_0 and B_0 are determined, the optimal bias is the largest eigenvalue of M, where

$$M=-\frac{1}{3}A_0\otimes B_0+\frac{1}{3}A_0\otimes\mathbb{I}_{\mathcal{Y}}+\frac{1}{3}\mathbb{I}_{\mathcal{X}}\otimes B_0.$$

One can show that

$$M^2=-\frac{2}{3}M+\frac{1}{3}\mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}}$$

from where we conclude that the largest eigenvalue of M is $1/3$. Since the largest eigenvalue is clearly the optimum bias of the game, we conclude that $\omega_q(G) = 2/3$. Combining this with the fact that $2/3 = \omega_c(G \otimes G) \leq \omega_q(G \otimes G) \leq \omega_q(G)$, we obtain $\omega_q(G \otimes G) = 2/3$.

6.5 Feigi-Lovász relaxations and XOR games

In their seminal paper, Feigi and Lovász [FL92] showed that computing the classical value of a two-prover one-round game is equivalent to optimizing a quadratic programming problem. The optimization problem can be cast as optimizing a linear objective function subject to affine constraints and a non-convex rank constraint. Relaxing the non-convex constraint leads to the first semidefinite programming relaxation for the classical value of the two-prover one-round game. In the same paper, they considered another semidefinite programming relaxation, which satisfies a nice multiplicative property as discussed later in this section.

For any game G, the optimum value of the first relaxation (given by equations (5)-(9) in their paper) is denoted $\sigma(G)$ and the optimum value of the second relaxation (given by equations (12)-(17) in the same paper) is denoted $\bar{\sigma}(G)$. The semidefinite programming relaxations are such that the set of feasible solutions of the first relaxation is a subset of the set of feasible solutions of the second relaxation and hence $\sigma(G) \leq \bar{\sigma}(G)$. We write both the semidefinite relaxations for the special case of XOR games below. The general case have the same set of constraints as shown below, however, with different answer sets A and B of the provers.

First of all, let C be the matrix with rows indexed by $(s, a) \in S \times \{0, 1\}$, columns indexed by $(t, b) \in T \times \{0, 1\}$, and the matrix entries $C[(s, a), (t, b)] = \pi(s, t)V(a, b|s, t)$. We refer to C as the *cost matrix* of the XOR game and

$$\hat{C} = \frac{1}{2} \begin{pmatrix} 0 & C \\ C^\top & 0 \end{pmatrix}.$$

to be the *symmetric cost matrix* of the XOR game. The first semidefinite programming relaxation that appears in Ref. [FL92] is as follows. Let the optimum value of the relaxation be denoted $\sigma(G)$.

$$\begin{aligned} \sigma(G) = \text{maximize:} \quad & \left\langle \hat{C}, P \right\rangle \\ \text{subject to:} \quad & \forall s, t \in S \cup T, \quad \sum_{a,b \in \{0,1\}} P[(s, a), (t, b)] = 1, \qquad (6.7) \\ & \forall s, t \in S \cup T, a, b \in \{0, 1\}, \quad P[(s, a), (t, b)] \geq 0, \qquad (6.8) \\ & P \succeq 0. \end{aligned}$$

The second semidefinite programming relaxation is as follows. Let the optimum value of the relaxation be $\bar{\sigma}(G)$.

$$\begin{aligned}
\bar{\sigma}(G) = \text{maximize:} \quad & \left\langle \hat{C}, P \right\rangle \\
\text{subject to:} \quad & \forall s,t \in S \text{ or } s,t \in T, \quad \sum_{a,b \in \{0,1\}} \left| P[(s,a),(t,b)] \right| \leq 1, \qquad (6.9) \\
& \forall s,t \in S \cup T, a,b \in \{0,1\}, \quad P[(s,a),(t,b)] \geq 0, \qquad (6.10) \\
& P \succeq 0.
\end{aligned}$$

We have the following theorem that relates the quantum value of an XOR game with the optimum values of the two SDP relaxations. We remark that it is not known if any of the three quantities discussed are related to each other for general two-prover one-round games.

Theorem 22. *For any XOR game G, $\omega_q(G) = \sigma(G) = \bar{\sigma}(G)$.*

Proof. Let G be an XOR game. We already know that $\sigma(G) \leq \bar{\sigma}(G)$. Hence it suffices to show that $\omega_q(G) \leq \sigma(G)$ and $\bar{\sigma}(G) \leq \omega_q(G)$. We first show that $\omega_q(G) \leq \sigma(G)$. To begin let us assume an optimal strategy for G as follows. The provers Alice and Bob share a maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{k}} \sum_{i \in [k]} |i\rangle |i\rangle \in \mathcal{X} \otimes \mathcal{X}$$

and upon questions $s \in S$ and $t \in T$, the provers apply projective measurements described by operators $\{\Pi_s^0, \Pi_s^1\}$ and $\{\Delta_t^0, \Delta_t^1\}$, respectively. Let

$$u_s^a = \begin{cases} (\Pi_s^a \otimes \mathbb{I}_{\mathcal{X}}) |\psi\rangle & \text{for } s \in S \\ (\mathbb{I}_{\mathcal{X}} \otimes \Delta_s^a) |\psi\rangle & \text{for } s \in T \end{cases}$$

and let $\hat{P}$ be a positive semidefinite matrix with entries $\hat{P}[(s,a),(t,b)] = \left\langle u_s^a, u_t^b \right\rangle$. Since

$$\Pi_s^0 + \Pi_s^1 = \mathbb{I}_{\mathcal{X}} \qquad \text{and} \qquad \Delta_t^0 + \Delta_t^1 = \mathbb{I}_{\mathcal{X}}$$

it is straightforward to check that (6.7) holds. Moreover, for positive semidefinite operators Π_s^a and Δ_t^b, we have for any $(s,t) \in S \times T$

$$\hat{P}[(s,a),(t,b)] = \left\langle u_s^a, u_t^b \right\rangle = \langle \psi | \, \Pi_s^a \otimes \Delta_t^b \, |\psi\rangle \geq 0.$$

The above set of inequalities takes care of the non-negativity constraints of the off-diagonal

blocks of $\hat{P}$ (corresponding to matrix C). Given that $|\psi\rangle$ is the maximally entangled state, we also have

$$\hat{P}[(s,a),(t,b)] = \left\langle u_s^a, u_t^b \right\rangle = \langle\psi| I_{\mathcal{X}} \otimes \Delta_s^a \Delta_t^b |\psi\rangle = \frac{1}{k} \left\langle \Delta_s^a, \Delta_t^b \right\rangle \geq 0$$

for any $s, t \in T$. Similarly for any $s, t \in S$,

$$\hat{P}[(s,a),(t,b)] = \left\langle u_s^a, u_t^b \right\rangle = \langle\psi| \Pi_s^a \Pi_t^b \otimes I_{\mathcal{X}} |\psi\rangle = \frac{1}{k} \left\langle \Pi_s^a, \Pi_t^b \right\rangle \geq 0.$$

The above two sets of inequalities take care of the non-negativity constraints of the diagonal blocks of $\hat{P}$. Hence $\hat{P}[(s,a),(t,b)] \geq 0$ for every choice of $s, t \in S \cup T$ and $a, b \in \{0,1\}$, which shows that $\hat{P}$ satisfies the constraints described in (6.8). It is fairly straightforward to observe that the objective value of $\hat{P}$ is equal to the quantum value of the XOR game, which implies that

$$\omega_q(G) \leq \sigma(G).$$

We now proceed to establish that $\bar{\sigma}(G) \leq \omega_q(G)$. To begin, assume an optimal solution $\bar{P}$ for the semidefinite program with optimum value $\bar{\sigma}(G)$. Since $\bar{P}$ is a positive semidefinite matrix, we can find vectors u_s^a for $s \in S$ and $a \in \{0,1\}$, and v_t^b for $t \in T$ and $b \in \{0,1\}$, such that

$$\bar{P}[(s,a),(t,b)] = \begin{cases} \left\langle u_s^a, u_t^b \right\rangle & s,t \in S \\ \left\langle v_s^a, v_t^b \right\rangle & s,t \in T \\ \left\langle u_s^a, v_t^b \right\rangle & s \in S, t \in T. \end{cases}$$

We can view $\{u_s^a : a \in \{0,1\}\}$ as Alice's collection of vectors for each question $s \in S$ and $\{v_t^b : b \in \{0,1\}\}$ as Bob's collection of vectors for each question $t \in T$. From (6.9), we have that

$$\sum_{a,b \in \{0,1\}} \left| \left\langle u_s^a, u_s^b \right\rangle \right| \leq 1$$

and hence

$$\left| \sum_{a,b \in \{0,1\}} \left\langle u_s^a, u_s^b \right\rangle \right| = \left\| \sum_{a \in \{0,1\}} u_s^a \right\|^2 \leq 1.$$

Therefore $u_s^0 + u_s^1$ and similarly $v_t^0 + v_t^1$ lie in a unit ball. We have

$$\left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle \leq 1$$

Let $x_s = u_s^0 - u_s^1$ and $y_t = v_t^0 - v_t^1$. Then

$$\langle x_s, y_t \rangle = \left\langle u_s^0, v_t^0 \right\rangle - \left\langle u_s^0, v_t^1 \right\rangle - \left\langle u_s^1, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle.$$

A straightforward calculation shows that the above two equations imply the following two inequalities:

$$\left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle \leq \frac{1 - \langle x_s, y_t \rangle}{2} \qquad \text{and} \qquad \left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle \leq \frac{1 + \langle x_s, y_t \rangle}{2}. \quad (6.11)$$

From (6.11), $\bar{\sigma}(G)$ is upper bounded by

$$\sum_{s,t} \pi(s,t) \frac{1}{2} \begin{cases} (1 + \langle x_s, y_t \rangle) & \text{if the correct answer is 0} \\ (1 - \langle x_s, y_t \rangle) & \text{if the correct answer is 1} \end{cases}$$

which is at most $\omega_q(G)$ (Proposition 5.7 in Cleve, Høyer, Toner and Watrous [CHTW04a]). Hence $\bar{\sigma}(G) \leq \omega_q(G)$, which completes the proof.

□

Feige and Lovász demonstrated that the second relaxation is multiplicative in the following sense. If the optimum values of the second semidefinite programming relaxation for two games G_1 and G_2 are $\bar{\sigma}(G_1)$ and $\bar{\sigma}(G_1)$, respectively, then the optimum value of the same relaxation for the game $G_1 \otimes G_2$ is exactly $\bar{\sigma}(G_1)\bar{\sigma}(G_2)$. Combining our Theorem 18 and the multiplicative property of the second relaxation, we have the following simple observation.

Proposition 23. *For any XOR games $G_1, \ldots, G_n$,*

$$w_q \left(\bigotimes_{i=1}^{n} G_i \right) = \sigma \left(\bigotimes_{i=1}^{n} G_i \right) = \bar{\sigma} \left(\bigotimes_{i=1}^{n} G_i \right).$$

This automatically implies that the first semidefinite programming relaxation is also multiplicative for XOR games in the sense as described above. For a larger class of games such as the ones discussed in the next chapter, we do not know whether the above proposition holds.

We finally remark that in the journal version of their paper, Kempe, Regev, and Toner [KRT10] use the same approach to show multiplicative property of a large class of semidefinite programs that subsumes our proof of the additivity theorem.

6.6 Related work

In this section, we briefly mention few further developments related to the results and techniques used in this chapter.

A consequence of our result and an earlier result on relationship between classical and quantum XOR games [CHTW04a] also gives a parallel repetition theorem for classical XOR games as follows.

Theorem 24. *Let G be an XOR game with classical value $1-\epsilon$. Then the classical value of the game $G^{\otimes n}$ is at most $(1-\Omega(\epsilon^2))^n$.*

This bound was reported in a paper by Feige, Kindler, and O'Donnell [FKO07] who attributed it to Feige and Lovász [FL92], and Goemans and Williamson [GW95]. This was subsequently improved by Rao [Rao08] for a large class of games, called *projection games* that includes XOR games as well. Rao's upper bound shows a quadratic dependence on ϵ. It is natural to ask whether the dependence of $\omega_c\left(G^{\otimes n}\right)$ on ϵ is tight or not. What we know about classical XOR games is that they do not obey a perfect parallel repetition theorem, the *CHSH* game being a counterexample to this fact, but one can hope of improving the dependence on $\omega_c\left(G^{\otimes n}\right)$ on ϵ from quadratic to let's say linear. However, Raz [Raz11] showed that for the odd cycle game of size m, which is an example of XOR games (refer to the papers [CHTW04a, Raz11] for a description of an odd cycle game),

$$\omega_c\left(G_m^{\otimes n}\right) \geq 1 - \frac{1}{m} O\left(\sqrt{n}\right).$$

Here G_m denotes the odd cycle game of size m. It is easy to establish that $\omega_c(G_m) = 1 - 1/2m$. Consequently, the above relationship implies that one needs $O(m^2)$ repetitions to reduce the classical value of the game to a constant, thereby showing that the upper bound on $\omega_c(G^{\otimes n})$ as mentioned in Theorem 24 is tight. In other words, one cannot hope for a better dependence of ϵ in the above theorem.

The measure of how well the provers perform is defined to be the maximum average success probability of the provers under a predefined distribution. One can also define another measure of how well the provers perform by considering *worst-case success probability* of an XOR game. By Tsirelson's vector characterization, the worst-case bias in the quantum case is the optimum of the following semidefinite program as described below.

Let $S = [N]$ and $T = [M]$ be the question sets, $R \subseteq S \times T$ to be the set of valid questions asked to the provers, and A be the cost matrix defined as $A[s,t] = (-1)^{f(s,t)}$ if f is defined on the pair (s,t) and zero otherwise. Note that the distribution over the question set is not important when considering the worst-case value (or worst-case bias) of the game.

$$\begin{aligned}
\text{maximize:} \quad & \eta \\
\text{subject to:} \quad & \forall (i,j) \in R, \qquad X[i, N+j] \geq (-1)^{f(i,j)} \eta, \\
& \forall k \in [N+M], \qquad X[k,k] = 1, \\
& X \in \operatorname{Pos}\left(\mathbb{R}^{N+M}\right).
\end{aligned}$$

It holds that the worst-case bias of an XOR game is multiplicative in the sense of Theorem 17 and one can easily show that the worst-case quantum value of a conjunction of XOR games $G_1, G_2, \ldots, G_n$ is the product of the worst-case quantum values of the individual games. It also holds that the optimal dual solution of the semidefinite program described above gives us the "hardest distribution" for playing an XOR game. That is, if the XOR game is played using this distribution, then the verifier can minimize the success probability of the provers, where the minimum is taken over all possible choices of the distribution.

The multiplicative property of the worst-case bias of an XOR game was first observed by Lee, Shraibman and Špalek [LSS08] in connection with obtaining direct product theorem for discrepancy. They considered the quantity $\gamma_2^\alpha(A)$ defined by Linial and Shraibman [LS09] (the preliminary version of their paper appeared in 2007) and observed that when $\alpha \to \infty$, the quantity approaches the reciprocal of the worst-case bias of an XOR game corresponding to matrix A. These quantities are related to the one-way communication complexity models and we refer to the Refs. [LMSS07, LS09] for more details.

Another avenue explored following this result was the question of multiplicative property of semidefinite programs. In particular, Mittal and Szegedy [MS07] investigated the following questions. For two semidefinite programs ($i \in \{1, 2\}$)

$$\alpha_i := \max\left\{ \langle C_i, X_i \rangle : \left\langle A_i^{(k)}, X_i \right\rangle = b[k] \text{ for all } k \in [m_i] \quad \text{and} \quad X_i \in \operatorname{Pos}\left(\mathbb{R}^{N_i}\right) \right\} \tag{6.12}$$

what are the conditions on them such that the optimum value of the following semidefinite program

$$\max\left\{ \langle C_1 \otimes C_2, X \rangle : \left\langle A_1^{(j)} \otimes A_2^{(k)}, X \right\rangle = b[j] \cdot b[k] \text{ for all } (j,k) \in [m_1] \times [m_2] \right\} \tag{6.13}$$

and

$$X \in \operatorname{Pos}\left(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}\right)$$

is the product of α_1 and α_2? They showed that if $C_i \in \operatorname{Pos}\left(\mathbb{R}^{N_i}\right)$ for $i \in \{1, 2\}$, then the multiplicative property holds. The other case for which they showed that the multiplicative property holds is when the matrices for both semidefinite programs are of the following form:

$$C_i = \begin{pmatrix} 0 & R_i \\ R_i^\top & 0 \end{pmatrix} \qquad \text{and} \qquad A_i^{(j)} = \begin{pmatrix} P_i^{(j)} & 0 \\ 0 & Q_i^{(j)} \end{pmatrix}.$$

Note that quantum bias of an XOR game has a semidefinite program of the following form and although we analyze a different product operation on the semidefinite program, it holds that if the semidefinite programs in (6.12) represent the quantum bias of XOR games G_1 and G_2, then the optimum of the SDP in (6.13) is equal to $\varepsilon_q(G_1 \oplus G_2)$ [CSUU08, KRT10].

Finally, Slofstra [Slo10] answered a question on the amount of entanglement needed to play an XOR game optimally or near-optimally. Although the result is not directly related to this chapter, we briefly mention that he showed that for certain XOR game, the only optimum strategy is to use the Tsirelson's construction of Hermitian unitary operators from the vector system. In particular, he obtained that there exists XOR games wherein the provers need exponential number of entangled qubits for optimal performance. We refer to his paper for further details.

Chapter 7

Quantum unique games and parallel repetition

This chapter is concerned with parallel repetition theorems for unique games and is based on the Ref. [Upa07]. The work presented in this chapter was obtained independently of Kempe, Regev, and Toner [KRT10] (a preliminary version of their paper appeared in 2007).

In this chapter, we introduce a class of games that includes XOR games and show a parallel repetition theorem for such games when the provers are allowed to share *a priori* entanglement. These games are called *unique games* and have been widely studied in the literature because of their connection with hardness of approximation. Unlike XOR games, unique games are not known to have a semidefinite programming characterization in the quantum world. However, we consider a semidefinite programming relaxation to quantum unique games and use the *quantum rounding procedure* presented in the Ref. [KRT10] to round an optimal solution of the relaxation to a valid quantum strategy. The relaxation is such that the multiplicative property holds, which easily translates into a parallel repetition theorem for unique games.

The organization of the chapter is as follows.

- In Section 7.1, we introduce unique games, their classical and quantum variant, briefly touch upon the huge body of work devoted on them, and state the main results of the chapter.
- In Section 7.2, we discuss two semidefinite programming relaxations for quantum unique games and the quantum rounding procedure. The first relaxation and the rounding procedure appear in the Ref. [KRT10]. The main focus of this chapter is on the second relaxation, which we use to prove parallel repetition theorem for unique games [Upa07].

- In Section 7.3, we show a class of semidefinite programs that are multiplicative under a product operation similar to the one discussed in Section 6.4.1. The class of semidefinite programs considered is a subclass of semidefinite programs mentioned in Lee and Mittal [LM08].
- In Section 7.4, we prove the parallel repetition theorem for unique games and provide even stronger parallel repetition theorem for certain class of unique games known as MOD_k games.
- In Section 7.5, we discuss research works related to the main results and techniques used in this chapter. A detailed comparison of our result with the Ref. [KRT10] is done in this section.

7.1 Unique games, classical and quantum provers

Two-prover one-round games: Two-prover one-round games with classical provers have been central to many important discoveries in computational complexity theory. Such games are operationally defined as follows.

- The verifier generates a pair of questions $(s,t) \in S \times T$ according to a probability distribution $\pi : S \times T \to [0,1]$ and sends s to the first prover (Alice) and t to the second prover (Bob).
- Upon receiving s and t, Alice and Bob reply with their respective answers $a \in A$ and $b \in B$.
- The verifier evaluates a predicate $V : A \times B \times S \times T \to \{0,1\}$ based on the questions send and answers received, and accepts if and only if $V(a,b|s,t) = 1$.

The objective of the provers is to maximize verifier's acceptance probability. The maximum average success probability of the provers over all possible (and valid) classical strategy is called the *classical value* of the game.

The focus of the current chapter is on unique games. Unique games are a class of two-prover one-round games wherein the verifier's predicate evaluates to 1 if and only if $a = \Gamma_{st}(b)$ for some permutation Γ_{st} for every choice of questions (s,t). Before we focus on unique games with quantum provers, we briefly mention the relevance of two-prover one-round games in computational complexity theory.

Various NP-complete problems such as satisfiability of a 3SAT formula can be cast as an instance of two-prover one-round games. The existence of a satisfiable assignment for a

3SAT formula is equivalent to determining whether classical value of a certain two-prover one-round game is 1 or not. Therefore, computing the value of the game is clearly NP-hard. A more relaxed version of the problem is the following problem: Given the promise that either the classical value of the game is exactly 1 or a small constant $\epsilon > 0$, what is the complexity of distinguishing between the two cases. A consequence of the famous PCP Theorem [ALMSS98, AS98] and Raz's parallel repetition theorem [Raz98] is that it is NP-hard to determine whether the classical value of a game is 1 or at most $\epsilon > 0$. The class of games that arise from the PCP Theorem is known as *projection games*. These games satisfy the following two properties.

1. The answer sets of the provers are $[k] := \{1, \ldots, k\}$ and $[l] := \{1, \ldots, l\}$, where k, l depend onϵ.

2. Conditioned on verifier's acceptance, the answer of the first prover uniquely determines the answer of the second prover. This property is referred to as *projection property*.

The hardness of such games have led to optimal inapproximability results for problems like MAX-3SAT and MAX-3LIN2 [Hås01]. Khot [Kho02] investigated a restricted class of projection games, where the projection property holds both ways. In other words, conditioned on verifier's acceptance the answer of the first prover uniquely determines the answer of the second prover and vice-versa.[1] This implies that for every choice of question pair asked to the provers, there is a bijective mapping between the answer sets of the two prover. Note that the bijective mapping may depend on the question pair. Such games are known as *unique games*. An example of a unique game is the CHSH game mentioned in the previous chapter, or more generally, any XOR game. One of the major open problems in the theory of computational complexity is the Unique Games Conjecture of Khot [Kho02] as described below.

Conjecture 25. *For any constants $\varepsilon, \delta > 0$, there exists a unique game such that determining whether the classical value of the game is at least $1 - \varepsilon$ or at most δ is* NP*-hard.*

Note that for any unique game, determining whether the classical value of the game is 1 can be easily done in polynomial-time in the number of the questions. In the past decade, unique games have found applications in hardness of approximation of various problems. A considerable amount of work has been devoted in either proving optimal inapproximability results for various optimization problems assuming the conjecture [KKMO07, KR08, Rag08], or refuting the conjecture [CMM06a, CMM06b, Tre08]. The uncertainty around

[1] A similar notion of *uniqueness* was investigated by Feige and Lovász [FL92], and Cai, Condon and Lipton [CCL90, CCL92] for two-prover one-round games.

the conjecture has been a notoriously difficult problem in theoretical computer science. A few research work has also focused on coming up with the best possible algorithm (not necessarily polynomial-time) that distinguishes between the two cases [ABS10, Kol10].

In the quantum case, Kempe, Regev and Toner [KRT08] showed that one can efficiently approximate the *quantum value* (defined below) of such games in the following sense. If the optimum value of a semidefinite programming relaxation for quantum value of unique games is $1 - \epsilon$, then there exists a valid quantum strategy of the provers that succeeds with probability at least $1 - 6\epsilon$. One of the consequences of their result is that one cannot hope for NP-hardness of unique games with quantum provers. In the classical case, similar techniques of rounding a SDP solution to a valid classical strategy have been used by various researchers. However, the rounding solution does not provide a strategy with "good enough" classical value to refute the conjecture. In fact, Khot and Vishnoi [KV05] explicitly constructed a unique game where the optimum of the SDP is close to one, while the value of the game is close to zero.

We now proceed to explain what we mean by a quantum strategy for unique games. Any quantum strategy of the provers for a unique game G can be described in terms of a shared quantum state and measurement for each pair of questions (s, t) as follows.

- The quantum provers (sometimes referred to as Alice and Bob) share a bipartite quantum state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{X}$.
- Upon receiving questions s and t from the verifier, Alice and Bob apply projective measurements described by the operators $\{\Pi_s^a : a \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ and $\{\Delta_t^b : b \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$. The provers send the outcome of the measurement operation to the verifier.

The quantum value of the game for the strategy described above is the average success probability of the provers when they adopt the strategy and is given by the expression

$$\sum_{s,t} \sum_{a,b} \pi(s,t) V(a,b|s,t) \langle\psi| \Pi_s^a \otimes \Delta_t^b |\psi\rangle .$$

The quantum value of the game $\omega_q(G)$ is the best possible average success probability achieved over all valid quantum strategies as described above. In mathematical terms,

$$\omega_q(G) = \lim_{\dim(\mathcal{X}) \to \infty} \max_{|\psi\rangle \in \mathcal{X} \otimes \mathcal{X}} \max_{\Pi_s^a, \Delta_t^b} \sum_{s,t} \sum_{a,b} \pi(s,t) V(a,b|s,t) \langle\psi| \Pi_s^a \otimes \Delta_t^b |\psi\rangle . \tag{7.1}$$

Finally, the n-fold repetition of a unique game G with questions sets S and T, answer set $[k]$ for each prover, and a probability distribution $\pi : S \times T \to [0, 1]$ proceeds as follows.

- The verifier generates n pairs of questions $\{(s_i, t_i) \in S \times T : i \in [n]\}$ independently according to the probability distribution π and sends the n-tuple $(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n)$ to Alice and Bob, respectively.
- Upon receiving the questions, Alice replies with $(a_1, \ldots, a_n)$ and Bob replies with $(b_1, \ldots, b_n)$.
- The verifier accepts if and only if the provers win all instances of the game. That is the predicate $V(a_i, b_i|s_i, t_i)$ evaluates to 1 for all $i \in [n]$.

We denote the n-fold repetition of games by $G^{\otimes n}$. A fundamental question in the theory of two-prover one-round games is the rate at which the value of a game decreases if multiple instances of the game are played simultaneously. For a while it was believed that playing n instances of the game simultaneously would decrease the value of the game from p to p^n, where p is the value of the original game. As mentioned in the previous chapter, Feige and Lovász [FL92] showed that there exists a game with classical value $\frac{2}{3}$ such that playing it twice in parallel does not decrease the classical value. The same game has been used to demonstrate that perfect parallel repetition does not hold for quantum strategies [Wat04] and non-signaling strategies [Hol07]. Subsequent to their counterexample, many results were established on parallel repetition of various types of games (see the references [CCL90, CCL92, FV02, Ver95, Ver96]) culminating in Raz's parallel repetition theorem [Raz98].

Theorem 26. *For any game G with classical value $\omega_c(G) = 1 - \epsilon$ and answer sets A and B, there exists universal constants c_1 and c_2 such that the classical value of the game $G^{\otimes n}$ is*

$$\omega_c\left(G^{\otimes n}\right) \leq \left(1 - \frac{\epsilon^{c_1}}{c_2}\right)^{n/\log(|A||B|)}.$$

Feige and Verbitsky [FV02] showed that Raz's parallel repetition theorem is almost tight in the following sense. They showed that there exists a family of games for which reducing the classical value of the game from one constant to another fixed constant requires $O\left(\frac{\log k}{\log\log k}\right)$ repetitions, where k is the size of the answer sets. Subsequently, Holenstein [Hol07] gave a simplified proof of Raz's parallel repetition theorem and showed that the constant c_1 in Theorem 26 is 3.

Although Raz's and Holenstein's proof give sufficiently strong quantitative bounds on the classical value of $G^{\otimes n}$ in terms of classical value of G, there have been renewed interest in obtaining better bounds in parallel repetition theorem for unique games in connection with Khot's unique games conjecture [AS08, BHHRRS08, BRRRS09, FKO07]. Shortly after Holenstein's simplified proof, Rao [Rao08] improved the generic bounds due to Raz and Holenstein for the class of projection games.

Theorem 27. *For any projection game G with classical value $\omega_c(G) = 1 - \epsilon$ and answer sets $[k]$ and $[l]$, there exists a universal constant c such that the classical value of the game $G^{\otimes n}$ is*

$$\omega_c\left(G^{\otimes n}\right) \leq \left(1 - \frac{\epsilon^2}{c}\right)^n.$$

As briefly discussed in Section 6.6, Raz [Raz11] showed that there exists an XOR game for which Rao's bound is tight wioth respect to the dependence on ϵ. Barak, Rao, Raz, Rosen, and Shaltiel [BRRRS09] have recently shown that if the projection game has the property that the probability distribution of the question pair is a product distribution, then one can obtain an even stronger parallel repetition theorem in the sense that the dependence on ϵ is linear. A similar bound is known to hold for projection games on bipartite expanders [RR10].

In the quantum case, prior to the work presented in this chapter, a parallel repetition theorem was known only for XOR games [CSUU08]. This has already been discussed in the previous chapter. In this chapter, we establish a parallel repetition theorem for unique games, which is based on the manuscript [Upa07]. Independent of the results presented here, Kempe, Regev, and Toner [KRT08] showed a parallel repetition theorem for unique games.

Results: We now proceed to list the main results of this chapter. Our first result is for general unique games.

Theorem 28. *For any unique game G with quantum value $\omega_q(G) = 1 - \epsilon$, we have*

$$\omega_q(G^{\otimes n}) \leq \left(1 - \frac{\epsilon^2}{49}\right)^n.$$

The bound obtained for unique games has been proven to be essentially tight [KR10] with respect to the dependence on ϵ. A very special case of unique games are XOR games. For any XOR game, we show that its quantum value is equal to the optimum value of the SDP relaxations referred to in Section 7.2. This gives an alternate proof of the perfect parallel repetition theorem for XOR games. For a class of games that generalize XOR games, which we call as MOD_k games, an essentially tight parallel repetition theorem holds. The MOD_k games are generalization of XOR games wherein the provers succeed if and only if their answers a and b satisfy the linear equation $a - b \equiv f(s,t) \mod k$.

Theorem 29. *For any MOD_k game G with quantum value $\omega_q(G) = 1 - \epsilon$, we have*

$$\omega_q(G^n) \leq \left(1 - \frac{\epsilon}{4}\right)^n.$$

Our technique for proving the results is as follows. We consider a semidefinite programming relaxation for quantum value of a two-prover one-round game and show that it is multiplicative in the following sense. If the optimum of the SDP corresponding to a game G is α, then the optimum of the SDP corresponding to the n-fold repetition of G is α^n. For a given unique game, we use the the quantum rounding procedure of Ref. [KRT10]to obtain a quantitative relationship between the optimum value of the semidefinite programming relaxation and the quantum value. This immediately translates into a parallel repetition theorem for such games.

7.2 SDP relaxations and quantum rounding

In this section, we discuss the two semidefinite programming relaxations for the quantum value of a unique game and the quantum rounding procedure. The relaxations presented here are relaxations for any two-prover one-round quantum game. However, we limit our exposition to quantum unique games.

7.2.1 Semidefinite programming relaxations

For a quantum strategy described by a shared quantum state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{X}$, and projective measurements $\{\Pi_s^a : a \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ and $\{\Delta_t^b : b \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ for every choice of s and t, the quantum value of the associated strategy is given by the expression

$$\sum_{s,t} \sum_{a,b} \pi(s,t) V(a,b|s,t) \langle\psi| \Pi_s^a \otimes \Delta_t^b |\psi\rangle .$$

We first observe the following three constraints that arise from the associated measurement operators.

1. Since the sets $\{\Pi_s^a : a \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ and $\{\Delta_t^b : b \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ are measurement operators, it holds that

$$\sum_{a \in [k]} \Pi_s^a = \mathbb{I}_{\mathcal{X}} \qquad \text{and} \qquad \sum_{b \in [k]} \Delta_t^b = \mathbb{I}_{\mathcal{X}}.$$

   for every choice of $(s,t) \in S \times T$.
2. Since the elements of the sets $\{\Pi_s^a : a \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ and $\{\Delta_t^b : b \in [k]\} \subset \mathrm{Pos}(\mathcal{X})$ are projection operators, it holds that

$$\left\langle \Pi_s^a, \Pi_s^{a'} \right\rangle = 0 \qquad \text{and} \qquad \left\langle \Delta_t^b, \Delta_t^{b'} \right\rangle = 0$$

for every choice of $(s,t) \in S \times T$ such that $a \neq a'$ and $b \neq b'$.

3. Since the expression $\langle\psi| \Pi_s^a \otimes \Delta_t^b |\psi\rangle$ describes the probability of outputting (a,b) on questions (s,t), it holds that
$$\langle\psi| \Pi_s^a \otimes \Delta_t^b |\psi\rangle \geq 0$$
for every choice of $(s,t) \in S \times T$ and $a, b \in [k]$.

For every choice of operators Π_s^a and Δ_t^b, define the vectors u_s^a and v_t^b as

$$u_s^a := \left(\Pi_s^a \otimes \mathbb{I}_{\mathcal{X}}\right) |\psi\rangle \qquad \text{and} \qquad v_t^b := \left(\mathbb{I}_{\mathcal{X}} \otimes \Delta_t^b\right) |\psi\rangle .$$

It follows from item 1 that

$$\sum_{a \in [k]} u_s^a = |\psi\rangle \qquad \text{and} \qquad \sum_{b \in [k]} v_t^b = |\psi\rangle$$

and from item 2 that

$$\left\langle u_s^a, u_s^{a'} \right\rangle = 0 \qquad \text{and} \qquad \left\langle v_t^b, v_t^{b'} \right\rangle = 0.$$

Moreover, item 3 implies that $\left\langle u_s^a, v_t^b \right\rangle \geq 0$. These constraints can be written down as the following semidefinite program [KRT10], which is denoted $\mathrm{SDP}_1(G)$. Let $\sigma_q(G)$ be the optimum value of the SDP.

$$\begin{aligned}
\sigma_q(G) = \text{maximize:} \quad & \sum_{s,t} \sum_{a,b} \pi(s,t) V(a,b|s,t) \left\langle u_s^a, v_t^b \right\rangle \\
\text{subject to:} \quad & ||z|| = 1, \\
& \forall s,t, \quad \sum_{a \in [k]} u_s^a = \sum_{b \in [k]} v_t^b = z, \\
& \forall s,t,a,b : a \neq b, \quad \left\langle u_s^a, u_s^b \right\rangle = \left\langle v_t^a, v_t^b \right\rangle = 0, \\
& \forall s,t,a,b, \quad \left\langle u_s^a, v_t^b \right\rangle \geq 0.
\end{aligned}$$

We note that the non-convex constraints

$$\sum_{a \in [k]} u_s^a = \sum_{b \in [k]} v_t^b = z \qquad \text{and} \qquad ||z|| = 1$$

can be converted into following linear constraints

$$\sum_{a \in [k]} \langle u_s^a, u_s^a \rangle = \sum_{b \in [k]} \left\langle v_t^b, v_t^b \right\rangle = \sum_{a,b} \left\langle u_s^a, v_t^b \right\rangle = 1.$$

Throughout the chapter, we will let s and t to be the elements of the sets S and T, respectively. Without loss of generality, we can assume the vectors associated with a feasible solution of $\mathrm{SDP}_1(G)$ to be real vectors [KRT10]. This fact implicitly helps in analyzing the quantum rounding procedure.

For MOD_k games, there always exists an optimal strategy in which the provers' output distribution is uniform on the answer set. This allows us to put an extra constraint on the SDP for such games:

$$\forall s,t,a,b, \quad ||u_s^a|| = ||v_t^b|| = \frac{1}{\sqrt{k}}. \tag{7.2}$$

The existence of an optimal solution satisfying (7.2) is proved in Proposition 38. Note that there may be other unique games for which an optimal quantum strategy satisfies (7.2). However, one cannot guarantee that the n-fold repetition of the game also satisfies the same property.

It is known that the optimum value of $\mathrm{SDP}_1(G)$ is a very good approximation to the quantum value of a unique game. One might be tempted to use it to obtain parallel repetition results similar to XOR games. Unfortunately, it is not true that the optimum value of $\mathrm{SDP}_1\left(G^{\otimes n}\right)$ is $(\sigma_q(G))^n$. Hence one needs to come up with a suitable SDP relaxation for which this property holds and the optimum value is still a good enough approximation to the quantum value of the unique game. One such semidefinite program is given below, which is denoted $\mathrm{SDP}_2(G)$. Let $\bar{\sigma}_q(G)$ be the optimum value of the SDP.

$$\begin{aligned}
\bar{\sigma}_q(G) = \text{maximize:} \quad & \sum_{s,t}\sum_{a,b} \pi(s,t)V(a,b|s,t)\left\langle u_s^a, v_t^b\right\rangle \\
\text{subject to:} \quad & \forall s,t, \quad \|z_s\| = \|z_t\| = 1, \\
& \forall s,t, \quad \sum_{a\in[k]} u_s^a = z_s \quad \text{and} \quad \sum_{b\in[k]} v_t^b = z_t, \\
& \forall s,t,a,b : a \neq b, \quad \left\langle u_s^a, u_s^b\right\rangle = \left\langle v_t^a, v_t^b\right\rangle = 0, \\
& \forall s,t,a,b, \quad \left\langle u_s^a, v_t^b\right\rangle \geq 0.
\end{aligned}$$

Note that the only difference is that the system of vectors $\{u_s^a : a \in [k]\}$ and $\{v_t^b : b \in [k]\}$ may sum up to different unit vectors. One can easily see that the two semidefinite programs are relaxations of the quantum value of any two-prover one-round quantum game with answer sets $[k]$ and $[l]$. We begin with a simple lemma that establishes a qualitative relationship between the two semidefinite programs.

Lemma 30. *For any two-prover one-round game G, whether unique or not, $\sigma_q(G) = 1$ if and only if $\bar{\sigma}_q(G) = 1$.*

Proof. Since $\mathrm{SDP}_2(G)$ is a relaxation of $\mathrm{SDP}_1(G)$, it is clear that $\sigma_q(G) = 1$ implies

$\bar{\sigma}_q(G) = 1$. For the converse direction, assume that $\bar{\sigma}_q(G) = 1$ and that the system of vectors $\{u_s^a : a \in [k], s \in S\}$ and $\{v_t^b : b \in [l], t \in T\}$ form a feasible solution of $\mathrm{SDP}_2(G)$ achieving that value. Now

$$\langle z_s, z_t \rangle = \sum_{a,b} \langle u_s^a, v_t^b \rangle \geq \sum_{a,b:V(a,b|s,t)=1} \langle u_s^a, v_t^b \rangle = 1$$

and therefore $z_s = z_t = z$. Hence the same system of vectors is a feasible solution of $\mathrm{SDP}_1(G)$. Since the objective function is the same, we have the desired result.

□

7.2.2 Quantum rounding procedure

In this section, we briefly discuss the quantum rounding procedure of Kempe, Regev and Toner [KRT10]. Let the vectors obtained from any feasible solution of the semidefinite programs lie in $\mathbb{R}^M$. The dimension M can be as large as $k(|S| + |T|)$. Alternately, one can consider a large enough N and embed these vectors in $\mathbb{R}^N$. The rounding procedure begins with Alice and Bob sharing a maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle |i\rangle \in \mathcal{X} \otimes \mathcal{X}$$

where $\mathcal{X} = \mathbb{C}^M$. For every question s and t, the vectors $\{u_s^a : a \in [k]\}$ and $\{v_t^b : b \in [k]\}$ can be used to define the measurement operators of the provers as follows:

$$\Pi_s^a = \left| \frac{u_s^a}{||u_s^a||} \right\rangle \left\langle \frac{u_s^a}{||u_s^a||} \right| \qquad \text{and} \qquad \Delta_t^b = \left| \frac{v_t^b}{||v_t^b||} \right\rangle \left\langle \frac{v_t^b}{||v_t^b||} \right|.$$

The operators $\{\Pi_s^a : a \in [k]\}$ and $\{\Delta_t^b : b \in [k]\}$ not necessarily sum up to $\mathbb{I}_{\mathcal{X}}$ and so one defines

$$\Pi_s^0 = \mathbb{I}_{\mathcal{X}} - \sum_{a \in [k]} \Pi_s^a \qquad \text{and} \qquad \Delta_t^0 = \mathbb{I}_{\mathcal{X}} - \sum_{b \in [k]} \Delta_t^b$$

to be operators corresponding to outcome 0. This outcome means that the prover did not get a valid outcome. We note that

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle |i\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |w_i\rangle |w_i\rangle$$

for any set of real vectors $\{w_1, w_2, \ldots, w_M\}$ that form an orthonormal basis for $\mathbb{R}^M$. This

property together with the fact that the k-tuple of vectors arising from a feasible solution of $\mathrm{SDP}_1(G)$ or $\mathrm{SDP}_2(G)$ are orthogonal for every choice of s and t helps in coming up with a quantum strategy with a property that the success probability on questions (s,t) is *close* to the inner product of vectors that contribute to the objective value.

The provers, Alice and Bob, share unlimited copies of maximally entangled state on $\mathcal{X} \otimes \mathcal{X}$ and a sequence of real variables $\Lambda = (\lambda_1, \lambda_2, \dots)$. Each element of the sequence is independently and uniformly sampled from $[0,1]$. In round 1, Alice measures her part of the state as described by the operators $\{\Pi_s^0, \dots, \Pi_s^k\}$ above. Alice outputs a if and only if $a \neq 0$ and $\lambda_1 \leq \|u_s^a\|^2$. If one of the two conditions are not met, Alice proceeds to round 2 and repeats the same strategy but with another copy of maximally entangled state and the next real variable from Λ, which in this case is λ_2. Alice iterates the procedure unless she gives an output. Bob's strategy is same except that Bob uses projective measurements described by operators $\{\Delta_t^0, \dots, \Delta_t^k\}$ and he outputs $b \neq 0$ at round i if $\lambda_i \leq \left\|v_t^b\right\|^2$.

Although Alice and Bob cannot ensure that they output in the same round, if the optimum value of the relaxations is very high, the quantum strategy described above performs very well. This is summarized in the following lemma.

Lemma 31 (**[KRT10]**). *Let $\{u_s^a : a \in [k]\}$ and $\{v_t^b : b \in [k]\}$ be two sets of orthogonal vectors such that they sum to (possibly different) unit vectors, and $R_{st} = \{(a,b) : V(a,b|s,t) = 1\}$. Then there exists a quantum strategy such that the probability of Alice and Bob outputting $(a,b) \in R_{st}$ is at least*

$$p_{st} \geq 1 - 2\left(1 - \sum_{(a,b)\in R_{st}} \left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle^2 \left(\min\left\{||u_s^a||, ||v_t^b||\right\}\right)^2\right). \tag{7.3}$$

We use Lemma 31 to round an optimal vector solution of $\mathrm{SDP}_2(G)$ into a quantum strategy. The proof of the following lemma appears as Lemma 5.7 in the reference [KRT10] and was independently obtained using the proof of Theorem 4.5 in the same paper.

Lemma 32. *For a unique game G, let the optimum of $\mathrm{SDP}_2(G)$ be $\bar{\sigma}_q(G) = 1 - \epsilon$. Then there exists a quantum strategy $\mathcal{S}$ that achieves the value at least $1 - 7\sqrt{\epsilon}$.*

Proof. Unsurprisingly the quantum strategy is the one defined by the quantum rounding procedure. The major portion of the proof of this lemma appears as the proof of Theorem 4.5 of Kempe, Regev and Toner, which rounds a feasible solution of $\mathrm{SDP}_1(G)$. To account for the fact that we are rounding a feasible solution of $\mathrm{SDP}_2(G)$, we use the Cauchy-Schwarz inequality to bound the difference between q_{st} and q'_{st} (defined below). Let

$$\epsilon_{st} = 1 - \sum_{(a,b)\in R_{st}} \left\langle u_s^a, v_t^b \right\rangle \qquad \text{so that} \qquad \epsilon = \sum_{s,t} \pi(s,t)\epsilon_{st}. \tag{7.4}$$

Let

$$q_{st} = \sum_{(a,b)\in R_{st}} \left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle^2 \|u_s^a\| \, ||v_t^b|| \qquad \text{and} \qquad r = \sum_{(a,b)\in R_{st}} \|u_s^a\| \, ||v_t^b||.$$

Since the game under consideration is a unique game, by the Cauchy-Schwarz inequality, we have that $r \leq 1$. It follows from above that

$$q_{st} = r \sum_{(a,b)\in R_{st}} \frac{\|u_s^a\| \left\|v_t^b\right\|}{r} \left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle^2 \geq r \left(\sum_{(a,b)\in R_{st}} \frac{\|u_s^a\| \, ||v_t^b||}{r} \left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle \right)^2.$$

The last inequality follows from the fact that $px^2 + (1-p)y^2 \geq (px + (1-p)y)^2$. Hence

$$q_{st} \geq \frac{1}{r} \left(\sum_{(a,b)\in R_{st}} \left\langle u_s^a, v_t^b \right\rangle \right)^2 = \frac{(1-\epsilon_{st})^2}{r} \geq \frac{1-2\epsilon_{st}}{r} \geq 1 - 2\epsilon_{st}. \tag{7.5}$$

Let

$$q'_{st} = \left(\sum_{(a,b)\in R_{st}} \left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle^2 \right) \left(\min \left\{ \|u_s^a\|, ||v_t^b|| \right\} \right)^2$$

so that

$$p_{st} \geq 1 - 2\left(1 - q'_{st}\right) = 2q'_{st} - 1 \tag{7.6}$$

by (7.3). By the arithmetic-geometric inequality, we have $\sqrt{xy} - \min\{x, y\} \leq |x-y|/2$, and therefore

$$\begin{aligned}
q_{st} - q'_{st} &\leq \sum_{(a,b)\in R_{st}} \left(\left\langle \frac{u_s^a}{\|u_s^a\|}, \frac{v_t^b}{||v_t^b||} \right\rangle^2 \frac{\left| ||u_s^a||^2 - ||v_t^b||^2 \right|}{2} \right) \\
&\leq \frac{1}{2} \sum_{(a,b)\in R_{st}} \left| ||u_s^a|| - ||v_t^b|| \right| \cdot \left| ||u_s^a|| + ||v_t^b|| \right| \\
&\leq \frac{1}{2} \left[\left(2 - 2 \sum_{(a,b)\in R_{st}} ||u_s^a|| \cdot ||v_t^b|| \right) \left(2 + 2 \sum_{(a,b)\in R_{st}} ||u_s^a|| \cdot ||v_t^b|| \right) \right]^{1/2} \\
&= \left[1 - \left(\sum_{(a,b)\in R_{st}} ||u_s^a|| \cdot ||v_t^b|| \right)^2 \right]^{1/2} \leq \left[1 - \left(\sum_{(a,b)\in R_{st}} \left\langle u_s^a, v_t^b \right\rangle \right)^2 \right]^{1/2} \\
&= \left[1 - (1-\epsilon_{st})^2 \right]^{1/2} \leq \sqrt{2\epsilon_{st}}.
\end{aligned}$$

Now from (7.5)

$$q'_{st} \geq q_{st} - \sqrt{2\epsilon_{st}} \geq 1 - 2\epsilon_{st} - \sqrt{2\epsilon_{st}}.$$

It follows from (7.6) that

$$p_{st} \geq 2q'_{st} - 1 \geq 1 - 4\epsilon_{st} - 2\sqrt{2\epsilon_{st}} \geq 1 - 7\sqrt{\epsilon_{st}}.$$

The success probability of the quantum strategy $\mathcal{S}$ defined by the quantum rounding procedure is

$$\omega_q(G, \mathcal{S}) = \sum_{s,t} \pi(s,t) p_{st} \geq 1 - 7 \sum_{s,t} \pi(s,t) \sqrt{\epsilon_{st}} \geq 1 - 7 \left(\sum_{s,t} \pi(s,t) \right)^{1/2} \left(\sum_{s,t} \pi(s,t) \epsilon_{st} \right)^{1/2}$$

and therefore $\omega_q(G) \geq \omega_q(G, \mathcal{S}) \geq 1 - 7\sqrt{\epsilon}$. This completes the proof.

□

For any MOD_k game G, Kempe, Regev and Toner [KRT10] established the following relationship between the quantum value and the optimum value of $\mathrm{SDP}_1(G)$.

Lemma 33. *For any* MOD_k *game* G*, let the optimum of* $\mathrm{SDP}_1(G)$ *be* $\sigma_q(G) = 1 - \epsilon$*. Then there exists a quantum strategy* $\mathcal{S}$ *such that* $\omega_q(G, \mathcal{S}) \geq 1 - 4\epsilon$*.*

In Section 7.4, we will show that for any MOD_k game G, the optimum values of $\mathrm{SDP}_1(G)$ and $\mathrm{SDP}_2(G)$ are equal.

7.3 Multiplicative semidefinite programs

In this section, we define a product operation on a class of semidefinite program. We show that multiplicative property holds under the product operation assuming that the SDP satisfies a certain condition. The product operation is pertinent to the results presented in this chapter as we later show that $\mathrm{SDP}_2(G)$ is an instance of the class of semidefinite programs discussed below. It is also notationally convenient for us to consider the semidefinite program described below instead of $\mathrm{SDP}_2(G)$.

$$\begin{aligned}
\alpha(\mathcal{G}) = \text{supremum:} \quad & \langle C, X \rangle \\
\text{subject to:} \quad & \forall i \in [m], \quad \langle A_i, X \rangle = a[i], \\
& \forall j \in [n], \quad \langle B_j, X \rangle = b[j], \\
& \forall k \in [l], \quad \langle D_k, X \rangle \geq 0, \\
& X \succeq 0.
\end{aligned}$$

Throughout the rest of the section, we assume that the strong duality holds and the optimum of the primal and the dual problems are attained. The SDP is such that there exists a block partition of the associated matrices such that across that partition the cost matrix is of the form

$$C = \frac{1}{2}\begin{pmatrix} 0 & \hat{C} \\ \hat{C}^\top & 0 \end{pmatrix}$$

and the constraint matrices A_i, B_j, D_k are of the form

$$A_i = \begin{pmatrix} \hat{A}_i & 0 \\ 0 & 0 \end{pmatrix}, \qquad B_j = \begin{pmatrix} 0 & 0 \\ 0 & \hat{B}_j \end{pmatrix}, \qquad \text{and} \qquad D_k = \begin{pmatrix} 0 & \hat{D}_k \\ \hat{D}_k^\top & 0 \end{pmatrix}.$$

Let us assume that the matrices are $(N+M)$-by-$(N+M)$, where the top diagonal block is N-by-N and the bottom diagonal block is M-by-M The dual of the semidefinite program is

$$\begin{aligned} \alpha(\mathcal{G}) = \text{infimum:} \quad & \sum_{i\in[m]} a[i]y[i] + \sum_{j\in[n]} b[j]z[j] \\ \text{subject to:} \quad & Y = \begin{pmatrix} \sum_{i=1}^m y[i]\hat{A}_i & \sum_{k=1}^l v[k]\hat{B}_k \\ \sum_{k=1}^l v[k]\hat{B}_k^\top & \sum_{j=1}^n z[j]\hat{D}_j \end{pmatrix} = Z + \frac{1}{2}\begin{pmatrix} 0 & \hat{C} \\ \hat{C}^\top & 0 \end{pmatrix}, \\ & \forall k \in [l], \quad v[k] \leq 0, \\ & y \in \mathbb{R}^m \quad \text{and} \quad z \in \mathbb{R}^n \quad \text{and} \quad Z \succeq 0. \end{aligned}$$

The *bipartite tensor product* of SDP with itself is defined as the following semidefinite program, which we denote $\mathcal{G} \otimes_b \mathcal{G}$.

$$\begin{aligned} \alpha(\mathcal{G} \otimes_b \mathcal{G}) = \text{supremum:} \quad & \left\langle \bar{C}, X \right\rangle \\ \text{subject to:} \quad & \forall i_1, i_2 \in [m], \quad \left\langle \bar{A}_{i_1 i_2}, X \right\rangle = a[i_1] \cdot a[i_2], \\ & \forall j_1, j_2 \in [n], \quad \left\langle \bar{B}_{j_1 j_2}, X \right\rangle = b[j_1] \cdot b[j_2], \\ & \forall k_1, k_2 \in [l], \quad \left\langle \bar{D}_{k_1 k_2}, X \right\rangle \geq 0, \\ & X \succeq 0. \end{aligned}$$

Here the cost matrix is

$$\bar{C} = \frac{1}{2}\begin{pmatrix} 0 & \hat{C} \otimes \hat{C} \\ (\hat{C} \otimes \hat{C})^\top & 0 \end{pmatrix},$$

and the constraint matrices are

$$\bar{A}_{i_1 i_2} = \begin{pmatrix} \hat{A}_{i_1} \otimes \hat{A}_{i_2} & 0 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \bar{B}_{j_1 j_2} = \begin{pmatrix} 0 & 0 \\ 0 & \hat{B}_{j_1} \otimes \hat{B}_{j_2} \end{pmatrix}$$

and

$$\bar{D}_{k_1k_2} = \begin{pmatrix} 0 & \hat{D}_{k_1} \otimes \hat{D}_{k_2} \\ \left(\hat{D}_{k_1} \otimes \hat{D}_{k_2}\right)^\top & 0 \end{pmatrix}.$$

We have the following multiplicative property for the class of semidefinite programs discussed above.

Theorem 34. *Let $u \in \mathbb{R}^l$ such that $u[k] \geq 0$ for all $k \in [l]$ and*

$$\frac{1}{2}\begin{pmatrix} 0 & \hat{C} \\ \hat{C}^\top & 0 \end{pmatrix} = \begin{pmatrix} 0 & \sum_{k\in[l]} u[k]\hat{D}_k \\ \sum_{k\in[l]} u[k]\hat{D}_k^\top & 0 \end{pmatrix}.$$

If for any dual feasible solution (y, z, v), it holds that

$$\sum_{i\in[m]} a[i]y[i] > 0 \qquad \text{and} \qquad \sum_{j\in[n]} b[j]z[j] > 0$$

then $\alpha(\mathcal{G} \otimes_b \mathcal{G}) = \alpha(\mathcal{G})^2$.

To prove the above theorem, we need the following lemma that gives a necessary condition for optimal dual solutions. The lemma claims that for any optimal dual solution, the contribution from the dual feasible vectors y and z towards the objective value is equal.

Lemma 35. *For any feasible solution of the dual (y, z, v), let*

$$\sum_{i\in[m]} a[i]y[i] > 0 \qquad \text{and} \qquad \sum_{j\in[n]} b[j]z[j] > 0.$$

Then for any optimal solution of the dual $(\hat{y}, \hat{z}, \hat{v})$,

$$\sum_{i\in[m]} a[i]\hat{y}[i] = \sum_{j\in[n]} b[j]\hat{z}[j].$$

Proof. Let (Y, Z) be an optimal dual solution of the semidefinite program. Since Z is positive semidefinite, we have

$$Z = \begin{pmatrix} B \\ D \end{pmatrix}\begin{pmatrix} B \\ D \end{pmatrix}^\top = \begin{pmatrix} BB^\top & BD^\top \\ DB^\top & DD^\top \end{pmatrix}.$$

Let

$$p := \sum_{i\in[m]} a[i]y[i] > 0 \qquad \text{and} \qquad q := \sum_{j\in[n]} b[j]z[j] > 0.$$

Assume that $p \neq q$ and let $p > q$. For any $\lambda > 0$, let Z' be the following matrix

$$Z' = \begin{pmatrix} \lambda B \\ (1/\lambda)D \end{pmatrix} \begin{pmatrix} \lambda B \\ (1/\lambda)D \end{pmatrix}^\top = \begin{pmatrix} \lambda^2 BB^\top & BD^\top \\ DB^\top & (1/\lambda^2)DD^\top \end{pmatrix}$$

and define $Y' = Z' + C$. Since the off diagonal blocks of Z' is same as the off diagonal blocks of Z, it is immediate that the associated vector v' satisfies $v'[k] = v[k] \leq 0$. Moreover, the dual variables that contribute to the diagonal blocks of Y' are either a scalar multiple of λ^2 or $1/\lambda^2$ of the corresponding dual variables that contribute to the diagonal blocks of Y:

$$y'[i] = \lambda^2 y[i] \text{ for all } i \in [m] \qquad \text{and} \qquad z'[j] = \frac{1}{\lambda^2} z[j] \text{ for all } j \in [n].$$

Therefore (Y', Z') form a dual feasible solution with objective value $\lambda^2 p + q/\lambda^2$, which is $2\sqrt{pq}$ $(< p + q)$ for $\lambda^2 = \sqrt{q/p}$, contradicting our assumption.

□

Now we proceed to prove Theorem 34. First of all, let X be an optimal solution of the primal problem $\mathcal{G}$. One can write X as

$$X = \begin{pmatrix} U & V \\ V^\top & W \end{pmatrix}.$$

It follows that $\alpha(\mathcal{G}) = \mathrm{Tr}(\hat{C}V^\top)$. Let $\bar{X}$ be the following matrix:

$$\bar{X} = \begin{pmatrix} U \otimes U & V \otimes V \\ (V \otimes V)^\top & W \otimes W \end{pmatrix}.$$

It is evidently clear that $\bar{X}$ satisfy all the constraints of bipartite tensor product of the SDP with itself. All we need to show is that $\bar{X} \succeq 0$. This easily follows from item 5 of Theorem 3 since $\bar{X}$ is a symmetric restriction of $X \otimes X$. Finally, the objective value of the feasible solution $\bar{X}$ is

$$\mathrm{Tr}\left((\hat{C} \otimes \hat{C})(V \otimes V)^\top\right) = \left(\mathrm{Tr}(\hat{C}V^\top)\right)^2$$

which is equal to $\alpha(\mathcal{G})^2$, since X is an optimal solution of $\mathcal{G}$. Since the primal problem is a maximization problem, this establishes

$$\alpha(\mathcal{G} \otimes_b \mathcal{G}) \geq (\alpha(\mathcal{G}))^2.$$

For the opposite inequality, let (Y, Z) form a dual optimal solution. By Lemma 35, we

can assume that the contributions of the variable y and z to the dual objective value is equal. We first the dual optimal solutions as

$$Y = \begin{pmatrix} \sum_{i=1}^{m} y[i]\hat{A}_i & \sum_{k=1}^{l} v[k]\hat{D}_k \\ \sum_{k=1}^{l} v[k]\hat{D}_k^\top & \sum_{j=1}^{n} z[j]\hat{B}_j \end{pmatrix}$$

and

$$Z = \begin{pmatrix} \sum_{i=1}^{m} y[i]\hat{A}_i & \sum_{k=1}^{l} v[k]\hat{D}_k - \frac{\hat{C}}{2} \\ \sum_{k=1}^{l} v[k]\hat{D}_k^\top - \frac{\hat{C}^\top}{2} & \sum_{j=1}^{n} z[j]\hat{B}_j \end{pmatrix}.$$

By the assumption in Theorem 34, we have

$$Z = \begin{pmatrix} \sum_{i=1}^{m} y[i]\hat{A}_i & \sum_{k=1}^{l} (v[k] - u[k])\, \hat{D}_k \\ \sum_{k=1}^{l} (v[k] - u[k])\, \hat{D}_k^\top & \sum_{j=1}^{n} z[j]\hat{B}_j \end{pmatrix} =: \begin{pmatrix} E & H \\ H^\top & F \end{pmatrix} \tag{7.7}$$

for some $u \in \mathbb{R}^l$ such that $u[k] \geq 0$ such that

$$\sum_{k \in [l]} u[k]\hat{D}_k = \frac{\hat{C}}{2}.$$

Note that $v[k] \leq 0$ and Z is positive semidefinite. For the bipartite tensor product of SDP, we define the dual variables as follows:

$$y[i_1, i_2] = 2y[i_1] \cdot y[i_2] \qquad \text{and} \qquad z[j_1, j_2] = 2z[j_1] \cdot z[j_2] \tag{7.8}$$

and

$$v[k_1, k_2] = -2v[k_1] \cdot v[k_2] + 2u[k_1] \cdot v[k_2] + 2v[k_1] \cdot u[k_2] \leq 0. \tag{7.9}$$

The inequality $v[k_1, k_2] \leq 0$ follows from the inequalities $u[k_1], u[k_2] \geq 0$ and $v[k_1], v[k_2] \leq 0$. Let

$$\bar{Y} = \begin{pmatrix} 2\sum_{i_1,i_2=1}^{m} y[i_1, i_2]\hat{A}_{i_1} \otimes \hat{A}_{i_2} & -2\sum_{k_1,k_2=1}^{l} v[k_1, k_2]\hat{D}_{k_1} \otimes \hat{D}_{k_2} \\ -\sum_{k_1,k_2=1}^{l} v[k_1, k_2] \left(\hat{D}_{k_1} \otimes \hat{D}_{k_2}\right)^\top & \sum_{j_1,j_2=1}^{n} z[j_1, j_2]\hat{B}_{j_1} \otimes \hat{B}_{j_2} \end{pmatrix}.$$

Using (7.8) and (7.9), and

$$2 \sum_{k_1,k_2 \in [l]} u[k_1] \cdot u[k_2]\hat{D}_{k_1} \otimes \hat{D}_{k_2} = \frac{\hat{C} \otimes \hat{C}}{2} \tag{7.10}$$

one obtains that

$$\bar{Z} = 2 \begin{pmatrix} E \otimes E & -H \otimes H \\ -(H \otimes H)^\top & F \otimes F \end{pmatrix}.$$

To show that $(\bar{Y}, \bar{Z})$ is dual feasible, we only need to show that $\bar{Z}$ is positive semidefinite. Since $Z \succeq 0$, it follows from (7.7) that $\bar{Z} \succeq 0$. To see why this is true, notice that for any vector $\begin{pmatrix} u \\ v \end{pmatrix}$, we have

$$\begin{pmatrix} u \\ v \end{pmatrix}^\top \begin{pmatrix} E \otimes E & H \otimes H \\ (H \otimes H)^\top & F \otimes F \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u \\ -v \end{pmatrix}^\top \begin{pmatrix} E \otimes E & -H \otimes H \\ -(H \otimes H)^\top & F \otimes F \end{pmatrix} \begin{pmatrix} u \\ -v \end{pmatrix}$$

and therefore the first matrix is positive semidefinite if and only if the second matrix is positive semidefinite. The first matrix is positive semidefinite because it is a symmetric restriction of $Z \otimes Z$. It follows that $\bar{Z}$, a scalar multiple of the second matrix, is also positive semidefinite. Since objective value of this dual solution is an upper bound on $\alpha(\mathcal{G} \otimes_b \mathcal{G})$, we have that

$$\begin{aligned} \alpha(\mathcal{G} \otimes_b \mathcal{G}) &\leq \sum_{i_1=1}^{m} \sum_{i_2=1}^{m} a[i_1]a[i_2]y[i_1, i_2] + \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} b[j_1]b[j_2]z[j_1, j_2] \\ &= 2\Big(\sum_{i=1}^{m} a[i]y[i]\Big)\Big(\sum_{i=1}^{m} a[i]y[i]\Big) + 2\Big(\sum_{j=1}^{n} b[j]z[j]\Big)\Big(\sum_{j=1}^{n} b[j]z[j]\Big) \\ &= \Big(\sum_{i=1}^{m} a[i]y[i]\Big)^2 + \Big(\sum_{j=1}^{n} b[j]z[j]\Big)^2 = \frac{\big(\alpha(\mathcal{G})\big)^2}{2} + \frac{\big(\alpha(\mathcal{G})\big)^2}{2} = \big(\alpha(\mathcal{G})\big)^2, \end{aligned}$$

where the second last equality follows from Lemma 35. This establishes that

$$\alpha(\mathcal{G} \otimes_b \mathcal{G}) \leq (\alpha(\mathcal{G}))^2$$

which completes the proof of Theorem 34.

It should be noted that if we have two different instances of the semidefinite programs $\mathcal{G}_1$ and $\mathcal{G}_2$ such that both of them satisfy the conditions stated in Theorem 34 and Lemma 35, and are of the required form (as described in the beginning of the section), then the optimum value of the semidefinite program $\mathcal{G}_1 \otimes_b \mathcal{G}_2$ is the product of the optimum values of the individual semidefinite programs.

Moreover, the condition of Theorem 34 is inspired by Lee and Mittal [LM08]. In the next section, we will demonstrate that for any game G, $\mathrm{SDP}_2(G)$ is an instance of the class of semidefinite programs considered in this section. In the manuscript [Upa07], we only

prove the multiplicative property of $\mathrm{SDP}_2(G)$ by essentially following the same argument. However, the proof presented in the Ref. [Upa07] shows the desired multiplicative property for $\mathrm{SDP}_2(G)$ under the product operation arising from the repetition of games instead of the bipartite tensor product. The main reason for present exposition is to avoid messy notations that seems unavoidable while proving the multiplicative property of $\mathrm{SDP}_2(G)$.

7.4 Parallel repetition theorems for quantum unique games

To prove the main results of this chapter, we first show that $\mathrm{SDP}_2(G)$ is an instance of the class of semidefinite programs considered in Section 7.3. To this end, we rewrite the constraints of $\mathrm{SDP}_2(G)$ in the standard form. Consider a two-prover one-round game specified by the question sets of the provers S and T, the answer set $[k]$, the probability distribution $\pi : S \times T \to [0,1]$, and a predicate $V : [k] \times [k] \times S \times T \to \{0,1\}$.

Let C be a matrix whose rows are indexed by $(s,a) \in S \times [k]$ (Alice's questions and answers) and columns are indexed by $(t,b) \in T \times [k]$ (Bob's questions and answers). We call C the *cost matrix* of the game with matrix entries

$$C[(s,a),(t,b)] = \pi(s,t)V(a,b|s,t).$$

Define

$$\bar{C} := \frac{1}{2}\begin{pmatrix} 0 & C \\ C^\top & 0 \end{pmatrix} \tag{7.11}$$

to be the *symmetric cost matrix* of the game. We first start by writing the constraints of the $\mathrm{SDP}_2(G)$ in a form more amenable to the form discussed in the previous section.

1. The constraints
$$\left\langle u_s^a, u_s^b \right\rangle = \left\langle v_t^a, v_t^b \right\rangle = 0 \text{ for all } a,b \in [k] : a \neq b$$
can be alternately written as
$$X[(s,a),(s,b)] = X[(t,a),(t,b)] = 0 \text{ for all } a,b \in [k] : a \neq b$$
for all choices of $(s,t) \in S \times T$. For every choice of $(s,t) \in S \times T$ and $a,b \in [k]$, let E_{sab} and E_{tab} be $0-1$ matrices such that $E_{sab}[(s,a),(s,b)] = E_{sab}[(s,b),(s,a)] = 1$ and $E_{tab}[(t,a),(t,b)] = E_{tab}[(t,b),(t,a)] = 1$, and zero elsewhere. Then for any choices of $(s,t) \in S \times T$ and $a,b \in [k]$, the constraints can be rewritten as
$$\langle E_{sab}, X \rangle = 1 \qquad \text{and} \qquad \langle E_{tab}, X \rangle = 1.$$

2. The constraints $\|z_s\| = \|z_t\| = 1$ and

$$\sum_{a\in[k]} u_s^a = z_s \text{ for all } s \in S \qquad \text{and} \qquad \sum_{b\in[k]} v_t^b = z_t \text{ for all } t \in T$$

can be alternately written as

$$\sum_{a\in[k]} X[(s,a),(s,a)] = 1 \qquad \text{and} \qquad \sum_{b\in[k]} X[(t,b),(t,b)] = 1$$

for all choices of $s \in S$ and $t \in T$. For all $s \in S$ and $t \in T$, let E_{ss} and E_{tt} be $0-1$ matrices such that $E_{ss}[(s,a),(s,a)] = 1$ for all $a \in [k]$ and $E_{tt}[(t,b),(t,b)] = 1$ for all $b \in [k]$, and zero elsewhere. Then for any choice of $s \in S$ and $t \in T$, the constraints can be rewritten as

$$\langle E_{ss}, X\rangle = 1 \qquad \text{and} \qquad \langle E_{tt'}, X\rangle = 1.$$

3. The constraints $\left\langle u_s^a, v_t^b\right\rangle \geq 0$ can be written as

$$X[(s,a),(t,b)] + X[(t,b),(s,a)] \geq 0$$

for every choice of $(s,t) \in S \times T$ and $a, b \in [k]$. For any $(s,t) \in S \times T$ and $a, b \in [k]$, let E_{satb} be the matrix with $E_{satb}[(s,a),(t,b)] = E_{satb}[(t,b),(s,a)] = 1$ and zero elsewhere. Then for any choices of $(s,t) \in S \times T$ and $a, b \in [k]$, the above constraints can be rewritten as

$$\langle E_{satb}, X\rangle \geq 0.$$

We now proceed to show that the constraint matrices are of the form as described in Section 7.3. From items 1 and 2, it is immediate that the matrices E_{ss}, E_{sab} and the matrices E_{tt}, E_{tab} are of the form

$$\begin{pmatrix} * & 0 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} 0 & 0 \\ 0 & * \end{pmatrix},$$

while the matrices E_{satb} are of the form

$$\begin{pmatrix} 0 & * \\ * & 0 \end{pmatrix}.$$

The partition of the blocks is done with respect to the partition in (7.11). Therefore, the semidefinite program $\mathrm{SDP}_2(G)$ is an instance of the class of semidefinite programs considered in the previous section.

It follows that for $u[s,a,t,b] = \pi(s,t)V(a,b|s,t)/2 \geq 0$,

$$\sum_{s,a,t,b} u[(s,a)(t,b)]E_{satb} = \bar{C}$$

and hence the first condition of Theorem 34 is satisfied. One also notes that for any dual feasible variables y and z, their contribution to the objective value are always non-negative:

$$\sum_{s\in S} y[s] > 0 \qquad \text{and} \qquad \sum_{t\in T} z[t] > 0.$$

Therefore, the second condition of Theorem 34 is satisfied. Hence the optimum value of the bipartite tensor product of the semidefinite programs corresponding to games G_1 and G_2, denoted $\mathrm{SDP}_2(G_1 \otimes_b G_2)$, is equal to the product of the optimum values of the $\mathrm{SDP}_2(G_1)$ and $\mathrm{SDP}_2(G_2)$.

However, there is a minor problem. The semidefinite programming relaxation $\mathrm{SDP}_2(G\otimes G)$ corresponding to $G \otimes G$ is not exactly the semidefinite program $\mathrm{SDP}_2(G \otimes_b G)$, the bipartite tensor product of $\mathrm{SDP}_2(G)$ with itself [2]. This can be seen as follows. The constraints of $\mathrm{SDP}_2(G \otimes G)$ are the following three sets of constraints:

1. For all $s_1, s_2 \in S$ and $t_1, t_2 \in T$,

$$\sum_{a_1,a_2\in[k]} u_{s_1s_2}^{a_1a_2} = z_{s_1s_2} \qquad \text{and} \qquad \sum_{b_1,b_2\in[k]} v_{t_1t_2}^{b_1b_2} = z_{t_1t_2}$$

   such that $z_{s_1s_2}$ and $z_{t_1t_2}$ are unit vectors.

2. For all $s_1, s_2 \in S$ and $t_1, t_2 \in T$,

$$\left\langle u_{s_1s_2}^{a_1a_2}, v_{t_1t_2}^{b_1b_2} \right\rangle \geq 0$$

   for all $a_1, a_2 \in [k]$ and $b_1, b_2 \in [k]$.

3. For all $s_1, s_2 \in S$ and $t_1, t_2 \in T$,

$$\left\langle u_{s_1s_2}^{a_1a_2}, u_{s_1s_2}^{b_1b_2} \right\rangle = 0 \qquad \text{and} \qquad \left\langle v_{t_1t_2}^{a_1a_2}, v_{t_1t_2}^{b_1b_2} \right\rangle = 0$$

   if $(a_1, a_2) \neq (b_1, b_2)$.

The bipartite tensor product of $\mathrm{SDP}_2(G)$ with itself does contain the set of constraints stated in items 1 and 2. However, there are other constraints in $\mathrm{SDP}_2(G \otimes_b G)$ that are

[2] This fact was pointed to us by Oded Regev.

different from the set of constraints stated in item 3. These are the set of constraints that arise from taking the "bipartite tensor product" of constraint matrices $E_{s_1s_1}$ and $E_{s_2a_2b_2}$, $E_{t_1t_1}$ and $E_{t_2a_2b_2}$, $E_{s_1a_1b_1}$ and $E_{s_2a_2b_2}$, and $E_{t_1a_1b_1}$ and $E_{t_2a_2b_2}$, for appropriate choices of (a_1, b_1) and (a_2, b_2). Some of these constraints do appear in item 3 above but there are other constraints that do not appear in item 3.

The set of constraints that arise from taking the bipartite tensor product of $E_{s_1a_1b_1}$ and $E_{s_2a_2b_2}$, and $E_{t_1a_1b_1}$ and $E_{t_2a_2b_2}$ can be rewritten as

$$\left\langle u_{s_1s_2}^{a_1b_1}, u_{s_1s_2}^{a_2b_2} \right\rangle = 0 \qquad \text{and} \qquad \left\langle v_{t_1t_2}^{a_1b_1}, v_{t_1t_2}^{a_2b_2} \right\rangle = 0$$

for $a_1 \neq a_2$ and $b_1 \neq b_2$. These are the set of constraints that are included in item 3 above. The constraints that arise from the bipartite tensor product of $E_{s_1s_1}$ and $E_{s_2a_2b_2}$ can be rewritten as

$$\sum_{a_1 \in [k]} \left\langle u_{s_1s_2}^{a_1a_2}, u_{s_1s_2}^{a_1b_2} \right\rangle = 0 \qquad \text{and} \qquad \sum_{b_1 \in [k]} \left\langle v_{t_1t_2}^{b_1a_2}, v_{t_1t_2}^{b_1b_2} \right\rangle = 0 \tag{7.12}$$

for $a_2 \neq b_2$ and the constraints that arise from the bipartite tensor product of $E_{t_1t_1}$ and $E_{t_2a_2b_2}$ can be rewritten as

$$\sum_{a_1 \in [k]} \left\langle u_{s_1s_2}^{a_2a_1}, u_{s_1s_2}^{b_2a_1} \right\rangle = 0 \qquad \text{and} \qquad \sum_{b_1 \in [k]} \left\langle v_{t_1t_2}^{a_2b_1}, v_{t_1t_2}^{b_2b_1} \right\rangle = 0. \tag{7.13}$$

for $a_2 \neq b_2$. It is immediate that if the set of constraints in item 3 are satisfied then the constraints in (7.12) and (7.13) are also satisfied. Therefore, any primal feasible solution of $\mathrm{SDP}_2(G \otimes G)$ is also a feasible solution of the bipartite tensor product of $\mathrm{SDP}_2(G)$ with itself. Recall that the optimum value of $\mathrm{SDP}_2(G)$ is denoted $\bar{\sigma}_q(G)$. It immediately follows that the optimum value of $\mathrm{SDP}_2(G \otimes G)$ is at most the optimum of $\mathrm{SDP}_2(G \otimes_b G)$, which is exactly $(\bar{\sigma}_q(G))^2$. Using this argument inductively, we obtain that the optimum value of $\mathrm{SDP}_2\left(G^{\otimes n}\right)$ is at most the optimum value of n-fold bipartite tensor product of $\mathrm{SDP}_2(G)$, which implies that

$$\bar{\sigma}\left(G^{\otimes n}\right) = \left(\bar{\sigma}(G)\right)^n.$$

The proof of Theorem 28 follows easily. Let $\omega_q(G) = 1 - \epsilon$. Using Lemma 32, we have that

$$\omega_q\left(G^{\otimes n}\right) \leq \bar{\sigma}_q\left(G^{\otimes n}\right) = \left(\bar{\sigma}_q(G)\right)^n \leq \left(1 - \frac{\epsilon^2}{49}\right)^n.$$

7.4.1 XOR games

Next we consider the case of XOR games for which we already know that a perfect parallel repetition theorem holds. Below we show an alternate proof of perfect parallel repetition

theorem for XOR games that goes via analyzing the quantum value of the game instead of the quantum bias of the game. To begin, we state the following proposition from Cleve, Høyer, Toner, and Watrous [CHTW04a].

Proposition 36. *Let G be an XOR game. Then*

$$\omega_q(G) = \frac{1}{2} + \frac{1}{2} \max_{x_s, y_t} \sum_{s,t} \pi(s,t)(V(0|s,t) - V(1|s,t))\langle x_s, y_t \rangle$$

where x_s, y_t lie in a unit ball for all $s \in S$ and $t \in T$.

A consequence of Proposition 36 is that the quantum value of an XOR game is

$$\omega_q(G) = \max_{x_s, y_t} \sum_{s,t} \pi(s,t) \frac{1}{2} \begin{cases} (1 + \langle x_s, y_t \rangle) & \text{if } V(0,0|s,t) = V(1,1|s,t) = 1 \\ (1 - \langle x_s, y_t \rangle) & \text{if } V(0,1|s,t) = V(1,0|s,t) = 1. \end{cases} \tag{7.14}$$

The following theorem establishes that the quantum value of an XOR game is equal to the optimum of SDP relaxations considered in this chapter. The proof is similar to the proof presented in Section 6.5 that relates the quantum value of XOR games and the optimum values of the Feige-Lovász relaxations.

Theorem 37. *For an XOR game G, $\bar{\sigma}_q(G) = \sigma_q(G) = \omega_q(G)$.*

Proof. It is easy to see that $\omega_q(G) \leq \sigma_q(G) \leq \bar{\sigma}_q(G)$. Therefore, we only need to show that $\bar{\sigma}_q(G) \leq \omega_q(G)$ to establish the theorem .

To this end, let $\{u_s^a : a \in \{0,1\}, s \in S\}$ and $\{v_t^b : b \in \{0,1\}, t \in T\}$ be a system of vectors that form an optimal solution of $\mathrm{SDP}_2(G)$. Define the following vectors for every choice os $(s,t) \in S \times T$

$$x_s := u_s^0 - u_s^1, \qquad \text{and} \qquad y_t := v_t^0 - v_t^1.$$

Since $\langle u_s^0, u_s^1 \rangle = 0$ and $\langle v_t^0, v_t^1 \rangle = 0$, and the fact that

$$z_s = u_s^0 + u_s^1, \qquad \text{and} \qquad z_t = v_t^0 + v_t^1$$

are unit vectors, it is immediate that $\{x_s : s \in S\}$ and $\{y_t : t \in T\}$, z_1 are also unit vectors. For any question (s,t) such that $a \oplus b = 0$, the contribution of the vectors towards the objective value of $\mathrm{SDP}_2(G)$ is

$$\begin{aligned} \langle u_s^0, v_t^0 \rangle + \langle u_s^1, v_t^1 \rangle &= \langle z_s, z_t \rangle - \left(\left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle \right) \\ &\leq 1 - \left(\left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle \right) \\ &= 1 - \left(\left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle - \langle x_s, y_t \rangle \right) \end{aligned}$$

and hence

$$\left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle \leq \frac{1 + \langle x_s, y_t \rangle}{2}. \tag{7.15}$$

Similarly, for any question (s,t) such that $a \oplus b = 1$, the contribution of the vectors towards the objective value of $\mathrm{SDP}_2(G)$ is

$$\begin{aligned} \langle u_s^0, v_t^1 \rangle + \langle u_s^1, v_t^0 \rangle &= \langle z_s, z_t \rangle - \left(\left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle \right) \\ &\leq 1 - \left(\left\langle u_s^0, v_t^0 \right\rangle + \left\langle u_s^1, v_t^1 \right\rangle \right) \\ &= 1 - \left(\left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle + \langle x_s, y_t \rangle \right) \end{aligned}$$

and hence

$$\left\langle u_s^0, v_t^1 \right\rangle + \left\langle u_s^1, v_t^0 \right\rangle \leq \frac{1 - \langle x_s, y_t \rangle}{2}. \tag{7.16}$$

From (7.15) and (7.16) the optimum value of $\mathrm{SDP}_2(G)$ is at most

$$\sum_{s,t} \pi(s,t) \frac{1}{2} \begin{cases} (1 + \langle x_s, y_t \rangle) & \text{if } V(0,0|s,t) = V(1,1|s,t) = 1 \\ (1 - \langle x_s, y_t \rangle) & \text{if } V(0,1|s,t) = V(1,0|s,t) = 1 \end{cases}$$

which is at most $\omega_q(G)$ from (7.14). The completes the proof of the theorem.

□

The prefect parallel repetition theorem follows easily from the fact that for an XOR game G,

$$\omega_q(G^{\otimes n}) \leq \bar{\sigma}_q(G^{\otimes n}) = \left(\bar{\sigma}_q(G)\right)^n = \left(\omega_q(G)\right)^n.$$

7.4.2 MOD_k games

Our final result concerns with improving the bound in the parallel repetition theorem for MOD_k games. The MOD_k games can be viewed as generalization of XOR games, where each prover's response $a, b \in [k]$ to questions s and t must satisfy

$$a - b \equiv f(s,t) \mod k.$$

In this subsection, we show that for any MOD_k game G, the optimum values of $\mathrm{SDP}_1(G)$ and $\mathrm{SDP}_2(G)$ are equal. Then a straightforward application of Lemma 33 proves Theorem 29. In general, one cannot hope for the optimum values of both relaxations to be equal. However, the following proposition, which shows that there exists an optimal solution for MOD_k games with certain additional properties, allows us to prove the desired relationship between the two relaxations.

Proposition 38. *For a* MOD_k *game* G*, there exists an optimal solution* $\{u_s^a : a \in [k], s \in S\}$ *and* $\{v_t^b : b \in [k], t \in T\}$ *for* $\mathrm{SDP}_1(G)$ *and* $\mathrm{SDP}_2(G)$ *satisfying*

$$\sum_{a\in[k]} u_s^a = z_s \qquad \text{and} \qquad \sum_{b\in[k]} v_t^b = z_t.$$

such that for all s, t, a, b*,*

$$\langle u_s^a, z_t\rangle = \frac{1}{k}\langle z_s, z_t\rangle \qquad \text{and} \qquad \left\langle z_s, v_t^b\right\rangle = \frac{1}{k}\langle z_s, z_t\rangle . \tag{7.17}$$

Moreover, $||u_s^a|| = ||v_t^b|| = \frac{1}{\sqrt{k}}$.

Proof. We prove the proposition for $\mathrm{SDP}_2(G)$. The proof for $\mathrm{SDP}_1(G)$ follows from setting $z_s = z_t$ for every choices of the pair (s, t). To prove the claim, assume that the sets $\{u_s^a : a \in [k], s \in S\}$ and $\{v_t^b : b \in [k], t \in T\}$ form an optimal solution of $\mathrm{SDP}_2(G)$. Define the vectors

$$\hat{u}_s^a := \frac{1}{\sqrt{k}} \bigoplus_{j=a+1}^{a} u_s^j \qquad \text{and} \qquad \hat{v}_t^b := \frac{1}{\sqrt{k}} \bigoplus_{j=b+1}^{b} v_t^j.$$

Here the addition operation in the direct sum is modulo k. One easily observes that $\left\langle \hat{u}_s^a, \hat{v}_t^b\right\rangle \geq 0$ and

$$\|\hat{u}_s^a\| = \frac{1}{\sqrt{k}} \qquad \text{and} \qquad \left\|\hat{v}_t^b\right\| = \frac{1}{\sqrt{k}}.$$

Moreover,

$$\sum_{a\in[k]} \hat{u}_s^a = \frac{1}{\sqrt{k}} \bigoplus_{j\in[k]} z_s =: \hat{z}_s \qquad \text{and} \qquad \sum_{b\in[k]} \hat{v}_t^b = \frac{1}{\sqrt{k}} \bigoplus_{j\in k} z_t =: \hat{z}_t.$$

Therefore, the two sets of vectors

$$\{\hat{u}_s^a : a \in [k], s \in S\} \qquad \text{and} \qquad \{\hat{v}_t^b : b \in [k], t \in T\}$$

form a feasible solution of $\mathrm{SDP}_2(G)$. In fact, the objective value of the new solution is same as the objective value of the optimal solution. To see why this is true, note that the contribution of the new set of vectors for a linear equation of the form $a - b \equiv c \mod k$ is

$$\sum_{a,b:a-b=c} \left\langle \hat{u}_s^a, \hat{v}_t^b\right\rangle = \sum_{a\in[k]} \left\langle \hat{u}_s^a, \hat{v}_t^{a-c}\right\rangle = \frac{1}{k}\sum_{a\in[k]}\sum_{j=1}^{k} \left\langle u_s^j, v_t^{j-c}\right\rangle = \sum_{a\in[k]} \left\langle u_s^a, v_t^{a-c}\right\rangle .$$

Since the objective function depends on $a - b$, we obtain that the two sets of vectors

$\{\hat{u}_s^a : a \in [k], s \in S\}$ and $\{\hat{v}_t^b : b \in [k], t \in T\}$ also form an optimal solution of $\mathrm{SDP}_2(G)$. Moreover, we also observe that

$$\langle \hat{u}_s^a, \hat{z}_t \rangle = \frac{1}{k} \left\langle \bigoplus_{j=a+1}^{a} u_s^j, \bigoplus_{j \in k} z_t \right\rangle = \frac{1}{k} \sum_{a \in [k]} \langle u_s^a, z_t \rangle = \frac{\langle z_s, z_t \rangle}{k} = \frac{1}{k} \langle \hat{z}_s, \hat{z}_t \rangle$$

and similarly $\left\langle \hat{z}_s, \hat{v}_t^b \right\rangle = \frac{\langle \hat{z}_s, \hat{z}_t \rangle}{k}$.

□

To show that the optimum values of $\mathrm{SDP}_1(G)$ and $\mathrm{SDP}_2(G)$ are equal, we introduce another semidefinite program denoted $\mathrm{SDP}_3(G)$. For the remainder of the section, we assume the answer set to be $\{0, 1, \ldots, k-1\}$ and $\omega = \exp(2\pi i/k)$, the k-th root of unity. Let $\hat{\sigma}_q(G)$ denote the optimum value of the optimization problem below.

$$\begin{aligned}
\hat{\sigma}_q(G) = \text{maximize:} \quad & \left(\frac{1}{k} + \frac{1}{k} \sum_{s,t} \pi(s,t) \left(\sum_{j=1}^{k-1} \omega^{jf(s,t)} \left\langle x_s^j, y_t^j \right\rangle \right) \right) \\
\text{subject to:} \quad & \forall s,t,j, \quad \left\| x_s^j \right\| = \left\| y_t^j \right\| = 1, \\
& \forall s,t,i,j : i \neq j, \quad \left\langle x_s^i, x_s^j \right\rangle = \left\langle y_t^i, y_t^j \right\rangle = 0, \\
& \forall s,t, \quad -1 \leq \sum_{j=1}^{k-1} \omega^{jf(s,t)} \left\langle x_s^j, y_t^j \right\rangle \leq k-1, \\
& \forall s,t,a,b, \quad \sum_{i,j=1}^{k-1} \omega^{(ia-jb)} \left\langle x_s^i, y_t^j \right\rangle \geq -1.
\end{aligned}$$

The only use of the third set of constraints is to make sure that the objective function is always real. It will be easy to parse the optimization problem when the answer size of both the provers is 2. Note that for $k = 2$, the last two constraints are redundant because they reduce to the obvious fact that the inner product of two unit vectors lie between $[-1, 1]$. Hence $\mathrm{SDP}_3(G)$ reduces to the following.

$$\begin{aligned}
\hat{\sigma}_q(G) = \text{maximize:} \quad & \left(\frac{1}{2} + \frac{1}{2} \sum_{s,t} \pi(s,t) (-1)^{f(s,t)} \langle x_s, y_t \rangle \right) \\
\text{subject to:} \quad & \forall s,t, \quad \|x_s\| = \|y_t\| = 1.
\end{aligned}$$

which is the quantum value of an XOR game. For MOD_k games, the relationship between the set of unit vectors in the above problem and the set of vectors arising from a feasible solution of $\mathrm{SDP}_2(G)$ will be made precise in the proof of the following theorem.

Theorem 39. *For any* MOD_k *game* G, $\sigma_q(G) = \hat{\sigma}_q(G) = \bar{\sigma}_q(G)$.

Proof. We first prove that the optimum of $\mathrm{SDP}_2(G)$ is at most the optimum of $\mathrm{SDP}_3(G)$. To this end, let

$$\{u_s^a : a \in [k], s \in S\} \qquad \text{and} \qquad \{v_t^b : b \in [k], t \in T\}$$

form an optimal solution of $\mathrm{SDP}_2(G)$ satisfying (7.17) and the uniformity constraint (refer to Proposition 38). Define the unit vectors x_s^i and y_t^j as

$$x_s^i := \sum_{a=0}^{k-1} \omega^{ia} u_s^a \qquad \text{and} \qquad y_t^j := \sum_{b=0}^{k-1} \omega^{jb} v_t^b \tag{7.18}$$

for every choice of $s \in S$, $t \in T$, and $i, j \in [k-1]$. Let $x_s^0 = z_s$ and $y_t^0 = z_t$ for every choice of $(s,t) \in S \times T$. Below we prove that the vectors

$$\{x_s^i : i \in [k-1], s \in S\} \qquad \text{and} \qquad \{y_t^j : j \in [k-1], t \in T\}$$

form a feasible solution of $\mathrm{SDP}_3(G)$. The following four items show that all four sets of constraints are satisfied.

1. For the first set of constraints of $\mathrm{SDP}_3(G)$, it is immediate from (7.18) that

$$\left\|x_s^i\right\| = \left\|y_t^i\right\| = 1 \qquad \text{for all} \qquad i \in [k-1].$$

2. For the second set of constraints of $\mathrm{SDP}_3(G)$, we have for distinct i and j

$$\left\langle x_s^i, x_s^j \right\rangle = \left\langle \sum_{a=0}^{k-1} \omega^{ia} u_s^a, \sum_{b=0}^{k-1} \omega^{jb} u_s^b \right\rangle = \sum_{a=0}^{k-1} \omega^{(i-j)a} ||u_s^a||^2 = \sum_{a=0}^{k-1} \frac{\omega^{(i-j)a}}{k} = 0.$$

   Similarly, it can be argued that for distinct i and j, $\left\langle y_t^i, y_t^j \right\rangle = 0$.

3. For the third set of constraints of $\mathrm{SDP}_3(G)$, we observe that

$$\frac{1}{k} \sum_{j=0}^{k-1} \omega^{jf(s,t)} \left\langle x_s^j, y_t^j \right\rangle = \frac{1}{k} \sum_{a,b=0}^{k-1} \sum_{j=0}^{k-1} \omega^{j(f(s,t)-(a-b))} \left\langle u_s^a, v_t^b \right\rangle = \sum_{a,b:a-b=f(s,t)} \left\langle u_s^a, v_t^b \right\rangle \in [0,1].$$

   Using the fact that $x_s^0 = z_s$ and $y_t^0 = z_t$, we have

$$-1 \leq -\left\langle z_s, z_t \right\rangle \leq \sum_{j=1}^{k-1} \omega^{jf(s,t)} \left\langle x_s^j, y_t^j \right\rangle \leq k-1.$$

4. For the fourth set of constraints of $\mathrm{SDP}_3(G)$, using (7.18) we have

$$u_s^a = \frac{1}{k}\sum_{i=0}^{k-1}\omega^{-ia}x_s^i \qquad \text{and} \qquad v_t^b = \frac{1}{k}\sum_{j=0}^{k-1}\omega^{-jb}y_t^j, \tag{7.19}$$

and since $\left\langle u_s^a, v_t^b\right\rangle \geq 0$, it follows that

$$\begin{aligned} 0 \leq k^2\left\langle u_s^a, v_t^b\right\rangle &= \sum_{i,j=1}^{k-1}\omega^{(ia-jb)}\left\langle x_s^i, y_t^j\right\rangle + \left\langle \sum_{i=1}^{k-1}\omega^{-ia}x_s^i, y_t^0\right\rangle \\ &\quad + \left\langle x_s^0, \sum_{j=1}^{k-1}\omega^{-jb}y_t^j\right\rangle + \left\langle x_s^0, y_t^0\right\rangle. \end{aligned}$$

From (7.19), we obtain

$$\sum_{i=1}^{k-1}\omega^{-ia}x_s^i = z_s - k\cdot u_s^a \qquad \text{and} \qquad \sum_{j=1}^{k-1}\omega^{-jb}y_t^j = z_t - k\cdot v_t^b$$

and therefore from Proposition 38, we have that

$$\sum_{i,j=1}^{k-1}\omega^{(ia-jb)}\left\langle x_s^i, y_t^j\right\rangle \geq \left\langle z_s, z_t\right\rangle - k\left\langle z_s, v_t^b\right\rangle - k\left\langle u_s^a, z_t\right\rangle = -\left\langle z_s, z_t\right\rangle \geq -1.$$

Hence all the constraints of $\mathrm{SDP}_3(G)$ are satisfied, which implies that the system of unit vectors $\{x_s^i : i \in [k-1], s \in S\}$ and $\{y_t^j : j \in [k-1], t \in T\}$ as defined above forms a feasible solution to the optimization problem. It remains to prove that the objective value of this feasible solution is equal to the objective value of optimal solution of $\mathrm{SDP}_2(G)$, which is demonstrated below.

$$\begin{aligned} \hat{\sigma}_q(G) &\geq \left(\frac{1}{k} + \frac{1}{k}\sum_{s,t}\sum_{j=1}^{k-1}\pi(s,t)\omega^{jf(s,t)}\left\langle x_s^j, y_t^j\right\rangle\right) \geq \frac{1}{k}\sum_{s,t}\sum_{j=0}^{k-1}\pi(s,t)\omega^{jf(s,t)}\left\langle x_s^j, y_t^j\right\rangle \\ &= \frac{1}{k}\sum_{s,t}\sum_{a,b=0}^{k-1}\sum_{j=0}^{k-1}\pi(s,t)\omega^{j(f(s,t)-(a-b))}\left\langle u_s^a, v_t^b\right\rangle \\ &= \sum_{s,t}\sum_{a,b:a-b=f(s,t)}\pi(s,t)\left\langle u_s^a, v_t^b\right\rangle = \sum_{s,t}\sum_{a,b}\pi(s,t)V(a,b|s,t)\left\langle u_s^a, v_t^b\right\rangle, \end{aligned}$$

and hence $\hat{\sigma}_q(G) \geq \bar{\sigma}_q(G)$.

Next we proceed to show that the optimum of $\text{SDP}_3(G)$ is at most the optimum of $\text{SDP}_1(G)$. To this end, let

$$\{x_s^i : i \in [k-1], s \in S\} \qquad \text{and} \qquad \{y_t^j : j \in [k-1], t \in T\}$$

form an optimal solution of $\text{SDP}_3(G)$. Let z be a unit vector that is orthonormal to the above system of unit vectors and let $x_s^0 = y_t^0 = z$ for every choice of $(s,t) \in S \times T$. Define the vectors u_s^a and v_t^b as

$$u_s^a := \frac{1}{k} \sum_{i=0}^{k-1} \omega^{-ia} x_s^i \qquad \text{and} \qquad v_t^b := \frac{1}{k} \sum_{j=0}^{k-1} \omega^{-jb} y_t^j. \tag{7.20}$$

for all choice of $(s,t) \in S \times T$ and $a, b \in \{0, 1, \ldots, k-1\}$.

Now we proceed to show that the vectors in (7.20) form a feasible solution to $\text{SDP}_1(G)$. To begin, first observe that $\{x_s^i : i \in [k-1]\}$ are mutually orthonormal and likewise $\{y_t^j : j \in [k-1]\}$ are mutually orthonormal. It follows from (7.20) that

$$||u_s^a|| = ||v_t^b|| = \frac{1}{\sqrt{k}}.$$

The following three items establish the feasibility of vectors in (7.20) for $\text{SDP}_1(G)$.

1. It follows from (7.20) that

$$\sum_{a=0}^{k-1} u_s^a = \frac{1}{k} \sum_{a=0}^{k-1} \sum_{i=0}^{k-1} \omega^{-ia} x_s^i = \frac{1}{k} \sum_{i=0}^{k-1} \left(\sum_{a=0}^{k-1} \omega^{-ia} \right) x_s^i = x_s^0 = z.$$

   Similarly, it can be argued that

$$\sum_{b=0}^{k-1} v_t^b = \frac{1}{k} \sum_{b=0}^{k-1} \sum_{j=0}^{k-1} \omega^{-jb} y_t^j = \frac{1}{k} \sum_{j=0}^{k-1} \left(\sum_{b=0}^{k-1} \omega^{-jb} \right) y_t^j = y_t^0 = z.$$

   This implies that the collection of vectors $\{u_s^a\}$ and $\{v_t^b\}$ as defined in (7.20) sum to same unit vector z.

2. For distinct a and b

$$k^2 \left\langle u_s^a, u_s^b \right\rangle = \left\langle \sum_{i=0}^{k-1} \omega^{-ia} x_s^i, \sum_{j=0}^{k-1} \omega^{-jb} x_s^j \right\rangle = \sum_{i=0}^{k-1} \omega^{i(a-b)} \left\langle x_s^i, x_s^i \right\rangle = \sum_{i=0}^{k-1} \omega^{i(a-b)} = 0.$$

   Similarly, for distinct choices of a and b, it can be argued that $\left\langle v_t^a, v_t^b \right\rangle = 0$.

3. Using (7.20), we have that

$$\begin{aligned} k^2 \left\langle u_s^a, v_t^b \right\rangle &= \sum_{i,j=1}^{k-1} \omega^{(ia-jb)} \left\langle x_s^i, y_t^j \right\rangle + \left\langle \sum_{i=1}^{k-1} \omega^{-ia} x_s^i, y_t^0 \right\rangle + \left\langle x_s^0, \sum_{j=1}^{k-1} \omega^{-jb} y_t^j \right\rangle + \left\langle x_s^0, y_t^0 \right\rangle \\ &\geq -1 + \left\langle \sum_{i=1}^{k-1} \omega^{-ia} x_s^i, z \right\rangle + \left\langle z, \sum_{j=1}^{k-1} \omega^{-ia} y_t^j \right\rangle + 1 = 0. \end{aligned}$$

Here we used the fact that z is orthonormal to x_s^i and y_t^j for all $i, j \in [k-1]$ and $(s,t) \in S \times T$, and that the first term in the expression is at least -1 because x_s^i and y_t^j form a feasible solution of $\mathrm{SDP}_3(G)$.

This shows that all the constraints of $\mathrm{SDP}_1(G)$ is satisfied and hence $\{u_s^a : a \in [k], s \in S\}$ and $\{v_t^b : b \in [k], t \in T\}$ form a feasible solution of $\mathrm{SDP}_1(G)$. It holds that

$$\begin{aligned} \sigma_q(G) &\geq \sum_{s,t} \sum_{a,b} \pi(s,t) V(a,b|s,t) \left\langle u_s^a, v_t^b \right\rangle = \frac{1}{k} \sum_{s,t} \sum_{a,b=0}^{k-1} \sum_{j=0}^{k-1} \pi(s,t) \omega^{j(f(s,t)-(a-b))} \left\langle u_s^a, v_t^b \right\rangle \\ &= \frac{1}{k} \sum_{s,t} \sum_{j=0}^{k-1} \pi(s,t) \omega^{jf(s,t)} \left\langle \sum_{a=0}^{k-1} \omega^{ja} u_s^a, \sum_{b=0}^{k-1} \omega^{jb} v_t^b \right\rangle \\ &= \left(\frac{1}{k} + \frac{1}{k} \sum_{s,t} \sum_{j=1}^{k-1} \pi(s,t) \omega^{jf(s,t)} \left\langle x_s^j, y_t^j \right\rangle \right) = \hat{\sigma}_q(G). \end{aligned}$$

One cannot guarantee that the system of vectors $\{u_s^a : a \in [k], s \in S\}$ and $\{v_t^b : b \in [k], t \in T\}$ are real vectors. However, using Lemma 3.2 of Kempe, Regev and Toner [KRT10], one can convert the above set of vectors into real vectors that satisfy the feasibility of $\mathrm{SDP}_1(G)$ with same objective value. This completes the proof of the theorem.

□

The proof of parallel repetition theorem for MOD_k games follow easily from Theorem 39. For any such game G with $\omega_q(G) = 1 - \epsilon$, we have

$$\omega_q\left(G^{\otimes n}\right) \leq \bar{\sigma}_q\left(G^{\otimes n}\right) = \left(\bar{\sigma}_q(G)\right)^n = \left(\sigma_q(G)\right)^n \leq \left(1 - \frac{\epsilon}{4}\right)^n.$$

A consequence of Theorem 39 is that for any MOD_k game G, $\sigma_q\left(G^{\otimes n}\right) = \left(\sigma_q(G)\right)^n$. This is provably not true for unique games [KR10].

7.5 Related work

The result presented in this chapter is based on the manuscript [Upa07]. However, the present exposition is vastly different from the manuscript. We stress that the proof of the multiplicative property of a class of semidefinite program in Section 7.3 is based on an observation made by Lee and Mittal [LM08] and was not proved in the original manuscript. The purpose of this section is merely to avoid messy notations one encounters when proving the multiplicative property of $\mathrm{SDP}_2(G)$. In fact, the original proof showed that

$$\bar{\sigma}_q\left(G^{\otimes n}\right) = \left(\bar{\sigma}_q(G)\right)^n$$

without any reference to bipartite tensor product.

The parallel repetition results in this chapter was independently obtained by Kempe, Regev and Toner [KRT10]. We obtained the result after the appearance of a preliminary version of the Ref. [KRT10] on arXiv in October 2007. The preliminary version did not include the parallel repetition results. The techniques used to show parallel repetition results for general unique games is almost same as the one presented in their paper. One of the differences is the choice of semidefinite program used to establish the theorem. The one discussed in their paper is a relaxation of $\mathrm{SDP}_2(G)$ and is described below for the sake of completeness.

$$\begin{aligned}
\text{maximize:} \quad & \sum_{s,t}\sum_{a,b} \pi(s,t)V(a,b|s,t)\left\langle u_s^a, v_t^b\right\rangle \\
\text{subject to:} \quad & \forall s,t, \quad \|z_s\| = \|z_t\| = 1, \\
& \forall s,t \quad \sum_{a\in[k]} u_s^a = z_s \quad \text{and} \quad \sum_{b\in[k]} v_t^b = z_t, \\
& \forall s,t,a,b : a \neq b \quad \left\langle u_s^a, u_s^b\right\rangle = \left\langle v_t^a, v_t^b\right\rangle = 0.
\end{aligned}$$

The second SDP relaxation discussed in this chapter has an extra set of non-negativity constraints:

$$\left\langle u_s^a, v_t^b\right\rangle \geq 0 \qquad \text{for all} \qquad (s,t) \in S \times T \text{ and } a,b \in [k].$$

The set of non-negativity constraints are not essential in obtaining parallel repetition theorems for quantum unique games. For such games, one can trivially observe that Lemma 30 holds. However, for two-prover one-round games that are not unique, in general the optimum value of the semidefinite program can be strictly greater than 1 whereas the quantum value is strictly less than 1. Therefore, for such games, one cannot hope to obtain parallel repetition result using the above SDP. One such class of games are two-prover one-round binary games [CHTW04a] for which semidefinite programming based approach might be useful [Bei10].

The analysis of the quantum rounding procedure for general unique games in Kempe, Regev and Toner is exactly the same as the one presented in this chapter. We obtain the proof of Lemma 32 by using the proof of Theorem 4.5 of their paper [KRT10] (Theorem 4.4 in in the preliminary arXiv version), and a simple application of the Cauchy-Schwarz inequality. The proof of Lemma 32 is presented in this chapter for the sake of completeness.

Although oure result on MOD_k games is same as the one presented in their paper, the approach is completely different. We show that for any such game G, the optimum values of $\mathrm{SDP}_1(G)$ and $\mathrm{SDP}_2(G)$ are equal. As a consequence, we are able to show that the first relaxation is also multiplicative. It also gives us an alternate proof of the perfect parallel repetition theorem for XOR games, discussed in the previous chapter. We remark that the proof of Lemma 33 does not use the fact that the system of vectors $\{u_s^a : a \in [k]\}$ and $\{v_t^b : b \in [k]\}$ for every question (s,t) should sum up to same unit vector z [KRT10]. Therefore, for MOD_k games, the quantum rounding procedure applied on the second relaxation in Section 7.2 or the relaxation mentioned above will also give the same quantitative performance. This provides a simpler proof of Theorem 29, which is mentioned in Ref. [KRT10]. We also remark that the quantum value of an XOR game and the optimum value of the above mentioned semidefinite program are equal. As a consequence, their result also implies perfect parallel repetition theorem for quantum XOR games, although this is not explicitly stated.

Another direction of research has focused on coming up with optimal lower bound on parallel repetition for quantum unique games. More precisely, Kempe and Regev [KR10] asked the following question on parallel repetition of unique games. For a unique game G with quantum value $\omega_q(G) = 1-\epsilon$, what is the optimal dependence of $\omega_q\left(G^{\otimes n}\right)$ on ϵ? They proved the existence of a unique game for which the bound obtained in Theorem 28 is tight with respect to the exponent in ϵ. This also establishes that while $\bar{\sigma}_q(G^{\otimes n}) = (\bar{\sigma}_q(G))^n$, the same does not hold for $\mathrm{SDP}_1(G)$ for if it holds that $\sigma_q(G^{\otimes n}) = (\sigma_q(G))^n$ for any unique game G, then the optimal dependence of $\omega_q\left(G^{\otimes n}\right)$ on ϵ will be linear instead of quadratic.

In the same paper, Kempe and Regev also show that for non-signaling strategies, the bound obtained by Holenstein [Hol07] is also tight. Roughly speaking, non-signaling strategies are strategies where the marginal distribution of Alice's answer is independent of Bob's question and likewise for Bob. One should note that for unique games as defined in this chapter, its non-signaling value is always 1: simply choose the following distribution P_{st} on the answer pairs (a,b)

$$P_{st}(a,b) = \frac{1}{k} \qquad \text{if and only if} \qquad a = \Gamma_{st}(b).$$

Here Γ_{st} is the permutation on question (s,t). It is immediate that the marginal distribution of Alice's answer is uniform and hence independent of Bob's question and likewise for Bob.

Lee and Mittal [LM08] demonstrated a class of semidefinite programs for which multiplicative property holds. The class of semidefinite programs considered in their paper is similar but more general than the one presented in Section 7.3. They considered the following semidefinite program:

$$\max\left\{\langle C, X\rangle : \left\langle A^{(k)}, X\right\rangle = b[k] \text{ and } \left\langle B^{(j)}, X\right\rangle \geq 0 \text{ for all } (j,k) \in [n] \times [m] \quad \text{and} \quad X \succeq 0\right\},$$

where the matrices are of the form

$$C = \begin{pmatrix} 0 & D \\ D^\top & 0 \end{pmatrix} \quad \text{and} \quad A^{(k)} = \begin{pmatrix} P^{(k)} & 0 \\ 0 & Q^{(k)} \end{pmatrix} \quad \text{and} \quad B^{(j)} = \begin{pmatrix} 0 & R^{(j)} \\ \left(R^{(j)}\right)^\top & 0 \end{pmatrix}$$

and there exists $u[j] \geq 0$ such that

$$D = \sum_{j=1}^{n} u[j] R^{(j)}.$$

Lee and Mittal showed that the two semidefinite programs of the above form (identified by subscripts 1 and 2 below) are multiplicative under the *tensor product operation* defined below. The cost matrix is the tensor product of the cost matrices of the individual semidefinite programs. The constraints are of the form

$$\left\langle A_1^{(k)} \otimes A_2^{(k')}, X \right\rangle = b[k] \cdot b[k'] \qquad \text{and} \qquad \left\langle B_1^{(j)} \otimes B_2^{(j')}, X \right\rangle \geq 0.$$

It is not hard to see that for the class of semidefinite programs considered in Section 7.3, the optimum value of the bipartite tensor product of two semidefinite programs is equal to the optimum value of the semidefinite program obtained from the product operation defined above. However, the multiplicative property of the class of semidefinite programs under the product operation defined in Ref. [LM08] is larger than the class of semidefinite programs considered in Section 7.3.

Although not explicitly stated, it is known that for a unique game G, if $\omega_q(G) = 1$ then $\omega_c(G) = 1$. One cannot hope for a quantitative relationship between these two quantities like the relationship we have for XOR games [CHTW04a]. Indeed, Khot and Vishnoi [KV05] explicitly constructed a unique game for which the classical value is close to zero but optimum of a semidefinite programming relaxation for the classical value is close to one. The optimal solution of the SDP relaxation can be used to extract a quantum strategy that is still close to one, which provides an arbitrary large separation between the two quantities.

Finally, Kempe and Vidick [KV11] have recently demonstrated an error reduction technique for general two-prover one-round quantum games. Informally speaking, their result

is as follows. If the quantum value of a two-prover one-round game G is at most α, then there exists a game G' defined in terms of repetitions of G with quantum value at most ϵ, where the number of repetitions depends polynomially on $(1-\alpha)^{-1}$ and ϵ^{-1}. The dependence implies that if, for instance, the quantum value of some game G is $1-1/n$, then one needs poly(n) repetitions to reduce the quantum value to a constant. They consider two different kinds of repetitions that defined game G' in terms of G: the Feige-Killian repetition [FK00] and the Dinur-Reingold repetition [DR06]. We point out that the game G' is *not* the standard n-fold repetition of the game G. We refer to their paper and the references therein for more detail on the types of repetitions and the proof technique employed.

Chapter 8

QMA variants with polynomially many provers

The chapter is based on the Ref. [GSU11] and is a joint work with Sevag Gharibian and Jamie Sikora.

The chapter introduces quantum Merlin-Arthur proof systems with multiple provers. Roughly speaking, multi-prover QMA proof systems are non-interactive proof system wherein each prover (Merlin) sends a polynomial-size proof that are unentangled with each other to the verifier (Arthur) who then runs a quantum verification procedure to decide whether to accept or reject. In this chapter, we are primarily concerned with three variants of multi-prover quantum Merlin-Arthur proof systems. The first variant is denoted $\mathrm{QMA}_{\log}(\mathrm{poly})$, the class of problems that can be efficiently verified by polynomially many unentangled quantum proofs each of logarithmic-size. The second variant is denoted BellQMA(poly), the class of problems that admits a verifier who first applies unentangled, nonadaptive measurements on each proof and then applies another operation on the outcomes of each measurement to decide whether to accept or reject. The third variant is SepQMA(poly), where the verifier's measurement operator corresponding to outcome *accept* is a *fully separable operator* across the polynomially many unentangled quantum proofs.

The organization of the chapter is as follows.

- In Section 8.1, we formally define multi-prover quantum Merlin-Arthur proof systems and their variants that aee relevant to this chapter.
- In Section 8.2, we discuss the main results of this chapter and brief overview of the proofs idea. We also give a short introduction to on conic programming relevant to this chapter.

- In Section 8.3, we show that $\mathrm{QMA}_{\log}(\mathrm{poly})$ is equivalent to MQA, the class of problems that admit a classical proof and a quantum verification procedure.
- In Section 8.4, we show that BellQMA(poly) is equivalent to QMA under the constraint that the number of outcomes after applying unentangled, nonadaptive measurements on each proof is at most a polynomial per proof.
- In Section 8.5, we provide an alternate proof of perfect parallel repetition theorem for SepQMA(poly), a result previously obtained by Harrow and Montanaro [HM10]. Our proof technique uses duality of cone programming and is similar in spirit to the techniques employed in previous two chapters.

8.1 Multiple prover quantum Merlin-Arthur proofs

This section introduces multi-prover quantum Merlin-Arthur proof systems. The study of non-interactive proof system in the classical world has yielded some of the greatest achievements in theoretical computer science, from the Cook-Levin theorem, which formally ushered in the age of NP verification systems and the now ubiquitous notion of NP-hardness, to the more modern PCP theorem [ALMSS98, AS98, Din07], which has led to significant advancements in our understanding of hardness of approximation.

A natural quantum generalization of the class NP, or more accurately its probabilistic analogue MA, is the class based on quantum Merlin-Arthur proof system [KSV02], where a computationally powerful but untrustworthy prover, Merlin, sends a *quantum* proof to convince an efficient *quantum* verifier, Arthur, that a given input string $x \in \{0,1\}^n$ belongs to the *yes*-instance of a specified promise problem. The proof system is at least as powerful as its classical counterparts, for the ability to process and exchange quantum information is no hindrance to Arthur's ability to act classically and prevent Merlin from using quantum information to his advantage.

The multi-prover variant of quantum Merlin-Arthur proof systems are non-interactive proof systems wherein multiple provers send quantum proofs that are guaranteed to be unentangled with each other. The complexity class based on such proof system is formally defined below.

Definition 40 ($\mathrm{QMA}(m)$)**.** *Let $p : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function, and $m : \mathbb{N} \to \mathbb{N}$ a function. A promise problem $A = (A_{yes}, A_{no})$ is in the class $\mathrm{QMA}(m)$ if there exists a polynomial-time generated family of verification circuits $Q = \{Q_n \mid n \in \mathbb{N}\}$ with the following properties:*

1. *Each Q_n acts on $n + p(n)$ input qubits, and outputs one qubit.*

2. *(Completeness) For every $x \in A_{yes}$, there exist $m(|x|)$ quantum proofs specified by the set $\{|\psi_i\rangle : i \in [m(|x|)]\} \subset \mathbb{C}^{2^{p(|x|)}}$ such that*

$$\Pr[Q_{|x|} \text{ accepts } (x, |\psi_1\rangle \otimes \ldots \otimes |\psi_{m(|x|)}\rangle)] \geq 2/3.$$

3. *(Soundness) For any $x \in A_{no}$ and any $m(|x|)$ quantum proofs specified by the set $\{|\psi_i\rangle : i \in [m(|x|)]\} \subset \mathbb{C}^{2^{p(|x|)}}$*

$$\Pr[Q_{|x|} \text{ accepts } (x, |\psi_1\rangle \otimes \ldots \otimes |\psi_{m(|x|)}\rangle)] \leq 1/3.$$

The class QMA(poly) *is defined as*

$$\text{QMA(poly)} = \bigcup_{m \in \text{poly}} \text{QMA}(m).$$

We point out a notational ambiguity here. We denote k-message quantum interactive proofs QIP(k) whereas we denote m-prover quantum Merlin-Arthur proofs by QMA(m). This will not lead to any ambiguity because multi-prover quantum Merlin-Arthur proof system are non-interactive and the number of messages exchanged between the provers and the verifier is always understood to be 1. We also note that the constants 2/3 and 1/3 can be replaced by any two real numbers a and b such that the difference between a and b is bounded from below by an inverse polynomial and one can amplify the two-sided errors to exponentially close to zero [HM10].

Few variants of QMA(m) mentioned in this chapter are listed below. All these classes satisfy the completeness and soundness conditions mentioned in Definition 40.

1. The complexity class QMA is simply QMA(1). The complexity class MQA is a subclass of QMA, where the proof is a polynomial-size classical string and the verification procedure is an efficient quantum computation.

2. The complexity class SepQMA(poly) is a subclass of QMA(poly), wherein Arthur's measurement operator corresponding to outcome "accept" is a fully separable operator across the proofs.

3. The complexity class $\text{QMA}_{\log}$(poly) is a subclass of QMA(poly), wherein each Merlin's message to Arthur is $O(\log(|x|))$ qubits.

We remark that MQA has been studied under the name QCMA in the literature — the notation MQA was suggested by Watrous [Wat09b]. It follows from definition that MA $\subseteq$ MQA $\subseteq$ QMA. A collapse of either of the two containments would be surprising,

as the first one would imply that quantum verification offers no advantage over classical verification and the second one would imply that quantum proofs do not offer any advantage over classical proofs in the non-interactive setting.

The expressive power of multiple Merlins was first studied by Kobayashi, Matsumoto and Yamakami [KMY03], who showed that QMA(2) = QMA(poly) if and only if the class of QMA(2) protocols with completeness c and soundness s (with at least inverse polynomial gap) is exactly equal to QMA(2) protocols with completeness 2/3 and soundness 1/3. A substantial amount of research has since been devoted to understanding the properties of multi-prover quantum Merlin-Arthur proof systems. Recently, Harrow and Montanaro [HM10] demonstrated a *product state test*, wherein given two copies of a *pure* quantum state on multiple systems, the test distinguishes between the cases when the quantum state is a *fully* product state across all the systems or *far* from any such state. Using this test, they answered a few important questions regarding QMA(poly). In particular, they showed that

$$\text{QMA}(2) = \text{QMA}(\text{poly})$$

and that error reduction is possible for such proof systems. Prior to their result, the answers to both the questions were known to be affirmative assuming a *weak* version of the Additivity Conjecture [ABDFS09]. One of the crucial properties of the product state test is that it can be converted into a QMA(2) protocol, where Arthur's measurement operator corresponding to outcome *accept* is a separable operator across the two proofs. Harrow and Montanaro established a perfect parallel repetition theorem for such proof systems, a crucial step in obtaining exponentially small error probabilities.

Blier and Tapp initiated the study of *logarithmic*-size quantum proofs [BT09] that are unentangled (a preprint of the paper first appeared in 2007). They showed that two unentangled quantum proofs suffice to show that a 3-coloring of an input graph exists, implying that NP has *succinct* unentangled quantum proofs. A drawback of their protocol is that although it has *perfect* completeness, its soundness is only inverse polynomially bounded away from 1. Shortly after, Aaronson, Beigi, Drucker, Fefferman and Shor [ABDFS09] showed that satisfiability of any 3-SAT formula of size n can be proven by $\widetilde{O}(\sqrt{n})$ unentangled quantum proofs of $O(\log n)$ qubits with perfect completeness and constant soundness (see also Chen and Drucker [CD10]). In a subsequent paper [Bei08], Beigi improved directly on Blier and Tapp's result [BT09] by showing that by sacrificing perfect completeness, one can show that NP has two logarithmic-size quantum proofs with a better gap between completeness and soundness probabilities than in [BT09].

Finally, one of the open questions raised in Ref. [ABDFS09] concerns the power of Arthur's verification procedure. In particular, the paper introduces two different classes of verification procedures, BellQMA and LOCCQMA verification. Roughly speaking, LOCCQMA verification corresponds to Arthur applying a measurement operation that can be implemented by Local Operations and Classical Communication (LOCC) (with respect to

the partition induced by the multiple proofs). The authors raised the question of whether BellQMA(poly) = QMA or not. Brandão [Bra08] showed that BellQMA(m) is equal to QMA for constant m. In a recent development, Brandão, Christandl and Yard [BCY11] showed that LOCCQMA(m) is equal to QMA for constant m.

A major focus of this chapter is on BellQMA verification, which is a two-stage verification procedure that proceeds as follows.

1. Arthur first performs measurement on each proof that are nonadaptive and unentangled with each other. The measurement need not be binary and the number of possible outcomes for each proof can be exponential.
2. Arthur's second step is to apply a quantum circuit on the measurement outcomes of each proof followed by measuring the first qubit of the resulting state. Depending on the measurement outcome, Arthur decides to accept or reject.

We remark that, as in the Ref. [CD10], our BellQMA protocols are allowed to use a *quantum* verification circuit in Stage 2, whereas originally in Refs. [Bra08, ABDFS09] only classical processing of measurement outcomes $\{ y_i \}$ was allowed in order to emulate the notion of a *Bell experiment* performed by Arthur. The result presented in Section 8.4 holds for both definitions. We now proceed on to define the class BellQMA(poly). The definition highlights the two-stage computation of the verifier and assumes that the completeness and soundness are 2/3 and 1/3, respectively. As stated above, they can be replaced by any two real numbers a and b such that the difference between a and b is bounded from below by an inverse polynomial without changing the expressive power of the class.

Definition 41. *Let $r, m : \mathbb{N} \to \mathbb{N}$ be two functions. A promise problem $A = (A_{yes}, A_{no})$ is in class* BellQMA$[r, m]$ *with completeness* 2/3 *and soundness* 1/3 *if there exists a quantum polynomial-time Arthur whose verification procedure is as follows.*

1. *Arthur first performs a polynomial-time quantum computation on the input and generates a description of quantum circuits $V_1(x), \ldots, V_m(x)$. The quantum circuits are independent and unentangled with each other.*
2. *Arthur's second step is to simultaneously apply $V_i(x)$ on the i-th quantum proof, the action of which can be described by a unitary operator followed by measurement in standard basis. At the end of this stage, Arthur obtains outcome $y_i \in [r(|x|)]$ for the i-th proof.*
3. *Arthur finally performs an efficient quantum computation following by a two-outcome measurement on the input and the measurement outcomes obtained in step 2, and decides to accept or reject based on the final outcome.*

The class BellQMA(m) *and* BellQMA(poly) *are defined as*

$$\text{BellQMA}(m) = \bigcup_{r \in \text{exp}} \text{BellQMA}[r, m] \qquad \textit{and} \qquad \text{BellQMA}(\text{poly}) = \bigcup_{m \in \text{poly}} \text{BellQMA}(m).$$

Note that the key distinction between BellQMA$[r, m]$ and BellQMA(poly) is that the former has the number of measurement outcomes in Stage 1 of the protocol bounded by $r(|x|)$, whereas the latter may allow exponentially many possible outcomes. Throughout this paper, we use the notation BellQMA[poly, poly] to denote

$$\text{BellQMA}[\text{poly}, \text{poly}] := \bigcup_{r \in \text{poly}} \bigcup_{m \in \text{poly}} \text{BellQMA}[r, m].$$

Finally, we remark that the class SepQMA(m) is equivalent to QMA(m). The equivalence between classes QMA(m) and SepQMA(m) was demonstrated by Harrow and Montanaro [HM10], where they used the equivalence to show that error amplification is possible for QMA(m) proof systems. This follows from their *product state test*, which shows that a QMA(m) protocol can be converted into a SepQMA(2) protocol.

8.2 Results and preliminaries

In this section, we discuss the main results and overview of the proof ideas adopted to obtain the results. We also briefly discuss cone programming and mention few relevant facts about them, which are implicitly used in Section 8.5.

8.2.1 Main results

We show the following three results regarding variants of QMA(poly).

1. Relationship to MQA. The first result is an easy observation that concerns with the expressive power of polynomially many logarithmic-size unentangled quantum proofs. We specifically show that

$$\text{QMA}_{\log}(\text{poly}) = \text{MQA}.$$

In other words, if each prover is restricted to sending short quantum proofs, then one can not only do away with multiple provers, but also of the need for *quantum* proofs altogether. The significance of this result is as follows: Understanding the expressive power of QMA(poly), or its relationship with QMA, is currently one of the biggest challenges in

quantum complexity theory. The above result settles this question in the logarithmic-size message setting.

The proof is based on a simple application of the fact that logarithmic-size quantum states can be described to within inverse exponential precision using a polynomial number of bits and efficiently prepared by a quantum circuit. Hence, roughly speaking, one can replace a polynomial number of logarithmic-size quantum proofs with a single polynomial-size classical proof. Each quantum proof can then be efficiently prepared from this classical information by the verifier to within inverse exponential precision since the original quantum proofs were of logarithmic-size. Although the proof is simple, one cannot hope for a better characterization using other techniques because the reverse containment, i.e. $\mathrm{MQA} \subseteq \mathrm{QMA}_{\log}(\mathrm{poly})$, also holds using similar ideas. In fact, MQA can be simulated by polynomially-many logarithmic-size quantum proofs, where each Merlin is supposed to send a bit of the optimal classical proof. We also note that Marriott and Watrous [MW05] showed that $\mathrm{QMA}_{\log}(1)$ is equal to BQP, a result that has been subsequently extended to two different models of single-prover quantum interactive proof systems, where the prover sends logarithmic-size message to the verifier [BSW11].

2. Towards a non-trivial upper bound on BellQMA(poly). Our second result concerns with BellQMA protocols. We show that for the class $\mathrm{BellQMA}[r, m]$, if $r, m : \mathbb{N} \to \mathbb{N}$ are polynomially bounded functions, then

$$\mathrm{BellQMA}[r, m] = \mathrm{QMA}.$$

Note that the number of outcomes after the first step of Arthur's verification procedure can be exponential in the input-size. The restriction that r has to be polynomially bounded function is crucial for our proof. One can view the result as a step towards answering whether the complexity classes BellQMA(poly) and QMA(poly) have the same expressive power or not.

The significance of BellQMA(poly) in our setting is that if one could show that $\mathrm{QMA} \neq \mathrm{BellQMA}(\mathrm{poly})$, then it would follow that $\mathrm{QMA} \neq \mathrm{QMA}(\mathrm{poly})$, since it holds that $\mathrm{QMA} \subseteq \mathrm{BellQMA}(\mathrm{poly}) \subseteq \mathrm{QMA}(\mathrm{poly})$. As mentioned above, Brandão has shown the negative result that $\mathrm{QMA} = \mathrm{BellQMA}(m)$ for *constant* m [Bra08]. Where BellQMA(poly) lies, however, remains open. For example, although we know $\mathrm{QMA}(2) = \mathrm{QMA}(\mathrm{poly})$ [HM10], the same techniques do not apply in any obvious way to show an analogous result for BellQMA protocols as they require entangled measurements (i.e., SWAP test measurements) across multiple proofs, which violate the definition of BellQMA.

The proof of this result is more technically challenging. To show the containment $\mathrm{BellQMA}[\mathrm{poly}, \mathrm{poly}] \subseteq \mathrm{QMA}$ (note that the reverse containment is trivial since $\mathrm{QMA} \subseteq \mathrm{BellQMA}[2, 1]$), we demonstrate a QMA protocol which simulates a $\mathrm{BellQMA}[\mathrm{poly}, \mathrm{poly}]$

protocol using the following observation: although consolidating m quantum proofs into a single quantum proof raises the possibility of cheating using entanglement, if Arthur is also sent an appropriate classical "consistency-check" string, then a dishonest Merlin can be caught with non-negligible probability.

Specifically, in our QMA protocol, we ask a single Merlin to send the m quantum proofs of the original BellQMA protocol (denoted by a single state $|\psi\rangle$), accompanied by a "consistency-check" string $\mathbf{p}$ which is a classical description of the probability distributions obtained as the output of Stage 1. One can think of this as having the QMA verifier *delegate* first stage of the BellQMA verification to Merlin. Arthur then performs a consistency check between $|\psi\rangle$ and $\mathbf{p}$ based on the premise that if Merlin is honest, then $\mathbf{p}$ should arise from running first stage of the original verification on $|\psi\rangle$. If this check passes, then Arthur runs second stage of the BellQMA verification on $\mathbf{p}$. If Merlin tries to cheat, however, we show that the check detects this with non-negligible probability. Note that the accuracy of the consistency check crucially uses the fact that there are at most polynomially many outcomes to check for each local measurement of first stage of the BellQMA verification.

A noteworthy observation based on the above two characterizations is that when restricted to BellQMA protocols with polynomial number of outcomes per proof, allowing each Merlin to send polynomially many qubits makes the model more powerful than the model where each Merlin is restricted to send logarithmic many qubits, unless MQA $\neq$ QMA.

3. Perfect parallel repetition for SepQMA(m). Our final result concerns with error reduction of SepQMA(m) protocols. We show that the property of *perfect* parallel repetition holds for SepQMA(m). This is in contrast with general QMA(m) protocols, where perfect parallel repetition does not hold. The property of perfect parallel repetition for SepQMA(m) protocols was previously proved by Harrow and Montanaro [HM10]. However, our proof technique is different from their technique. We use a simple cone program to characterize the success probability of SepQMA(m) protocol and use duality to show that perfect parallel repetition holds if the verifier's measurement operator is a separable operator across k proofs.

We remark that the duality of semidefinite programs have been used to show a perfect or strong parallel repetition for various models of (single or two-prover) quantum interactive proof systems as discussed in this thesis.

Finally, the following lemma is crucial to the proof of the first two results.

Lemma 42. [Wat02] *Let* $\{\rho_1 \ldots, \rho_k\} \subset \mathrm{D}(\mathcal{X})$ *and* $\{\sigma_1, \ldots, \sigma_k\} \subset \mathrm{D}(\mathcal{X})$. *Then*

$$\left\| \bigotimes_{i=1}^{k} \rho_i - \bigotimes_{i=1}^{k} \sigma_i \right\|_1 \leq \sum_{i=1}^{k} \|\rho_i - \sigma_i\|_1 .$$

8.2.2 Cone programming

This section briefly reviews basic notions in conic optimization (or cone programming), which is a generalization of semidefinite optimization. We say that a set $\mathbb{K}$ in an underlying Euclidean space is a cone if $x \in \mathit{cone}$ implies that $\lambda x \in \mathbb{K}$ for all $\lambda > 0$. A cone $\mathbb{K}$ is convex if $x, y \in \mathbb{K}$ implies that $x + y \in \mathbb{K}$. Cone programs are concerned with optimizing a linear function over the intersection of a convex cone and an affine space. It generalizes several well-studied models of optimization including semidefinite programming ($\mathbb{K} = \mathrm{Pos}(\mathcal{X})$) and linear programming ($\mathbb{K} = \mathbb{R}^n_+$). Associated with a cone $\mathbb{K}$ is its dual cone $\mathbb{K}^*$ defined as

$$\mathbb{K}^* = \{S : \langle X, S\rangle \geq 0 \text{ for all } X \in \mathbb{K}\}.$$

A cone program associates the following 4-tuple $(C, b, \mathcal{A}, \mathbb{K})$ to an optimization problem described as below.

$$\begin{aligned} \text{supremum:} \quad & \langle C, X\rangle \\ \text{subject to:} \quad & \mathcal{A}(X) = b, \\ & X \in \mathbb{K}, \end{aligned}$$

where $\mathcal{A} : \mathrm{Span}(\mathbb{K}) \to \mathbb{R}^M$ is a linear operator. transformation. Note that the inner product is defined as in the Euclidean space. For instance, if the cone under consideration is the set of positive semidefinite or separable operators, then the inner product is the standard Hilbert-Schmidt inner product over the space of Hermitian operators. As in the case of linear and semidefinite programs, cone programs also come in primal-dual pairs as described below.

Primal problem (P)		Dual problem (D)	
supremum:	$\langle C, X\rangle$	infimum:	$\langle b, y\rangle$
subject to:	$\mathcal{A}(X) = b,$	subject to:	$\mathcal{A}^*(y) = C + S,$
	$X \in \mathbb{K}.$		$S \in \mathbb{K}^*.$

The linear operator $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$ and is the unique map that satisfies

$$\langle Y, \mathcal{A}(X)\rangle = \langle \mathcal{A}^*(Y), X\rangle$$

for all X and Y in the underlying Euclidean space.

We say that the cone program is *primal feasible* and *strictly primal feasible*, respectively, if the sets

$$\{X : \mathcal{A}(X) = b\} \cap \mathbb{K} \qquad \text{and} \qquad \{X : \mathcal{A}(X) = b\} \cap \mathrm{int}(\mathbb{K})$$

are non-empty. Here $\mathrm{int}(\cdot)$ denotes the interior of a set.

One has a similar notion of *dual feasible* and *strictly dual feasible*. That is, the cone program is *dual feasible* and *strictly dual feasible*, respectively, if the sets

$$\{(y,S) : \mathcal{A}^*(y) = C + S\} \cap \mathbb{R}^m \oplus \mathbb{K}^* \qquad \text{and} \qquad \{(y,S) : \mathcal{A}^*(y) = C + S\} \cap \mathbb{R}^M \oplus \operatorname{int}(\mathbb{K}^*).$$

A convex cone $\mathbb{K}$ is closed if and only if $\mathbb{K} = \mathbb{K}^{**}$. Thus, if $\mathbb{K}$ is not closed we need to "order" the primal-dual pairs since the dual of the dual problem is no longer the primal problem. In this chapter, we are primarily concerned with the convex cone

$$\operatorname{Sep}(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m) = \left\{ \sum_{i=1}^{k} P_i(1) \otimes \cdots \otimes P_i(m) : \{P_1(j), \ldots, P_k(j)\} \subset \operatorname{Pos}(\mathcal{X}_j), k \in \mathbb{N} \right\}.$$

We call the elements of the set $\operatorname{Sep}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ *fully separable operator* and the set itself as the cone of fully separable operators. Note that the cone of fully separable operators is a closed convex cone so ordering the primal-dual pairs is not an issue.

Similar to linear programming and semidefinite programming, cone programming has a rich duality theory.

Lemma 43 (Weak Duality)**.** *If X is primal feasible and (y,S) is dual feasible then*

$$\langle b, y\rangle - \langle C, X\rangle \geq 0.$$

This result can be used to show upper bounds on the value of the primal problem or lower bounds on the value of the dual problem. There is also a notion of *strong duality*. We say that *strong duality holds for a problem* (P) if the optimal value of (P) equals the optimal value of (D) and (D) attains an optimal solution. Below we give a condition that guarantees strong duality for (P) when the underlying $\mathbb{K}$ lies in the space of Hermitian operators. The choice of this space is due to its relevance to Section 8.5 and the lemma holds as long as the cone $\mathbb{K}$ is a closed convex cone. Let α and β denote the optimum values for primal and dual problems, respectively.

Lemma 44. *Let $\mathbb{K}$ be a closed convex cone, $C \in \operatorname{Herm}(\mathcal{X})$, $b \in \mathbb{R}^M$, and $\mathcal{A} : \mathrm{L}(\mathcal{X}) \to \mathbb{C}^M$ be a map that maps every operator in $\operatorname{Herm}(\mathcal{X})$ to $\mathbb{R}^M$. Then the following holds.*

1. *Let β be finite and that there exists an operator $X' \in \operatorname{int}(\mathbb{K})$ such that $\mathcal{A}(X') = b$. Then $\alpha = \beta$ and there exists $(y,S) \in \mathbb{R}^M \oplus \mathbb{K}^*$ such that*

$$\mathcal{A}^*(y) = C + S \qquad \text{and} \qquad \langle b, y\rangle = \beta.$$

2. *Let α be finite and that there exists $(y',S') \in \mathbb{R}^M \oplus \operatorname{int}(\mathbb{K}^*)$ such that $\mathcal{A}^*(y') = C + S'$.*

Then $\alpha = \beta$ and there exists $X \in \mathbb{K}$ such that

$$\mathcal{A}(X) = b \qquad \textit{and} \qquad \langle C, X \rangle = \alpha.$$

This result can be used to show upper bounds on the value of the primal problem or lower bounds on the value of the dual problem. There is also a notion of *strong duality*. There are several conditions that guarantee strong duality for cone programs and we refer to Tunçel and Wolkowicz [TW08] and the references therein for detail.

We finally remark that unlike linear programming or semidefinite programing, cone programming are not efficiently solvable. There exist NP-hard problems that can be represented as instances of cone programs.

8.3 Equivalence of MQA and $\mathbf{QMA_{log}(poly)}$

In this section, we prove the equivalence between MQA and $\mathrm{QMA}_{\log}(\mathrm{poly})$. We first show the direction MQA $\subseteq \mathrm{QMA}_{\log}(\mathrm{poly})$. Let $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ be a promise problem in MQA and let $x \in \{0,1\}^n$ be the input string. Suppose the MQA prover sends an m-bit classical proof to the verifier, for polynomially bounded m. Then the following simple $\mathrm{QMA}_{\log}(m)$ protocol achieves the desired containment:

$\underline{\mathrm{QMA}_{\log}(m) \text{ Protocol}}$

1. **Embed classical bits into qubits.** Each (unentangled) prover $i \in [m]$ sends a single qubit $|\psi_i\rangle \in \mathbb{C}^2$ to Arthur. If the i-th prover is honest, his/her qubit is the computational basis state corresponding to the i-th bit of the classical MQA proof.
2. **Make things classical again.** Arthur measures all proofs in the computational basis, obtaining a classical string $y \in \{0,1\}^m$.
3. **Run MQA verification.** Arthur runs the MQA verification circuit on x and y and accepts if and only if acceptance occurs in the MQA verification.

The completeness property follows straightforwardly. The soundness property is also easy to observe. Note that Arthur runs the MQA verification on a classical string y and hence he accepts the string with probability at most $1/3$.

To show the reverse containment, let $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ be a problem in $\mathrm{QMA}_{\log}(\mathrm{poly})$ and let $x \in \{0,1\}^n$ be the input string. Suppose we have a $\mathrm{QMA}_{\log}(m)$ protocol for polynomially bounded m, where prover i sends a $\lceil c \log n \rceil$-qubit state $|\psi_i\rangle$ for some constant $c > 0$. Let

$$r(n) = 2^{\lceil c \log n \rceil} = O(n^c).$$

The MQA protocol proceeds as follows:

MQA Protocol

1. **Describe proofs classically.** The prover sends m classical registers represented by the tuple $(\mathsf{C}_1, \mathsf{C}_2, \ldots, \mathsf{C}_m)$, each of length $2n \cdot r(n)$ to Arthur. If the prover is honest, register C_i contains a classical description of the i-th quantum proof of the $\mathrm{QMA}_{\log}(m)$ protocol.
2. **State preparation.** Using the contents of register C_i, for every choice of $i \in [m]$, Arthur prepares the state $|\psi_i\rangle$ by first determining a unitary U_i such that $U_i |0 \ldots 0\rangle = |\psi_i\rangle$, and then implementing U_i with high precision using a finite set of approximately universal gates, obtaining states $|\psi_i'\rangle$.
3. **Run $\mathrm{QMA}_{\log}(m)$ verification.** Arthur runs the $\mathrm{QMA}_{\log}(m)$ verification circuit on $|\psi_1'\rangle \otimes \cdots \otimes |\psi_m'\rangle$ and accepts if and only if acceptance occurs in $\mathrm{QMA}_{\log}(m)$ verification.

Observe that each classical register C_i is of size polynomial in n, implying the overall proof length is of polynomial size. In Step 1, the prover uses n bits to represent the real and imaginary parts of each of the polynomially many entities ($r(n)$ entries) required to describe each $|\psi\rangle$. Let the unit vector described by register C_i be denoted $|\psi_i\rangle$. In Step 2, U_i is easily found as the unitary that maps $|0 \ldots 0\rangle$ to $|\psi_i\rangle$ as the inverse of the unitary that maps $|\psi_i\rangle$ to $|0 \ldots 0\rangle$. Such a unitary can be easily decomposed into a product of polynomially many 2×2 rotations on an $r(n)$-dimensional real space and a diagonal unitary as follows. The first step is to convert the vector $|\psi_i\rangle$ into a real vector by applying an appropriate diagonal unitary operator. The second step is to convert the resulting real unit vector into $|0 \ldots 0\rangle$ by shifting the amplitudes of any standard basis other than $|0 \ldots 0\rangle$ to $|0 \ldots 0\rangle$. Each of these unitary operators can be implemented by a finite set of approximately universal gates (see Bernstein and Vazirani [BV97] for details). This step also incurs some error, which can be made exponentially small.

Since Steps 1 and 2 can be performed to within inverse exponential error, we thus can ensure $\| |\psi_i\rangle - |\psi_i'\rangle \| \leq \epsilon$ for all $i \in [m]$ and for inverse exponential $\epsilon > 0$. By Lemma 42, it follows that the overall precision error is at most $m\epsilon$ for polynomial m, and thus the completeness and soundness of the protocol are bounded from below and above by (respectively)

$$\frac{2}{3} - m\epsilon \qquad \text{and} \qquad \frac{1}{3} + m\epsilon.$$

Alternatively, the containment $\mathrm{QMA}_{\log}(\mathrm{poly}) \subseteq \mathrm{MQA}$ can be shown using a slightly different protocol[1], where Merlin sends classical descriptions of the quantum circuits that generate the quantum proofs from $|0 \ldots 0\rangle$ instead of classical descriptions of the proofs.

[1]This protocol was mentioned to us by Richard Cleve.

8.4 Equivalence of BellQMA[poly, poly] and QMA

Recall that BellQMA$[r, m]$ is the class of problems, where the verifier receives m unentangled proofs from m Merlins, applies a quantum circuit $V_j(x)$ on j-th proof that can be physically interpreted as performing unentangled, nonadaptive measurement on each proof with r possible outcomes from the set $[r]$, and then applies a quantum circuit on the measurement outcomes to decide whether to accept or reject. One can assume without loss of generality that the number of possible outcomes is exactly r for each prover. In this section, we show that if r, m are polynomially-bounded functions, then

$$\text{QMA} = \text{BellQMA}[r, m].$$

For notational convenience, let $\Pi_j(i)$ denote Arthur's i-th POVM element in the first stage of BellQMA verification protocol for the j-th prover.

We proceed as follows. Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, and x be an input string of length $n := |x|$. As mentioned in Section 8.2.1, the containment QMA $\subseteq$ BellQMA[poly, poly] follows straightforwardly since QMA $\subseteq$ BellQMA$[2, 1]$. For the reverse containment, suppose we have a BellQMA$[r, m]$ protocol for polynomially bounded functions $r, m : \mathbb{N} \to \mathbb{N}$ with completeness $2/3$ and soundness $1/3$. We show that this protocol can be simulated by a QMA protocol where Merlin sends the following proof to Arthur.

Merlin sends two registers (X, Y), which should be thought of as the *classical* and *quantum* registers, respectively. Suppose optimal proofs for the BellQMA$[r, m]$ protocol for input x are given by ρ_j for $j \in [m]$. Then, in the quantum register Y, an honest Merlin should send many copies of the state ρ_j. Specifically, Y is partitioned into m registers Y_j, one for each original prover, and each Y_j should contain k copies of ρ_j, for k a carefully chosen polynomial. In other words, Y should contain the state $[\rho_1^{\otimes k}]_{\mathsf{Y}_1} \otimes \cdots \otimes [\rho_m^{\otimes k}]_{\mathsf{Y}_m}$. We further view each Y_j as a block of registers $(\mathsf{Y}_j^1, \ldots, \mathsf{Y}_j^k)$ where Y_j^l should contain the l-th copy of ρ_j.

In the classical register X, an honest Merlin prepares a quantum state in the computational basis, which intuitively corresponds to a bit string describing the m classical probability distributions Arthur induces upon applying the measurement operation corresponding to first stage of the BellQMA verification to each of the optimal proofs ρ_j, respectively.

More formally, we partition X into mr registers X_j^i corresponding to each of the $j \in [m]$ Merlins and $i \in [r]$ POVM outcomes for every Merlin. In the scenario when Merlins are honest, the content of X_j^i should be

$$p_j(i) := \langle \Pi_j(i), \rho_j \rangle$$

truncated to s bits of precision (s being polynomially bounded function), such that the set $\{p_j(1), \ldots, p_j(r)\}$ form a probability distribution. For example, if the j-th prover's proof was the single qubit state $\rho_j = |0\rangle\langle 0|$, with $\Pi_j(1) = |0\rangle\langle 0|$ and $\Pi_j(2) = |1\rangle\langle 1|$, then $\mathsf{X}_j = (1, 0)$. We remark that X plays the role of the classical "consistency check" string described in Section 8.2.1.

Of course, Merlin may elect to be dishonest and choose not to send a proof of the above form to Arthur by (let's say) sending a quantum state which is entangled across the registers (X, Y). To catch this, our QMA protocol is defined as follows:

Of course, Merlin may elect to be dishonest and choose not to send a proof of the above form to Arthur by, e.g., sending a quantum state which is entangled across the registers (X, Y). To catch this, our QMA protocol is defined as follows:

QMA Protocol

1. Merlin sends Arthur a quantum state in registers (X, Y), for X and Y defined as above.
2. **Force X to be classical.** Arthur measures register X in the computational basis and reads the measurement outcome. This forces X to essentially be a classical register of bits, and destroys any entanglement or correlations between X and Y.
3. **X should contain probability distributions.** Arthur checks whether the content of registers X_j form a probability distribution p_j, i.e., that $\sum_{i=1}^{r} p_j(i) = 1$. Arthur rejects if this is not the case.
4. **Consistency check: Can the quantum states in Y reproduce the distributions in X?** Arthur picks independently and uniformly at random, an index $j \in [m]$ and another index $i \in [r]$. He applies the measurement $\{\Pi_j(i)\}_{i=1}^{r}$ separately to each register $\mathsf{Y}_j^1, \ldots, \mathsf{Y}_j^k$, and counts the number of times outcome i appears, which we denote henceforth as $n_j(i)$. Arthur rejects if
$$\left| \frac{n_j(i)}{k} - p_j(i) \right| \geq \frac{1}{p}$$
for p a carefully chosen polynomial.
5. **Run Stage 2 of the BellQMA verification and repeat for error amplification.** For each prover j, Arthur samples an outcome from $[r]$ according to the distribution in $(\mathsf{X}_j^1, \ldots, \mathsf{X}_j^r)$, and runs Stage 2 of the BellQMA verification on the resulting set of samples. He repeats this process independently a polynomial number of times q, and accepts if and only if the BellQMA procedure accepts on the majority of the runs.

Let us give an intuition behind the above verification procedure. The key step above is Step 4, where Arthur cross-checks that the classical distributions sent in X really can

be obtained by measuring m quantum proofs, which for an honest Merlin should be unentangled. In this sense, our protocol can alternatively be viewed as using *quantum* proofs (Y) to check validity of a *classical* proof (X). Intuitively, the reason why entanglement in Y does not help a dishonest Merlin in Step 3 is due to the local nature of Arthur's checks/measurements. Finally, once Arthur is satisfied that X contains valid distributions, he runs Step 5. We remark that repetition is used here in order to boost the probability of acceptance in the $x \in A_{\text{yes}}$ case to exponentially close to 1, which is required to separate it from the $x \in A_{\text{no}}$ case, where the probability of catching a dishonest Merlin is only inverse polynomially bounded away from 1. Once such a gap exists, standard amplification techniques [KW00, MW05] can be used to further improve completeness and soundness parameters.

To formally analyze completeness and soundness of the protocol, we assign the following values to the parameters mentioned above, all of which are polynomial in n in our setting:

$$q = 50n \qquad \text{and} \qquad p = 20mr \qquad \text{and} \qquad k = 5p^3 \qquad \text{and} \qquad s = 20nmr.$$

We now analyze the completeness and soundness probabilities of the QMA protocol below.

Completeness. Intuitively, when $x \in A_{\text{yes}}$, Merlin passes Step 4 with probability exponentially close to 1 since he has no incentive to cheat — he can send an unentangled proof in Step 1 to Arthur corresponding to the optimal proofs ρ_j in the BellQMA protocol, such that the expected value of $n_j(i)/k$ is indeed $p_j(i)$. Arthur's checks in Step 4 are then independent local trials, allowing a Chernoff bound to be applied. We then show that Merlin passes each run in Step 5 with constant probability, and applying the Chernoff bound a second time yields the desired completeness exponentially close to 1 for the protocol.

To state this formally, suppose Merlin is honest and sends registers (X, Y) in the desired form, i.e., X_j^i contains $p_j(i) = \langle \Pi_j(i), \rho_j \rangle$ up to α bits of precision, and Y_j^i contains ρ_j. Then, the expected value of the random variable $n_j(i)$ is $\mathbb{E}[n_j(i)] = k \langle \Pi_j(i), \rho_j \rangle$, which is equal to $k \cdot p_j(i)$ up to the error incurred by representing $p_j(i)$ using α bits of precision. In other words,

$$\left| \frac{\mathbb{E}[n_j(i)]}{k} - p_j(i) \right| < \frac{1}{2^s} < \frac{1}{2p}. \tag{8.1}$$

We can hence upper bound the probability of rejecting in Step 3 by

$$\Pr\left[\left| \frac{n_j(i)}{k} - p_j(i) \right| \geq \frac{1}{p} \right] < \Pr\left[\left| \frac{n_j(i)}{k} - \frac{\mathbb{E}[n_j(i)]}{k} \right| \geq \frac{1}{2p} \right] \leq 2 \exp\left(-\frac{5p}{4} \right)$$

where the first inequality follows from (8.1) and the second from the Chernoff bound. Thus, Merlin passes Step 4 with probability exponentially close to 1.

We now turn to the final step. Since $x \in A_{\text{yes}}$, we know that the optimal distributions, denoted $q_j := (\langle \Pi_j(1), \rho_j \rangle, \ldots, \langle \Pi_j(r), \rho_j \rangle)$ for $j \in [m]$, obtained in first stage of the original BellQMA protocol are now accepted in Stage 2 with probability at least $2/3$. However, in our case, Merlin was only able to specify each q_j up to α bits of precision per entry as the distributions p_j. To analyze how this affects the probability of acceptance, let P_j and Q_j be diagonal operators with entries $P_j(i,i) = p_j(i)$ and $Q_j(i,i) = \langle \Pi_j(i), \rho_j \rangle$, respectively. Letting Λ_{accept} denote the POVM element corresponding to outcome *accept* in second stage of the BellQMA protocol, we thus bound the change in acceptance probability by:

$$\begin{aligned}
\left| \operatorname{Tr}\left[\Lambda_{\text{accept}} \left(\bigotimes_{j=1}^{m} P_j - \bigotimes_{j=1}^{m} Q_j \right) \right] \right| &\leq \left\| \bigotimes_{j=1}^{m} P_j - \bigotimes_{j=1}^{m} Q_j \right\|_{\text{tr}} \\
&\leq \sum_{j=1}^{m} \| P_j - Q_j \|_1 \\
&= \sum_{j=1}^{m} \sum_{i=1}^{r} |p_j(i) - \langle \Pi_j(i), \rho_j \rangle| \\
&\leq \frac{mr}{2^{20nmr}}
\end{aligned}$$

where the first inequality follows from the fact that $|\operatorname{Tr}(AB)| \leq \|A\|_\infty \cdot \|B\|_1$ and the second inequality follows from Lemma 42. Therefore, the probability of success for each of the q runs of the BellQMA protocol in Step 5 is at least

$$\left(\frac{2}{3} - \frac{mr}{2^{20nmr}} \right) > 0.6.$$

Since each run is independent, applying the Chernoff bound yields that Arthur accepts Merlin's proof in Step 5 with probability at least $1 - 2\exp(-0.02q)$, as desired. There may be some error incurred in sampling, which can be assumed to be exponentially small so that the success probability of each run is still at least 0.6.

Soundness. We now prove that when $x \in A_{\text{no}}$, a dishonest Merlin can win with probability at most inverse polynomially bounded away from 1. To show this, we bound the probability of passing Step 4 by relating the quantity $p_j(i)$ to the expected value of $n_j(i)/k$, and then apply the Markov bound. The desired relationship follows by observing first that the expected value of $n_j(i)/k$ is precisely the probability of obtaining outcome i when measuring proof j of some (honest) unentangled strategy, followed by arguing that the distribution p_j must hence be far from this latter (honest) distribution if Merlin is to pass Step 5 with probability at least $1/2$ (since $x \in A_{\text{no}}$). Combining these facts, we find that Arthur detects a cheating Merlin with inverse polynomial probability in Step 4.

More formally, let the quantum register Y_j contain an arbitrary quantum state σ_j whose reduced states in registers Y_j^l for $l \in [k]$ are given by $\sigma_j(l)$, and define

$$\xi_j := \frac{1}{k} \sum_{l=1}^{k} \sigma_j(l).$$

By the linearity of expectation, the expected value of the random variable $n_j(i)/k$ is

$$\mathbb{E}\left[\frac{n_j(i)}{k}\right] = \frac{1}{k} \sum_{l=1}^{k} \langle \Pi_j(i), \sigma_j(l) \rangle = \langle \Pi_j(i), \xi_j \rangle .$$

Our goal is to lower bound the expression

$$\Pr\left[\left| \frac{n_j(i)}{k} - p_j(i) \right| \geq \frac{1}{p}\right]. \tag{8.2}$$

To achieve this, we first substitute $p_j(i)$ above with a quantity involving $\mathbb{E}[n_j(i)/k]$, and then apply the Markov bound.

To relate $\mathbb{E}[n_j(i)/k]$ to $p_j(i)$, we first remark that in order for Merlin to pass each run of Step 5 with probability exponentially close to 1, he must send probability distributions p_j, which are accepted by Stage 2 of the BellQMA verification with probability at least $1/2$. Let

$$q_j(i) := \langle \Pi_j(i), \xi_j \rangle .$$

Let us imagine a BellQMA protocol where the j-th Merlin sends ξ_j as his quantum proof. Since $x \in A_{\text{no}}$, by the soundness property of the BellQMA(m) proof system, the success probability of the Merlins is at most $1/3$. In other words, sampling outcomes from the probability distributions $(q_j(1), \ldots, q_j(r))$ and then running the second stage of BellQMA verification procedure will yield outcome *accept* with probability at most $1/3$. Also, observe that

$$\mathbb{E}\left[\frac{n_j(i)}{k}\right] = q_j(i).$$

It follows that by letting P_j and Q_j be diagonal operators with the probability vectors p_j and q_j on their diagonals, respectively, and Λ_{accept} the POVM element corresponding to outcome *accept* in second stage of the BellQMA protocol, we have

$$\frac{1}{10} < \left| \operatorname{Tr}\left[\Lambda_{\text{accept}} \left(\bigotimes_{j=1}^{m} P_j - \bigotimes_{j=1}^{m} Q_j \right)\right] \right| \leq \left\| \bigotimes_{j=1}^{m} P_j - \bigotimes_{j=1}^{m} Q_j \right\|_1 \leq \sum_{j=1}^{m} \| P_j - Q_j \|_1 .$$

Here, the (loose) lower bound of $1/10$ comes from the following two observations. First,

the distributions represented by the diagonal operators Q_j's are derived from a BellQMA protocol and therefore achieve a success probability at most $1/3$ by the soundness property of the BellQMA verification. Second, the distributions represented by the diagonal operators P_j's have to achieve a success probability strictly greater than $1/2$ per run to guarantee that Merlin wins Step 5 with probability exponentially close to 1. Combining these two, we get that the difference between the success probabilities obtained by distributions described by operators $\{ P_j : j \in [m] \}$ and $\{ Q_j : j \in [m] \}$ should be at least $1/6$ modulo the error incurred due to finite precision when encoding the distributions p_j. The use of the constant $1/10$ overcompensates for this precision error. Hence, there exists a j such that

$$\| P_j - Q_j \|_1 = \sum_{i=1}^{r} |p_j(i) - q_j(i)| \geq \frac{1}{10m}$$

implying the existence of an i such that

$$|p_j(i) - q_j(i)| \geq \frac{1}{10mr}. \tag{8.3}$$

This is our desired relationship between $p_j(i)$ and $\mathbb{E}[n_j(i)/k] = q_j(i)$. Note that the probability of picking pair (i, j) in Step 4 is $1/mr$.

We now substitute this relationship into Eq. (8.2) and apply the Markov bound. Specifically, choose i and j as in Eq. (8.3), and assume that $p_j(i) > \langle \Pi_j(i), \xi_j \rangle$. Then, we have

$$\Pr\left[\left| \frac{n_j(i)}{k} - p_j(i) \right| < \frac{1}{p} \right] < \Pr\left[\frac{n_j(i)}{k} - \mathbb{E}\left[\frac{n_j(i)}{k} \right] > \frac{1}{10mr} - \frac{1}{p} \right] \leq 1 - \frac{1}{2p}.$$

The case when $p_j(i) < \langle \Pi_j(1), \xi_j \rangle$ is similar. We conclude that a dishonest Merlin is caught in Step 4 with probability at least $1/2p$. Therefore, the probability that Arthur proceeds to Step 5 is upper bounded by

$$\left(\frac{1}{mr} \right) \left(1 - \frac{1}{20mr} \right) + \left(1 - \frac{1}{mr} \right) (1) = 1 - \frac{1}{20m^2r^2}$$

where the first term represents the case where Arthur selects the correct pair (i, j) to check, and the second term the complementary case, in which we assume the cheating prover can win with probability 1. Hence the overall success probability of a dishonest Merlin is at most $1 - 1/20m^2r^2$, which is bounded away from 1 by an inverse polynomial.

Finally, as mentioned before, since m and r are polynomially bounded functions, we have that the completeness is exponentially close to 1, while the soundness is bounded away from 1 by an inverse polynomial. By known amplification techniques for QMA protocols [KW00, MW05], one can amplify the completeness and soundness errors to be

exponentially close to 0. This proves our desired containment.

We note that the result of Brandão that BellQMA(k) = QMA for constant k is not subsumed by our result. While Brandão's result works only for constant k, it does not make any assumption on the number of possible outcomes at stage 1 of the BellQMA verification. Although our result works for polynomially many Merlins, however, it crucially uses the fact that the number of outcome per proof.

8.5 Perfect parallel repetition for SepQMA(poly)

In this section, we answer the following question on SepQMA(poly) protocols. Before we proceed, recall that the closed convex cone $\mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right)$ is defined to contain operators of the form

$$\sum_{i=1}^{k} P_1(i) \otimes \cdots \otimes P_m(i)$$

where $P_j(i) \in \mathrm{Pos}\left(\mathcal{X}_j\right)$, for every $j \in [m]$ and $i \in [k]$. This is the cone of interest and it is known to be closed and convex with non-empty interior. Given C to be the measurement operator corresponding to outcome *accept*, the maximum success probability of the Merlins in any QMA(m) protocol can be written as the maximum of $\langle \rho, C \rangle$, where ρ is a density operator in $\mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right)$. By standard convexity argument, one can always assume that the maximum is achieved by a pure product state.

Suppose we have two instances of a SepQMA(m) protocol, the main question is to characterize the optimal strategy of both instances of the protocol in terms of the optimal strategy of a single instance. We begin by recalling that for vector spaces $\mathcal{X}_1, \ldots, \mathcal{X}_k$ we have the cone containments

$$\mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right) \subseteq \mathrm{Pos}\left(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m\right) \subseteq \mathrm{Sep}^*(\mathcal{X}_1, \ldots, \mathcal{X}_m) \subseteq \mathrm{Herm}\left(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m\right).$$

It will be convenient for us to distinguish the two instances of the SepQMA(m) protocols as the first and second protocol. For the first SepQMA(k) protocol we can write the maximum acceptance probability as the optimal value of the primal problem as the following primal-dual cone porgram:

Primal problem (P_1)		Dual problem (D_1)	
maximize:	$\langle C_1, X_1 \rangle$	minimize:	t_1
subject to:	$\mathrm{Tr}(X_1) = 1,$	subject to:	$t_1 \mathbb{I}_{\mathcal{X}} = C_1 + W_1,$
	$X \in \mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right).$		$W_1 \in \mathrm{Sep}^*(\mathcal{X}_1, \ldots, \mathcal{X}_k).$

Here $\mathcal{X}$ denotes $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m$ for notational convenience. The use of "maximum" and "minimum" is justified in the above program. Notice that

$$\hat{X}_1 := \frac{\mathbb{I}_{\mathcal{X}}}{\dim(\mathcal{X})} \qquad \text{and} \qquad (\hat{t}_1, \hat{W}_1) := (2, 2\mathbb{I}_{\mathcal{X}} - C_1)$$

are strictly feasible solutions for P_1 and D_1, respectively. That the normalized identity operator is in the interior of the cone of fully separable operators follows from the fact that a ball around the normalized identity operator is contained in the cone of fully separable operators [GB02, GB03, GB05]. Moreover, the value of P_1 and D_1 are bounded from above and below by 1, respectively. Hence by Lemma 44, strong duality holds and there exist primal and dual feasible solutions that achieve the optimum objective value. Indeed, since the primal problem is an optimization problem of a linear function over a compact set, the optimum value is attained by a primal feasible solution.

We can similarly formulate the acceptance probability of the second instance of the protocol as

Primal problem (P_2)		Dual problem (D_2)	
maximize:	$\langle C_2, X_2 \rangle$	minimize:	t_2
subject to:	$\mathrm{Tr}(X_2) = 1,$	subject to:	$t_2 \mathbb{I}_{\mathcal{Y}} = C_2 + W_2,$
	$X_2 \in \mathrm{Sep}\left(\mathcal{Y}_1, \ldots, \mathcal{Y}_m\right).$		$W_2 \in \mathrm{Sep}^*(\mathcal{Y}_1, \ldots, \mathcal{Y}_m).$

For the sake of clarity, we index the two instances of the protocol by subscripts 1 and 2. We let $\mathcal{Y}$ denote $\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m$. The success probability of the two-fold repetition of SepQMA(m) protocol is given by the following primal-dual pair.

Primal problem (P)

maximize:	$\langle C_1 \otimes C_2, X \rangle$
subject to:	$\mathrm{Tr}(X) = 1,$
	$X \in \mathrm{Sep}\left(\mathcal{X} \otimes \mathcal{Y}_1, \ldots, \mathcal{X} \otimes \mathcal{Y}_m\right).$

Dual problem (D)

minimize:	t
subject to:	$t\mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}} = C_1 \otimes C_2 + W,$
	$W \in \mathrm{Sep}^*(\mathcal{X} \otimes \mathcal{Y}_1, \ldots, \mathcal{X} \otimes \mathcal{Y}_m).$

We must make sure that the operators in the above program are defined in the correct operator spaces to make the inner products in the objective functions well-defined.

Observe that if X_1 and X_2 are any respective optimal solutions of (P_1) and (P_2), then $X_1 \otimes X_2$ is a primal feasible solution of P. Therefore the optimal value of (P) is at least the product of the optimal values of (P_1) and (P_2). It remains to show that in fact *no* other strategy for the prover can perform better than this honest strategy. To do so, we demonstrate a dual feasible solution for (D) attaining this same objective value.

More formally, let (t_1, W_1) and (t_2, W_2) be respective dual optimal solutions of (D_1) and (D_2). By strong duality, t_1 is the optimal value of (P_1) and t_2 is the optimal value of (P_2). We show that $t_1 \cdot t_2$ is an upper bound on the optimal value of (P) by exhibiting a solution $(t_1 \cdot t_2, W)$ which is feasible in (D), for some $W \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$. We first prove the following useful lemma.

Lemma 45. *For any $W_1 \in \mathrm{Sep}^*(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ and $C_2 \in \mathrm{Sep}(\mathcal{Y}_1, \ldots, \mathcal{Y}_m)$, we have*

$$W_1 \otimes C_2 \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m) \subseteq \mathrm{Herm}\left(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m \otimes \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m\right).$$

Similarly, for any $W_2 \in \mathrm{Sep}^*(\mathcal{Y}_1, \ldots, \mathcal{Y}_m)$ and $C_1 \in \mathrm{Sep}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$, we have

$$C_1 \otimes W_2 \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m) \subseteq \mathrm{Herm}\left(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m \otimes \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m\right).$$

Proof. We prove the first condition only, since the second condition is nearly identical. For $S \in \mathrm{Sep}(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$, we have

$$\langle W_1 \otimes C_2, S \rangle = \langle W_1, \mathrm{Tr}_{\mathcal{Y}}\left[S(\mathbb{I}_{\mathcal{X}} \otimes C_2)\right] \rangle \geq 0$$

if $\mathrm{Tr}_{\mathcal{Y}}\left[(\mathbb{I}_{\mathcal{X}} \otimes C_2)S\right] \in \mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right)$. All we need to show is that $\mathrm{Tr}_{\mathcal{Y}}\left[(\mathbb{I}_{\mathcal{X}} \otimes C_2)S\right] \in \mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right)$. To this end, let

$$S = \sum_i \rho_i(1) \otimes \cdots \otimes \rho_i(m) \qquad \text{and} \qquad C_2 = \sum_j \sigma_j(1) \otimes \cdots \otimes \sigma_j(m)$$

where $\rho_i(k) \in \mathrm{Pos}\left(\mathcal{X}_k \otimes \mathcal{Y}_k\right)$ for all choices of i and k, and $\sigma_j(l) \in \mathrm{Pos}\left(\mathcal{Y}_l\right)$ for all choices of j and l. Now

$$\begin{aligned}
\mathrm{Tr}_{\mathcal{Y}}\left[\left(\mathbb{I}_{\mathcal{X}} \otimes C_2\right) S\right] &= \mathrm{Tr}_{\mathcal{Y}}\left[\left(\sum_i \bigotimes_{k=1}^m \rho_i(k)\right)\left(\mathbb{I}_{\mathcal{X}} \otimes \sum_j \bigotimes_{l=1}^m \sigma_j(l)\right)\right] \\
&= \sum_{i,j} \bigotimes_{k=1}^m \mathrm{Tr}_{\mathcal{Y}_k}\left[\rho_i(k)\left(\mathbb{I}_{\mathcal{X}_k} \otimes \sigma_j(k)\right)\right]
\end{aligned}$$

which is clearly in $\mathrm{Sep}\left(\mathcal{X}_1, \ldots, \mathcal{X}_m\right)$. This concludes the proof.

□

The above lemma implies the following cone containments:

1. $\mathrm{Sep}^*(\mathcal{X}_1, \ldots, \mathcal{X}_m) \otimes \mathrm{Sep}(\mathcal{Y}_1, \ldots, \mathcal{Y}_m) \subseteq \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$.
2. $\mathrm{Sep}(\mathcal{X}_1, \ldots, \mathcal{X}_m) \otimes \mathrm{Sep}^*(\mathcal{Y}_1, \ldots, \mathcal{Y}_m) \subseteq \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$.

Here $\mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m) \subseteq \mathrm{Herm}\,(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m \otimes \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m)$.

We now use Lemma 45 to construct two operators in $\mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$, the appropriate convex combination of which is the dual feasible solution we are seeking. Specifically, observe first that since for the two instances of the SepQMA(m) protocol, we have $C_1 \in \mathrm{Sep}\,(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ and $C_2 \in \mathrm{Sep}\,(\mathcal{Y}_1, \ldots, \mathcal{Y}_m)$, and since $\mathbb{I}_\mathcal{X}$ and $\mathbb{I}_\mathcal{Y}$ are fully separable operators, it follows that

$$t_1\mathbb{I}_\mathcal{X} + C_1 \in \mathrm{Sep}(\mathcal{X}_1, \ldots, \mathcal{X}_m) \qquad \text{and} \qquad t_2\mathbb{I}_\mathcal{Y} + C_2 \in \mathrm{Sep}(\mathcal{Y}_1, \ldots, \mathcal{Y}_m)$$

for all $t_1, t_2 \geq 0$. Using Lemma 45, we thus obtain operators

$$(t_1\mathbb{I}_\mathcal{X} - C_1) \otimes (t_2\mathbb{I}_\mathcal{Y} + C_2) \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m) \tag{8.4}$$

and

$$(t_1\mathbb{I}_\mathcal{X} + C_1) \otimes (t_2\mathbb{I}_\mathcal{Y} - C_2) \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m) \tag{8.5}$$

where $t_1\mathbb{I}_\mathcal{X} - C_1 \in \mathrm{Sep}^*(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ by the constraints of (D_1), and similarly for $t_2\mathbb{I}_\mathcal{Y} - C_2$. Since $\mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m)$ is a convex cone, it follows that the average of (8.4) and (8.5) yields the desired operator

$$W := t_1 \cdot t_2\, \mathbb{I}_{\mathcal{X}\otimes\mathcal{Y}} - C_1 \otimes C_2 \in \mathrm{Sep}^*(\mathcal{X}_1 \otimes \mathcal{Y}_1, \ldots, \mathcal{X}_m \otimes \mathcal{Y}_m).$$

We conclude that $(t_1 \cdot t_2, W)$ is a feasible solution of the dual problem (D) with objective value $t_1 \cdot t_2$ as desired. This implies the perfect parallel repetition for SepQMA(m) for any choice of m.

We remark that the case of separable measurement operator corresponding to outcome *accept* is not the only case when perfect parallel repetition theorem holds. Harrow and Montanaro [HM10] showed that the maximum acceptance probability of a QMA(2) protocols is equal to the maximum output ∞-norm of an associated super-operator. To be more precise, let $C \in \mathrm{Pos}\,(\mathcal{Y} \otimes \mathcal{X})$ be the measurement operator corresponding to outcome *accept* of a QMA(2) protocol and let $\Phi : \mathrm{L}\,(\mathcal{X}) \to \mathrm{L}\,(\mathcal{Y})$ be a completely positive map such that $C = J(\Phi)$, the Choi-Jamiołkowski representation of the super-operator. For any completely positive map $\Phi : \mathrm{L}\,(\mathcal{X}) \to \mathrm{L}\,(\mathcal{Y})$, the *maximum output* ∞*-norm* is defined as

$$\|\Phi\|_{1\to\infty} = \max_{\rho \in \mathrm{D}(\mathcal{X})} \{\|\Phi(\rho)\|_\infty\}.$$

Using (2.2), we have that

$$\begin{aligned}
\|\Phi\|_{1\to\infty} &= \max_{\rho\in \mathrm{D}(\mathcal{X})} \left\{ \left\| \mathrm{Tr}_{\mathcal{X}} \left(C(\mathbb{I}_{\mathcal{Y}} \otimes \rho^{\top}) \right) \right\|_{\infty} \right\} \\
&= \max_{\rho\in \mathrm{D}(\mathcal{X})} \max_{\sigma\in \mathrm{D}(\mathcal{Y})} \left\{ \mathrm{Tr} \left[\sigma \, \mathrm{Tr}_{\mathcal{X}} \left(C \left(\mathbb{I}_{\mathcal{Y}} \otimes \rho^{\top} \right) \right) \right] \right\} \\
&= \max_{\rho\in \mathrm{D}(\mathcal{X})} \max_{\sigma\in \mathrm{D}(\mathcal{Y})} \left\{ \mathrm{Tr} \left[(\sigma \otimes \mathbb{I}_{\mathcal{X}}) \, C \, (\mathbb{I}_{\mathcal{Y}} \otimes \rho) \right] \right\} \\
&= \max_{\rho\in \mathrm{D}(\mathcal{X})} \max_{\sigma\in \mathrm{D}(\mathcal{Y})} \left\{ \langle C, \sigma \otimes \rho \rangle \right\}.
\end{aligned}$$

As a result, the question of perfect parallel repetition for QMA(2) protocols is equivalent to the multiplicative property of maximum output ∞-norm of the associated super-operator. This follows from the fact that for two super-operators Φ and Ψ, the Choi-Jamiołkowski representation of $\Phi \otimes \Psi$ is $J(\Phi) \otimes J(\Psi)$ up to reordering of operator spaces. Violations of the multiplicative property is known to hold for super-operators [GHP10, HW08, WH02] and that translates directly into QMA(2) protocols for which perfect parallel repetition does not hold. For instance, the Werner-Holevo channel [WH02] $\Phi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{X})$ defined as

$$\Phi(X) = \frac{1}{N-1} \left(\mathrm{Tr}(X) \, I - X^{\top} \right)$$

for $\mathcal{X} = \mathbb{C}^N$ forms an instance of QMA(2) protocol where perfect parallel repetition does not hold. Werner and Holevo showed that $\|\Phi\|_{1\to\infty}$ is not multiplicative for $N = 3$. For $N = 2$, the Choi-Jamiołkowski representation of Φ is a rank-one projection operator and the perfect parallel repetition does hold in this case. In fact, since the super-operator Φ is not *entanglement breaking* and hence C is not a separable operator [HSR03], it gives us a concrete example of perfect parallel repetition property that is not covered in this section.

Finally, we note that it is known that *entanglement breaking* maps do satisfy the multiplicative property of maximum output ∞-norm. For such maps, it holds that the Choi-Jamiołkowski representation is a separable operator [HSR03], a case which we have covered in this section.

Chapter 9

Conclusion and future work

In this thesis, we explored two questions on quantum interactive proof systems and their multi-prover variants. The first question we addressed pertains to the expressive powers of such models with or without resource constraints. The second question was related to error reduction technique of such proof systems via parallel repetition. We obtain few results that are summarized in the next section. We discuss few open problems related to the results presented in this thesis in the next section.

9.1 Conclusion

The question related to the expressive power of models of quantum interactive proof systems and their variants lead us to the following results. While the first result is obtained by an algorithmic technique described in Chapter 5, the next two results are obtained by *protocol simulations* as discussed in Chapter 8.

1. We saw that the expressive power of quantum interactive proof systems is exactly PSPACE, the class of problems that also admit a classical interactive proof systems. This result shows that in terms of complexity-theoretic characterization, both the models are equivalent. However, one should note that while three-message quantum interactive proof systems are sufficient to characterize PSPACE, classically the prover and the verifier require polynomial number of interactions for the same under complexity-theoretic assumptions. That the classical and quantum interactive proof systems have equivalent power should not be meant as the weakness of the quantum model for because they can simulate the classical model with constant number of interaction. The result should be viewed as extreme robustness of PSPACE. It is known that polynomial-space deterministic Turing machine can simulate their non-deterministic, probabilistic, and

quantum variants. This result merely tells us that the power of quantum interaction in single-prover interactive proof system models is also subsumed by PSPACE.

2. We also saw that polynomially many logarithmic-size unentangled quantum proofs are no more powerful than classical proof if the verifier has the ability to process quantum information. This result follows from an observation that logarithmic-size quantum states can be efficiently represented classically. Our result is in contrast with a result of Marriott and Watrous [MW05] who show that a single logarithmic-size quantum proof does not provide any advantage to the verifier. One can even think of them model of quantum interactive proof system with constant number of interactions, each of logarithmic-size message. This model is also known to characterize BQP [BSW11].
3. We also established that the model of multi-prover quantum Merlin Arthur proof system, where the verifier can only apply nonadaptive unentangled measurement on each proof and then apply a quantum circuit on the classical outcomes, under some restrictions is no more powerful than QMA. More specifically, under the restriction that there are only polynomial number of outcomes per proof, the multi-prover QMA protocol can be simulated by a (single-prover) QMA protocol. We view this result as a step towards answering the question on whether such restrictive verification procedure with multiple proofs can be simulated by a verification procedure with single proof.

The question related to error reduction via parallel repetition lead us to following results on a class of two-prover one-round quantum games and a class of multi-prover QMA proof systems. The first two results are obtained by using duality of semidefinite programming (discussed in Chapters 6 and 7) and th final result is obtained via cone programming duality (see Chapter 8).

1. We established that in the presence of quantum information, certain class of two-prover one-round games known as the XOR games, admit a *perfect* parallel repetition theorem in the following sense. When the provers play a collection of XOR games, an optimal strategy of the provers is to play each instance of the collection independently and optimally. In particular, the success probability of the quantum provers in the n-fold repetition of an XOR game G with quantum value $\omega_q(G)$ is exactly $(\omega_q(G))^n$. Our result together with the results of Cleve, Høyer, Toner and Watrous [CHTW04a] also established a better bound on the n-fold repetition of classical XOR games, which was proven tight by Raz [Raz11].
2. We also established a parallel repetition theorem for two-prover one-round unique games with quantum provers. Our result was obtained independently of Kempe, Regev and Toner [KRT10]. More specifically, we showed that if the quantum value of a unique game is $1 - \epsilon$, then the quantum value of n-fold repetition of the game is at most $(1 - \Omega(\epsilon^2))^n$. This bound was proven tight by Kempe and Regev [KR10]. We also

established that for certain class of unique games, the MOD_k games, the quantum value of the n-fold repetition of the game is at most $(1 - \Omega(\epsilon))^n$. Our proof technique also gave an alternate proof of perfect parallel repetition theorem for XOR games.

3. Our final result on parallel repetition is concerned with $\text{SepQMA}(m)$ proof systems, where the verifier receives m unentangled quantum proofs and the measurement operator corresponding to outcome *accept* is a fully separable operator. We give an alternate proof of a result of Harrow and Montanaro [HM10] that states that perfect parallel repetition theorem holds for such proof systems. Our proof goes via cone programming duality.

9.2 Future work

This section intends to summarize open problems and pointers to future research projects pertinent to the models, the results, and the techniques used in this thesis.

1. **Error reduction of two-prover one-round quantum games:** In the classical case, the question of parallel repetition of two-prover one-round games was an important question with implications to hardness of approximation. A series of results have established almost tight bounds on parallel repetition of such games [Hol07, Rao08, Raz98, Raz11]. As discussed in Chapters 6 and 7, parallel repetition theorem holds for certain class of two-prover one-round games with quantum provers. However, the proof technique seems to be inherently limited. One may not obtain such a parallel repetition theorem for any two-prover one-round game with quantum provers using semidefinite programming. A recent result of Kempe and Vidick [KV11] do provide an error reduction technique for such games but they do not show the type of parallel repetition theorems discussed in this thesis and their bound is significantly weaker than the bounds obtained in this thesis. An open question related to this thesis is to obtain parallel repetition theorems for such games.
2. **Classical and quantum XOR proof systems:** The parallelizability of any quantum interactive protocol into a three–message quantum interactive protocol is an evidence, which substantiates the belief that quantum information is beneficial in the context of interactive proof systems. However, interactive proof systems provide a counter-intuitive perspective too. As mentioned in Chapter 6, due to Håstad's seminal work [Hås01], we know that the class of problems accepted by classical XOR proof systems is NEXP for certain completeness and soundness probability. On the other hand, it is known that the quantum analogue of XOR proof systems follow the following relationship [JUW09, Weh06]:

$$\oplus\text{MIP}^*(2,1) \subseteq \text{PSPACE}.$$

This gives us an evidence that quantum entanglement can strictly weaken the expressive power of proof systems (of course, under the widely believed assumption that PSPACE $\neq$ NEXP). The question of possibly larger complexity-theoretic separation between the two models remains open.

3. **Classical and quantum values of binary games:** It is known that for binary games, a perfect quantum strategy implies a perfect classical strategy [CHTW04a]. This gives us a qualitative relationship between the two quantities. A quantitative relationship between these two quantities remains an open question. In a recent paper, Beigi [Bei10] showed that for such games, existence of a perfect classical strategy is guaranteed if the optimum value of a semidefinite programming relaxation for quantum strategies is 1 [KRT10]. This relaxation gives us hope to find a quantitative relationship between classical and quantum value of binary games similar to XOR games. Another natural question to ask in the wake of results presented in Chapter 7 is whether the second semidefinite programming relaxation can be used to obtain a parallel repetition theorem for binary games.

4. **Expressive power of multiple Merlins:** Quantum information allows us to study the models of proof systems for which no non-trivial classical analogues exist. One such model is discussed in this thesis, where multiple Merlins send quantum proofs that are guaranteed to be unentangled. For such proof systems, it is known that two provers are sufficient to simulate polynomial number of provers. However, no non-trivial complexity-theoretic lower or upper bound is known for such proof systems. An obvious question is to come up with a better complexity-theoretic bounds on such proof systems. In Chapter 8, we discussed a restrictive variant of QMA(poly) known as BellQMA(poly) (see the Ref. [GSU11]). Whether this class collapses to QMA or its expressive power is more than that of QMA is an open problem worth exploring.

5. **Multiplicative property of semidefinite programs:** Recently, there has been lot of interest in characterizing semidefinite programs that follow multiplicative property under a product operation defined on them [LM08, MS07]. The product operations are generally motivated from the applications and can be tensor product [LM08, MS07], bipartite tensor product [CSUU08, FL92, KRT10, Upa07], or graph product [Lov79]. This property has found applications in areas such as communication complexity, graph theory, and error reduction by parallel repetition. As such, a thorough study of the property is desirable.

Bibliography

[Aar05] S. Aaronson. Quantum computing, postselection, and probabilistic polynomial time. *Proceedings of the Royal Society A*, 461(2063):3473–3482, 2005. 9

[AB09] S. Arora and B. Barak. *Computational Complexity — A Modern Approach*. Cambridge University Press, 2009. 46

[ABDFS09] S. Aaronson, S. Beigi, A. Drucker, B. Fefferman, and P. Shor. The power of unentanglement. *Theory of Computing*, 5(1):1–42, 2009. A preliminary version appeared in *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*, pages 223–236, 2008. 156, 157

[ABS10] S. Arora, B. Barak, and D. Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 563-572, 2010. 122

[AK07] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 227–236, 2007. 10, 43, 75

[ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. 56, 121, 154

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. 56, 121, 154

[AS08] K Azimian and M. Szegedy. Parallel repetition of the odd cycle game. In *Proceedings of the 8th Latin American Symposium*, volume 4957 of *Lecture Notes in Computer Science*, pages 676–686. Springer, 2008. 123

[Bab85] L. Babai. Trading group theory with randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 421–429, 1985. 51, 53, 61

[BB84] C. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984. 8

[BBBMR06] E. Biham, M. Boyer, P. Boykin, T. More, and V. Rowchowdhury. A proof of the security of quantum key distribution. *Journal of Cryptology*, 19(4):381–439, 2006. A preliminary version appeared in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 715–724, 2000. 8

[BCLKP02] J. Barrett, D. Collins, L. Hardy, A. Kent, and S. Popescu. Quantum nonlocality, Bell inequalities and the memory loophole. *Physical Review A* 66:042111, 2002. 96, 104

[BCY11] F. Brandão, M. Christandl, and J. Yard. A quasipolynomial-time algorithm for the quantum separability problem. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 343–352, 2011. 157

[Bei08] S. Beigi. NP vs $\mathrm{QMA}_{\log}(2)$. Manuscript available as arXiv.org e-Print 0810.5109, 2008. 156

[Bei10] S. Beigi. A lower bound on the value of entangled binary games. *Quantum Information and Computation*, 10(11–12):911–924, 2010. 149, 179

[Bel64] J. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1(3):195–200, 1964. 99

[Ben82a] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29(3):515–546, 1982. 7

[Ben82b] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines that dissipate no energy. *Physical Review Letters*, 48:1581–1585, 1982. 7

[BFKT86] M. Ben-Or, E. Feig, D. Kozen, and P. Tiwari. A fast parallel algorithm for determining all roots of a polynomial with real roots. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 340–349, 1986. 91

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991. 53, 54, 56

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and non-approximability — towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998. 56, 98

[Bha97] R. Bhatia. *Matrix Analysis*. Springer New York, 1997. 13, 78

[BHHRRS08] B. Barak, M. Hardt, I. Haviv, A. Rao, O. Regev, and D. Steurer. Rounding parallel repetition of unique games. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 374–383, 2008. 123

[BOGKW88] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988. 53, 54, 56

[BGS11] S. Beigi, P. Shor, and J. Watrous. Quantum interactive proofs with short messages. *Theory of Computing*, 7:101–117, 2011. 159, 177

[BM88] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988. 53

[Bor77] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, 1977. 10, 64

[BP98] M. Bini and D. Pan. Computing matrix eigenvalues and polynomial zeros where the output is real. SIAM Journal on Computing, 27(4):1099–1115. 91

[Bra08] F. Brandão. *Entanglement Theory and the Quantum Simulation of Many-Body Physics*. Ph D. Thesis, Imperial College London, London, 2008. Also available as arXiv.org e-Print 1011.2751. 157, 159

[BRRRS09] B. Barak, A. Rao, R. Raz, R. Rosen, and R. Shaltiel. A strong parallel repetition theorem for free projection games. In *Proceedings of the 13th International Workshop on Randomization and Approximation Techniques*, volume 5687 of *Lecture Notes in Computer Science*, pages 352–365. Springer, 2009. 123, 124

[BSW11] S. Beigi, P. Shor, and J. Watrous. Quantum interactive proofs with short messages. *Theory of Computing*, 7(1):101–117, 2011. 159, 177

[BT09] H. Blier and A. Tapp. All languages in NP have very short quantum proofs. In *Proceedings of the 3rd International Conference on Quantum, Nano and Micro Technologies*, pages 34–37, 2009. 156

[Bug04] Y. Bugeaud. *Approximation by Algebraic Numbers*. Cambridge Tracts in Mathematics, volume 160, Cambridge University Press, Cambridge, 2004. 91

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. A preliminary version appeared in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 11–20, 1993. 7, 164

[BvN50] G. Brown and J. von Neumann. Solutions of games by differential equations. *Contributions to the Theory of Games*, 1:73–79, 1950. 43

[CCL90] J. Cai, A. Condon, and R. Lipton. On bounded round multi-prover interactive proof systems. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory*, pages 45–54, 1990. 56, 121, 123

[CCL92] J. Cai, A. Condon, and R. Lipton. On games of incomplete information. *Theoretical Computer Science*, 103(1):25–38, 1992. 56, 121, 123

[CD10] J. Chen and A. Drucker. Short multi-prover quantum proof systems for SAT without entangled measurements. Manuscript available as arXiv.org e-Print 1011.0716, 2010. 156, 157

[CGJ09] R. Cleve, D. Gavinsky, and R. Jain. Entanglement-resistant two-prover inteactive proof systems and non-adaptive PIRs. *Quantum Information and Computation*, 9(7-8):648–656, 2009. 98

[Cha00] B. Chazelle. *The Discrepancy Method — Randomness and Complexity*. Cambridge University Press, 2000. 43

[Cho75] M. Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975. 26

[CHSH69] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, 1969. 99

[CHTW04a] R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th Annual IEEE Conference on Computational Complexity*, pages 236–249, 2004. 11, 61, 98, 100, 115, 116, 141, 149, 151, 177, 179

[CHTW04b] R. Cleve, P. Høyer, B. Toner, and J. Watrous. Consequences and limits of nonlocal strategies. Presentation given at *Proceedings of the 19th IEEE Conference on Computational Complexity*, June 2004. 98

[Chu36] A. Church. An unsolvable problem of elementary number theory. American Journal of Mathematics, 58(2):345–363, 1936. 2

[CMM06a] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, 2006. 121

[CMM06b] E. Chlamtac, K. Makarychev, and Y. Makarychev. Hopw to play unique games using embeddings. In *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science*, pages 687–696, 2006. 121

[Coo71] S. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971. 3

[Csa76] L. Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5(4):618–623, 1976. 91

[CSUU08] R. Cleve, W. Slofstra, F. Unger, and S. Upadhyay. Perfect parallel repetition theorem for quantum XOR proof systems. *Computational Complexity*, 17(2):282–299, 2008. A preliminary version appeared in *Proceedings of 22nd IEEE Conference on Computational Complexity*, pages 109–114, 2007. Manuscript available as arXiv.org e-Print 0608146, 2006. 11, 96, 105, 117, 124, 179

[DdW11] A. Drucker and R. de Wolf. Quantum proofs for classical theorems. *Theory of Computing Library — Graduate Surveys*, 2:1–54, 2011. 9

[Deu85] D. Deutsch. Quantum theory, the Church–Turing principle, and the universal quantum Turing machine. *Proceedings of the Royal Society of London A*, 400:97–117, 1985. 7

[deW06] R. de Wolf. Lower bounds on matrix rigidity via a quantum argument. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 62–71, 2006. 9

[Din07] I.Dinur. The PCP Theorem by gap amplification. *Journal of the ACM*, 54(3):1–44 (Article 12), 2007. 154

[DLR79] D. Dobkin, R. Lipton, and S. Reiss. Linear programming is log-space hard for P. *Information Processing Letters*, 8(2):96–97, 1979. 94

[DR06] I. Dinur and O. Reingold. Assignment testers: Towards a combinatorial proof of the PCP Theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006. 152

[Edm65] J. Edmonds. Path, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. 3

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984. 46

[Fel86] P. Feldman. The optimal prover lies in PSPACE. Manuscript, 1986. 53

[Fey82] R. Feynman. Simulating physics by quantum computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982. 7

[Fey86] R. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, 1986. A preliminary version appeared in *Optic News*, 11:11–20, 1985. 7

[FG95] U. Feige and M. Goemans. Approximating the value of two prover proof systems, with applications to MAX2SAT and MAXDICUT. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995. 97, 104

[FK97] U. Feige and J. Kilian. Making games short. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 506–516, 1997. 53, 94, 95

[FK00] U. Feige and J. Kilian. Two-prover protocols—low error at affordable rates. *SIAM Journal on Computing*, 30(1):324–346, 2000. 152

[FKO07] U. Feige, G. Kindler, and R. O'Donnell. Understanding parallel repetition requires understanding foams. In *Proceedings of 22nd IEEE Conference on Computational Complexity*, pages 179–192, 2007. 97, 104, 116, 123

[FL92] U. Feige and L. Lovász. Two-prover one-round proof systems: their power and their problems. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 733–744, 1992. 53, 56, 97, 104, 111, 112, 116, 121, 123, 179

[For89] L. Fortnow. Complexity theoretic aspects of interactive proof systems. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1989. 56

[FRS94] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994. 54

[FS99] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1–2):79–103, 1999. 43

[FV02] U. Feige and O. Verbitsky. Error reduction by parallel repetition: A negative result. *Combinatorica*, 22(4):461–478, 2002. A preliminary version appeared in *Proceedings of 11th IEEE Conference on Computational Complexity*, pages 70–76, 1996. 56, 57, 123

[Gat93] J. v. z. Gathen. Parallel linear algebra. In *Synthesis of Parallel Algorithms*, J. Reif, Ed. Morgan Kaufmann Publishers, Inc., Chapter 13, pages 573–618, 1993. 90

[GB02] L. Gurvits and H. Barnum. Largest separable balls around the maximally mixed bipartite quantum state. *Physical Review A*, 66(6):062311, 2002. 172

[GB03] L. Gurvits and H. Barnum. Separable balls around the maximally mixed multipartite quantum states. *Physical Review A*, 68(4):042312, 2003. 172

[GB05] L. Gurvits and H. Barnum. Better bound on the exponent of the radius of the multipartite separable ball. *Physical Review A*, 72(3):032322, 2005. 172

[GHP10] A. Grudka, M. Horodecki, and L. Pankowski. Constructive counterexamples to the additivity of minimum output Rényi entropy of quantum channels for all $p > 2$. *Journal of Physics A: Mathematical and Theoretical*, 43(42):425304, 2010. 175

[GLS93] M Grötschel, L. Lovázs, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, second corrected edition, 1993. 42

[GM02] S. Goldwasser and D. Micciancio. Complexity of Lattice Problems — A Cryptographer Perspective. Springer International Series in Engineering and Computer Science, volume 671. Springer, 2002. 8

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *SIAM Journal of Computing*, 18(1):186–208, 1989. 51

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *Proceedings of the 17th Annual ACM Symposium on Theory of Computing* , pages 291–306, 1985. 53

[Gol05] O. Goldreich. On promise problems (a survey in memory of Shimon Even [1935 – 2004]). Electronic Colloquium on Computational Complexity, Report TR05–018, 2005. 46, 47

[Gol08] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008. 46

[GS89] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 73–90, 1989. A preliminary version appeared in *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 59–68, 1986. 53, 61

[GSU11] S. Gharibian, J. Sikora, and S. Upadhyay. QMA variants with polynomially many provers. Manuscript available as arXiv.org e-Print 1108.0617, 2011. 153, 179

[Gut09] G. Gutoski. *Quantum strategies and local operations.* PhD thesis, University of Waterloo, Waterloo, ON, Canada, 2009. 62

[GW95] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. A preliminary version appeared in *Proceedings of the 26th Annual ACM Symposium of Theory of Computing*, pages 422–431, 1994. 116

[GW05] G. Gutoski and J. Watrous. Quantum interactive proofs with competing provers. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer, 2005. 61

[GW07] G. Gutoski and J. Watrous. Toward a general theory of quantum games. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 565–574, 2007. 44, 61, 94, 95

[GW10] G. Gutoski and X. Wu. Parallel approximation of min-max problems with applications to classical and quantum zero-sum games. Manuscript available at arXiv.org e-Print 1011.2787v2, 2010. 59, 93, 94

[Hal07] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the Principal ideal problem. *Journal of the ACM*, 54(1):1–19, 2007. A preliminary version appeared in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 653–658, 2002. 8

[Hås96] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37th Annual IEEE Symposium on Foundation of Computer Science*, pages 627–636, 1996. 56

[Hås01] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 56, 98, 121, 178

[Hol07] T. Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 411–419, 2007. 56, 123, 150, 178

[HJ85] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985. 13

[HKSZ08] S. Hallgren, A. Kolla, P. Sen, and S. Zhang. Making classical honest verifier zero knowledge against quantum attacks. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 592–603. Springer, 2008. 61

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their Applications. *Bulletins of American Mathematical Society*, 43:439–561, 2006. 44

[HM10] A. Harrow and A. Montanaro. An efficient test for product states with applications to quantum Merlin-Arthur games. In *Proceedings of the 51st Annual IEEE Symposium on Foundation of Computer Science*, pages 633–642, 2010. iv, 12, 154, 155, 156, 158, 159, 160, 174, 178

[HSR03] M. Horodecki, P. Shor, and M. Ruskai. Entanglement breaking channels. *Reviews in Mathematical Physics*, 15(6):629–641, 2003. 175

[HW08] P. Hayden and A. Winter. Counterexamples to the maximal p-norm multiplicativity conjecture for all $p > 1$. *Communications in Mathematical Physis*, 284(1):263–280, 2008. 175

[Jam72] A. Jamiołkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. *Reports on Mathematical Physics*, 3(4):275–278, 1972. 26

[JJUW09] R. Jain, Z. Ji, S. Upadhyay and J. Watrous. QIP = PSPACE. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 573–582, 2010. Manuscript available as arXiv.org e-Print 0907.4737, 2009. 10, 58, 93

[JUW09] R. Jain, S. Upadhyay and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 573–582, 2009. Manuscript available as arXiv.org e-Print 0905.1300, 2009. 43, 62, 178

[JW09] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 243–253, 2009. Manuscript available as arXiv.org e-Print 0808.2775, 2008. 43

[JY11] R. Jain and P. Yao. A parallel approximation algorithm for positive semidefinite programming. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011. To appear. Manuscript available at arXiv.org e-Print 1104.2502, 2011. 59, 93, 94

[Kal07] S. Kale. *Efficient algorithms using the multiplicative weights update method*. PhD thesis, Princeton University, Princeton, NY, USA, 2007. 10, 43, 67

[Kar72] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, Eds. R Miller and J. Thatcher, 85–103, 1972. 3

[KdW03] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 106–115, 2003. 9

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002. 121

[KKMO07] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other two-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007. 121

[KKMTV11] J. Kempe, H. Kobayashi, K. Matsumoto, B. Toner, and T. Vidick. Entangled games are hard to approximate. *SIAM Journal on Computing*, 40(3):848–877, 2011. 61

[KKMV09] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. *Computational Complexity*, 18(2):273–307, 2009. 61

[KM03] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3):429–450, 2003. 61

[KMY03] H. Kobayashi, K. Matsumoto and T. Yamakami. Quantum Arthur-Merlin proof systems: Are multiple Merlins more helpful? In *Proceedings of the 14th International Symposium on Algorithms and Computation*, volume 3341 of *Lecture Notes in Computer Science*, pages 189–198. Springer, 2003. 156

[Kob08] H. Kobayashi. General properties of quantum zero-knowledge proofs. In *Proceedings of the 5th IACR Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2008. 61

[Kol10] A. Kolla. Spectral algorithms for unique games. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pages 122–130, 2010. 122

[KR08] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008. 121

[KR10] J. Kempe and O. Regev. No strong parallel repetition with entangled and non-signaling provers. In Proceedings of the 25th Annual IEEE Conference on Computational Complexity, pages 7–15, 2010. 124, 148, 150, 177

[KRT08] J. Kempe, O. Regev, and B. Toner. Unique games with entangled provers are easy. In *Proceedings of the 49th Annual IEEE Symposium on Foundation of Computer Science*, pages 457–466, 2008. 11, 104, 122, 124

[KRT10] J. Kempe, O. Regev, and B. Toner. Unique games with entangled provers are easy. *SIAM Journal on Computing*, 39(7):3207–3229, 2010. Manuscript available as arXiv.org e-Print 0710.06555, 2007. 115, 117, 119, 120, 125, 126, 127, 128, 129, 131, 148, 149, 150, 177, 179

[KSV02] A. Kitaev, A. Shen, and M. Vyali. *Classical and Quantum Computation*. Volume 47 of *Graduate School in Mathematics*, American Mathematical Society, 2002. 13, 154

[KV05] S. Khot and N. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proceedings of 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 53–62, 2005. 122, 151

[KV11] J. Kempe and T. Vidick. Parallel repetition of entangled games. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 353–362, 2011. Manuscript available as arXiv.org e-Print 1012.4726, 2010. 151, 178

[KW00] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 608–617, 2000. 61, 167, 170

[LB90] A. Lozano and J. Balcázar. The complexity of graph problems for succinctly represented graphs. In *Proceedings of the 15th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 411 of *Lecture Notes in Computer Science*, pages 277–286. Springer, 1990. 65

[Lev73] L. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973. In Russian. The English version appears in *Problems of Information Transmission*, 9(3): 265–266, 1973. 3

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. Journal of the ACM, 39(4):859–868, 1992. 10, 52, 58

[LM08] T. Lee and R. Mittal. Product theorems via semidefinite programming. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 674–685. Springer, 2008. 120, 136, 149, 151, 179

[LMSS07] N. Linial, S. Mendelson, G. Schechtman, and A. Shraibman. Complexity measures of sign matrices. *Combinatorica*, 27(4):439–463, 2007. 117

[LN93] M. Luby and N. Nisan. A parallel approximation algorithm for positive linear programming. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 448–457, 1993. 94

[Lov79] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979. 44, 179

[Lov00] L. Lovász. Integer sequences and semidefinite programming. *Publicationes Mathematicae Debrecen* 56:475–479, 2000. 44

[Lov03] L. Lovász. Semidefinite programming and combinatorial optimization. *Recent Advances in Algorithms and Combinatorics*, B Reed and C. Linhares-Sales Eds. CMS Books Math./Ouvrages Math. SMC, 11, Springer, New York, pages 137–194, 2003. 38

[LR05] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In *Handbook on Discrete Optimization*, K. Aardal, G. Nemhauser, R. Weismantel Eds. Elsevier B.V, Volume 12, 393–514, 2005. 38

[LS91] D. Lapidot and A. Shamir. Fully parallelized multi prover protocols for NEXP-time. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 13–18, 1991. 54

[LS09] N. Linial and A. Shraibman. Lower bounds in communication complexity based on factorization norms. *Random Structures and Algorithms*, 34:368–394, 2009. 117

[LSS08] T. Lee, A. Shraibman, and R. Špalek. A direct product theorem for discrepancy. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*, pages 71–80, 2008. 117

[Mah61] K. Mahler. *Lectures on Diophantine Approximations*. Cushing Malloy, 1961. 91

[Mah64] K. Mahler. An inequality for the discriminant of a polynomial. *Michigan Mathematical Journal*, 11(3):257–262, 1964. 91

[May01] D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001. 8

[Meg92] N. Meggido. A note on approximate linear programming. *Information Processing Letters*, 42(1):53, 1992. 94

[MS07] R. Mittal and M. Szegedy. Product rules in semidefinite programming. In *Proccedings of the 16th International Symposium on Fundamentals of Computation Theory*, volume 4679 of *Lecture Notes in Computer Science*, pages 435–445. Springer, 2007. 117, 179

[MW05] C. Marriott and J. Watrous. Quantum Arthur-Merlin games. *Computational Complexity*, 14(2):122-152, 2005. 10, 61, 63, 73, 159, 167, 170, 177

[NC00] M. Nielsen and I. Chaung. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 13

[Nef96] C. Neff. Specified precision polynomial root isolation is in NC. *Journal of Computer and System Sciences*, 48(3):429–463. 91

[PST95] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithm for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995. 43

[Rag08] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–254, 2008. 121

[Rao08] A. Rao. Parallel repetition in projection games and a concentration bound. In *proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2008. 11, 57, 116, 123, 178

[Raz98] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. 56, 104, 121, 123, 178

[Raz11] R. Raz. A counterexample to strong parallel repetition. *SIAM Journal on Computing*, 40(3):771–777, 2011. 11, 116, 124, 177, 178

[Rob51] J. Robinson. An iterative method of solving a game. *The Annals of Mathematics*, 54(2):296–301, 1951. 43

[Rom08] S. Roman. *Advanced Linear Algebra. Graduate Texts in Mathematics*, Springer, 2008. 13

[RR10] R. Rosen and R. Raz. A strong parallel repetition theorem for projection games on expanders. Electronic Colloquium on Computational Complexity, Report TR10-142, 2010. 124

[Ser91] M. Serna. Approximating linear programming is log-space complete for P. *Information Processing Letters*, 37(4):233–236, 1991. 94

[Sha92] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992. 10, 52, 58

[She92] A. Shen. IP = PSPACE: simplified proof. *Journal of the ACM*, 39(4):878–880, 1992. 10, 52, 58

[Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. A preliminary version appeared in *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994. 8

[Sim97] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1473–1483, 1997. A preliminary version appeared in *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 116–123, 1994. 8

[Slo10] W. Slofstra. Lower bounds on the entanglement needed to play XOR nonlocal games. Manuscript available at arXiv.org e-Print 1007.2248, 2010. 118

[SP00] P. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85(2):441–444, 2000. 8

[SV03] A. Sahai and S. Vadhan. A complete problem for statistical zero-knowledge. *Journal of the ACM*, 50(2):1–54, 2003. 99

[Tre08] L. Trevisan. Approximation algorithms for unique games.. *Theory of Computing*, 4(1):111–128, 2008. 121

[Tsi80] B. S. (Tsirelson) Cirel'son. Quantum generalizations of Bell's inequality. *Letters in Mathematical Physics*, 4(2):93–100, 1980. 11, 99

[Tsi85] B. S. (Tsirelson) Tsirel'son. Quantum analogues of the Bell inequalities: The case of two spatially separated domains. *Journal of Soviet Mathematics*, 36:557–570, 1987. 11, 100

[Tun10] L. Tunçel. *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*. Fields Institute Monographs, Volume 27, American Mathematical Society, 2010. 38

[Tur37] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society (second series)*, 42(1):230–265, 1937. 2

[TW08] L. Tunçel and H. Wolkowicz. Strong duality and minimal representations for cone optimization. Technical Report CORR 2008-07, Department of Combinatorics and Optimization, University of Waterloo, August 2008. 163

[Upa07] S. Upadhyay. A parallel repetition theorem for unique games with quantum provers. Manuscript, 2007. 11, 104, 119, 124, 136, 137, 149, 179

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996. 38

[VB04] L. Vandenberghe and S. Boyd. *Convex Optimization*. Cambridge University Press, 2004. 38

[Ver95] O. Verbitsky. Parallel repetition conjecture for trees is true. Electronic Colloquium on Computational Complexity, Report TR95-013, 1995. 56, 123

[Ver96] O. Verbitsky. Towards the parallel repetition conjecture. *Theoretical Computer Science*, 157(2):277–282, 1996. 56, 123

[Wat02] J. Watrous. Limits on the power of quantum statistical zero-knowledge. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 459-468, 2002. 61, 160

[Wat03] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575-588, 2003. A preliminary version appeared in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 112–119, 1999. 59, 61

[Wat04] J. Watrous. Personal communication, 2004. 97, 104, 111, 123

[Wat08] J. Watrous. *Lecture Notes in Theory of Quantum Information* Manuscript available at http://www.cs.uwaterloo.ca/ watrous/quant-info/, 2008. 13, 28, 41, 42

[Wat09a] J. Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. 61

[Wat09b] J. Watrous. Quantum computational complexity. *Encyclopedia of Computer and System Sciences*. Springer, 2009. 48, 155

[Weh06] S. Wehner. Entanglement in interactive proof systems with binary answers. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2006. 98, 178

[WH02] R. Werner and A. Holevo. Counterexample to an additivity conjecture for output purity of quantum channels. *Journal of Mathematical Physics*, 43(9):4353–4357, 2002. 175

[WK06] M. Warmuth and D. Kuzmin. Online variance minimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 514–528. Springer, 2006. 10, 43

[Wu10] X. Wu. Equilibrium value method for the proof of QIP = PSPACE. Manuscript available at arXiv.org e-Print 1004.0264, 2010. 94

[Yao93] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 352–361, 1993. 7

[You01] N. Young. Sequential and parallel algorithm for mixed packing and covering. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 538–546, 2001. 94