From non-abelian anyons to quantum computation to coin-flipping by telephone

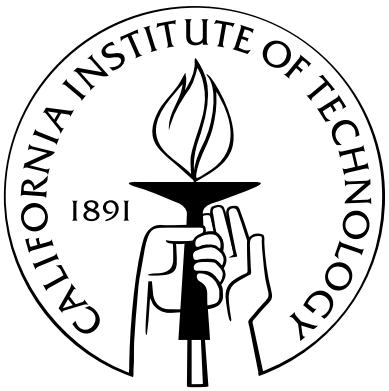
Thesis by

Carlos Mochon

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2005

(Defended May 9, 2005)

to Alice and Bob

(whose bitter divorce was a constant motivation)

Acknowledgements

I would like to thank my advisor John Preskill for his guidance and advice, and for pointing me in the direction of the two research topics that comprise this thesis, both of which ended up being exceedingly fruitful. I would also like to thank Alexei Kitaev for numerous discussions and for the opportunity to build on his results.

Over the years at Caltech I've had helpful discussions with many people including Patrick Hayden, Dave Bacon, Michael Ben-Or, Rob Spekkens, Debbie Leung and Andrew Childs, plus many others whom I've unfortunately forgotten to include here.

Extra special thanks are due to Jim Harrington, James Chakan, Meg Wessling, Charlene Ahn, Andrew Landahl, Ben Toner and Graeme Smith, who've proofread my papers, discussed my ideas and generally tolerated my incessant complaining.

Finally, I'd like to thank my candidacy and thesis defense committees: John Preskill, Alexei Kitaev, Jeff Kimble, Hideo Mabuchi, Anton Kapustin and Chris Umans for helping me move from frigid California to sunny Canada.

Abstract

Following their divorce, Alice and Bob would like to split some of their possessions by flipping a coin. Unwilling to meet in person, and without a trusted third party, they must figure out a scheme to flip the coin over a telephone that guarantees that neither party can cheat.

The preceding scenario is the traditional definition of two-party coin-flipping. In a classical setting, without limits on the available computational power, one player can always guarantee a coin-flipping victory by cheating. However, by employing quantum communication it is possible to guarantee, with only information-theoretic assumptions, that neither party can win by cheating, with a probability greater than two thirds. Along with the description of such a protocol, this thesis derives a tight lower bound on the bias for a large family of quantum weak coin-flipping protocols, proving such a protocol optimal within the family. The protocol described herein is an improvement and generalization of one examined by Spekkens and Rudolph. The key steps of the analysis involve Kitaev's description of quantum coin-flipping as a semidefinite program whose dual problem provides a certificate that upper bounds the amount of cheating for each party.

In order for such quantum protocols to be viable, though, a number of practical obstacles involving the communication and processing of quantum information must be resolved. In the second half of this thesis, a scheme for processing quantum information is presented, which uses non-abelian anyons that are the magnetic and electric excitations of a discrete-group quantum gauge theory. In particular, the connections between group structure and computational power are examined, generalizing previous work by Kitaev, Ogburn and Preskill. Anyon based computation has the advantage of being topological, which exponentially suppresses the rate of decoherence and the errors associated with the elementary quantum gates. Though no physical systems with such excitations are currently known to exist, it remains an exciting open possibility that such particles could be either engineered or discovered in exotic two-dimensional systems.

Contents

Chapter 1

Introduction

To be or not to be, is there no other choice?

Shakespeare's [Sha] famous words reflect a perception of the world in which an object can exist in only one of a set of mutually exclusive states. Such a view was held by most of humanity until the early twentieth century. The advent of quantum mechanics, however, revealed a world where these states were only part of a continuum composed of weighted superpositions of all classical states.

Unfortunately, the effects of decoherence on large systems, which typically projects them into one of their classical states, means that Hamlet's decision would be no easier today than in Shakespeare's time. The small impact that quantum mechanics has had on the world of theater, though, was offset by the large thud that eventually resonated through the field of Computer Science. The discovery of quantum error correction [Sho95, Ste96] and fault-tolerant quantum computation [Sho96, Kit97b] opened the possibility of building large scale quantum systems that can be controlled in the laboratory and used to process and communicate quantum information.

Computers capable of processing quantum information do not appear to be polynomially equivalent to the standard Turing machine, as they can factor large numbers and compute discrete logarithms efficiently [Sho94], a task that is not believed possible with a standard classical computer. Furthermore, quantum communication has been proven to be strictly superior to classical communication for a number of tasks, including key distribution [Wie83, BB84].

There are therefore two fundamental questions in this nascent field of quantum computation:

1. For what tasks does the ability to process quantum information offer an advantage over the purely classical setting?

2. How can we build a system capable of harnessing this quantum power without being affected by processing errors and decoherence?

Partial answers to both of these questions will be discussed as we examine a scheme for building quantum computers out of non-abelian anyons, and a two-party communication problem in which quantum information provides a considerable advantage over classical information.

Of course, these are just two of the myriad of answers that have appeared in the literature. Examples of other algorithms, protocols and implementations of quantum computing can be found, for instance, in [NC00]. Said reference also contains the longer version of the much abridged history of quantum computing presented above.

1.1 Anyons: a potential implementation of quantum computation

The race towards the construction of the first large-scale quantum computer began in earnest in the mid 1990s with proposals involving optical qubits [CY95], cavity QED [THL+95], ion traps [CZ95] and NMR [CFH97, GC97, SV99]. Many other schemes have followed. Most, however, face serious technical challenges involving noise reduction and gate accuracy, which must be addressed before a full fledged quantum computer can be built.

The difficulty arises in the seemingly conflicting set of requirements that a quantum computer must satisfy: On the one hand, a good quantum computer must store its information in qubits that are sufficiently isolated from each other and the environment that the probability of an error occurring is small. On the other hand the qubits must interact sufficiently strongly with each other, in an externally controllable fashion, such that fast quantum gates can be realized.

One innovative approach to quantum computation was proposed by Kitaev [Kit97a] who suggested storing quantum information in topologically protected subspaces. Because the environment typically acts locally, and topological degrees of freedom can only be accessed (i.e., modified or measured) using highly non-local operations, such subspaces are highly resistant to errors.

Of course, as discussed above, it is not sufficient to just isolate the qubits from the environment; there must also be a way of addressing them. Fortunately, there are topological operations such as braiding and fusion that can access the encoded information. Furthermore, the resulting operations on the computational subspace depend only on their topological parameters and not on the details of their implementation such as the exact paths involved. This allows the production of highly accurate quantum gates, the second holy grail of quantum computation (the first one being long-lived qubits).

A good candidate for the implementation of the ideas of topological quantum computation is a system with anyons, which are particles whose statistics are neither bosonic nor fermionic. In quantum mechanics, the particle statistics are given by the action of the exchange operators, which have the effect of exchanging neighboring particles. The square of an exchange operator is equivalent to transporting one particle around another, which in three spatial dimension is homotopically equivalent to doing nothing, hence the braiding operator in three dimensions has only ± 1 eigenvalues corresponding to bosons and fermions. However, the situation is more complicated in two spatial dimensions since clockwise and counterclockwise exchanges can be distinguished. In this case, the

exchange operator, when acting on indistinguishable particles, can belong to any representation of the braid group. This representation can be non-abelian, giving rise to non-abelian anyons.

Apart from unitary gates (some of which can be build out of the above exchange or braiding operators), a quantum computer also requires a way of reading out the information. In the setting of anyons this can be obtained from the fusion operation, which takes two anyons and combines them into a single anyon that carries their total charge, or into the vacuum in the case when their total charge is trivial. Fusion, like braiding, is a topological operation whose outcome depends only on the charges of the particles involved and not on the details of the procedure.

Each set of rules for braiding and fusion describes a type or model of anyons. Of course, the braiding and fusion rules must satisfy some consistency requirements, which are known as the quasi-triangularity conditions. In fact, there is a beautiful mathematical theory underlying the set of anyons, which are really representations of quasi-triangular Hopf algebras (or even more generally quasi-Hopf algebras). However, we shall not use this highly abstract mathematical language.

In this thesis we will focus on a particular set of physically-inspired models for anyons that correspond to the spectrum of electric and magnetic charges of a quantum gauge theory with discrete symmetry group. Mathematically, these correspond to representations of Drinfeld's quantum double of the symmetry group. Physically, such models arise when a regular continuous-group quantum gauge theory has its symmetry broken, via the Higgs mechanism, to a discrete group G. In such a field theory all the gauge particles are massive, and hence do not mediate long-range interactions. A set of electric and magnetic charged particles remain unscreened, however, and such charges can be detected via Aharonov-Bohm interactions. A complete description of the spectrum can be found in [dWPB95] (and the original work [BvDdWP92]). These particles will form the elementary building blocks out of which a quantum computer can be built.

Returning to the question of implementations of quantum computations, one may wonder how practical is an anyon based computer. Abelian anyons have already been observed in the fractional quantum Hall effect, and non-abelian anyons are conjectured to exist at certain levels as well [NW96, RR99]. Unfortunately, such anyons do not belong to the anyon model discussed above, but rather to the one analyzed in [Fre00, FKLW01]. Though no system is currently known that supports the model of discrete-group gauge theory anyons discussed herein, it is possible that such a system could be engineered. Recent proposals include optical lattices [DDL02] and Josephson-junction arrays [DIV03]. In the latter case, an explicit array is constructed that simulates on a lattice the gauge theory with group S_3, which is the smallest group for which our construction works.

The contribution of this thesis to the subject of topological quantum computation is primarily theoretical, however. We shall not be concerned with the details of how one may obtain and control a system with non-abelian anyons. Rather, we shall assume the existence of a system with anyons, described by a quantum gauge theory of discrete symmetry group G, and then proceed to ask for

what such groups can a quantum computer be built out of the braiding and fusion of anyons. The analysis will lead to some interesting connections between group structure and computational power. In particular, we shall show that if G is non-nilpotent (a group property that shall be defined in Chap. 3) then the related anyons can be used to build a universal quantum computer, and we shall describe an explicit realization of one.

Chap. 2 deals with the simpler case of non-solvable groups and was originally published as [Moc03]. It contains a generalization of the construction by Ogburn and Preskill [OP99, Pre97] for the group A_5. Chap. 3 generalizes the previous results to include all non-nilpotent groups and is published as [Moc04a]. This second result, though subsuming the previous chapter, involves a more difficult proof. The quantum computer devised in this chapter is also experimentally more challenging, as it involves using both electric and magnetic charges, whereas the construction of Chap. 2 requires control of magnetic charges exclusively. The work of Chap. 3 is built on unpublished research by Kitaev [Kit02] who described a quantum computer using anyons from the group S_3.

The two chapters also contain some novel gate-sets, a discussion of leakage correction and a technique for building a reservoir of magnetic fluxes.

1.2 Coin-flipping: a potential application of quantum information

The flip side of the difficulty of building quantum computers is their potential ability to accomplish tasks that are either impossible or impractical with a classical computer alone.

A quantum computer can efficiently factor numbers and compute discrete logarithms [Sho94], solve Pell's equation [Hal02] and find the zeros of the Zeta function for certain varieties [vD04], tasks that are believed to be hard for classical computers. In the more abstract oracle query model, we can also find separations between quantum and classical queries, for instance in the case of unstructured search [Gro96].

There are further surprises in the realm of quantum communication and cryptography. The well-known case of key distribution [BB84] is useful when a set of honest users are trying to protect their communication from potentially malicious third parties.

A different set of cryptographic problems arises, though, when two or more people try to accomplish a joint task while protecting themselves from potential cheating from the other players. Typical tasks involve agreeing on a random outcome or calculating a joint function without revealing some private information. For example, Alice and Bob may wish to find a common free day when they can meet, without revealing to each other their schedules. Most of these problems fall under the banner of secure computation.

Classically, given certain cryptographic assumptions (such as certain limits on computational

power) all two-player [Yao82] and multiplayer [GMW87] functions can be computed. In the multiplayer case it is even possible to do this with information-theoretic security (i.e., without any limits on computational power) so long as each pair of players is connected by a secure private communication channel (and possibly an authenticated broadcast channel) [CCD88, BOGW88, RBO89]. The results can even be generalized to allow multiparty quantum computations to take place under similar assumptions [CGS02].

In this thesis we shall focus on problems involving two players. One the one hand, this is a much simpler scenario as one does not need to worry about issues of authentication, secrecy and jamming. If an honest player detects a protocol deviation, he knows exactly who is cheating. On the other hand, the security guarantees of most of the multiplayer results are only valid as long as a majority (or sometimes a super-majority) of players are honest. This assumption obviously cannot be imposed on two-player games, where we would like to protect an honest player from potential cheating from the other player.

Unfortunately, there is very little that can be done classically with information-theoretic security for two-party problems. All the known results invoke cryptographic assumptions (some of which, such as the difficulty of factoring, will no longer be true once quantum computers become available). It was initially hoped that quantum information could provide protocols with information-theoretic security for all secure computation problems. This lofty goal turned out to be unattainable, but certain partial results have been found.

Most of the work on quantum protocols has focused on two primitives called bit-commitment and coin-flipping, to be defined below. The interest in these problems is that they are simple enough to be comprehensibly studied and can often be used as building blocks for solving more complicated problems. Any classical secure computation can be built up from a few primitive protocols such as oblivious transfer [Kil88]. Early indications [Yao95] were that quantum bit-commitment could be used to produce protocols for oblivious transfer, and hence the rest of the secure computation problems, and many protocols for quantum bit-commitment were proposed.

Bit commitment can be defined as a two-party protocol comprising the following two stages:

- Commit stage: Alice commits herself to a bit $x \in \{0, 1\}$.
- Reveal stage: Alice reveals the bit x to Bob.

Each stage may involve many rounds of communication between Alice and Bob, and an arbitrary amount of time may pass between the two stages. The protocol must satisfy two requirements: Before the reveal stage Bob should not be able to obtain any information about the bit that Alice committed to. However, once the commit state is done, Alice should not be able to change the value of the committed bit.

In an information-theoretic setting, it should be clear that classical bit-commitment cannot be

accomplished: either Bob has enough information at the end of the commit stage to know the value of the committed bit, or Alice can change her commitment.

By using quantum mechanics, it is possible to formulate a partial solution to the problem of bit-commitment, where Bob has a non-unity probability of learning the committed bit, and Alice has a non-unity probability of changing the committed bit. It should be noted that quantum mechanically, Alice can always commit to a superposition of bit values rather than to a single value, but this is not significantly different from the classical case where she may assign to someone else the task of picking the committed bit without telling her its value. In these cases, the bit-commitment restriction would say that having completed the commit phase, the maximum probability with which Alice can reveal a zero plus the maximum probability with which she can reveal a one (possibly using a different reveal strategy) should sum to one.

Unfortunately, it is known [May96, LC98] that ideal bit-commitment cannot be realized in a quantum setting with information-theoretic security. Soon followed proofs that many two-party functions cannot be securely computed either [Lo97], including all those where only one party learns the outcome (which includes oblivious transfer as a special case). The conclusion is that while quantum information is superior to classical information for producing bit-commitment protocols, it is not sufficient to fully accomplish the task.

Future work on quantum two-party protocols focused on a simpler task: coin-flipping, sometimes called coin-flipping by telephone, which was proposed initially by Manuel Blum [Blu81]. The protocol allows two mutually distrustful parties to agree on the outcome of a coin-flip, such that neither of them can affect the outcome by cheating. The task is easy if there exists a third party, trusted by the first two, who can take care of flipping the coin and announcing the results. However, in many cases no trusted third party is available, and that is where cryptographic protocols become important.

In this thesis we shall be mainly concerned with a variant called weak coin-flipping in which each player has a preferred outcome. It is defined as a two-party communication protocol where Alice and Bob begin uncorrelated (and unentangled in the quantum case) and end by outputting one bit each. We say that Alice wins on outcome zero and Bob wins on outcome one. The protocol must guarantee the following:

1. If both players are honest, then Alice's bit must be uniformly random and must equal Bob's bit.

2. If Alice is honest, then even if Bob cheats, she will not output one (i.e., Bob wins) with a probability greater than P_B^*.

3. Similarly, if Bob is honest, then even if Alice cheats, he will not output zero (i.e., Alice wins) with a probability greater than P_A^*.

The parameters P_A^* and P_B^*, which ideally should be as small as possible, describe the protocol. It is often convenient to summarize these by specifying just one parameter: the bias $\epsilon = \max(P_A^*, P_B^*) - 1/2$. Ideal coin-flipping (i.e., when no cheating is allowed) corresponds to zero bias.

The above rules of coin-flipping are set up to correspond to the colloquial notion of flipping a coin. The output bit of Alice and Bob when both are honest is an agreement of the coin outcome. The requirement of starting off uncorrelated is imposed because this is effectively a protocol for establishing correlations. The ability to begin with correlations makes the problem uninteresting.

Note that when one player is dishonest, we place no restriction on their output, as this would be infeasible to enforce. We also do not worry about the case of two cheating players. The protocol is only intended to protect honest players.

The requirements of coin-flipping can also be strengthened to not allow either player to cheat in either direction. This corresponds to the case when the honest players do not have a priori a preferred outcome. Such a task is known as strong coin-flipping, and its protocol must satisfy all the requirements for weak coin-flipping, in addition to:

2'. If Alice is honest, then even if Bob cheats, she will not output zero with a probability greater than $P_B^{*\prime}$.

3'. If Bob is honest, then even if Alice cheats, he will not output one with a probability greater than $P_A^{*\prime}$.

For strong coin-flipping the bias is defined as $\epsilon = \max(P_A^*, P_A^{*\prime}, P_B^*, P_B^{*\prime}) - 1/2$.

The original definition for coin-flipping by telephone [Blu81] imposes a few extra requirements, such as the ability to convince a third party, once the protocol is completed, of what the coin outcome was and whether each participant was honest. Unfortunately, such requirements cannot be demanded of a quantum protocol as the no-cloning theorem implies that a transcript of the messages cannot be kept. Therefore, we shall not consider such extensions in this thesis.

The definition of strong coin-flipping is sometimes expanded to include an "abort" outcome to be used when a player detects a protocol deviation. In weak coin-flipping such an outcome is never needed because an honest player who catches her opponent cheating can always output her desired outcome. For strong coin-flipping the situation is more complicated, though. For example, the strategy of simply outputting a random bit when a player catches their opponent cheating is faulty, as the cheating opponent can play normally until the last round and then only cheat if he is about to lose, thereby guaranteeing himself a winning probability of 75%. This issue will not be important, though, as we shall mainly be concerned with weak coin-flipping in this thesis.

Strong coin-flipping can be built out of bit-commitment as follows: Alice first commits a random bit to Bob, then Bob sends a second random bit to Alice, and finally Alice reveals her commitment. The coin outcome is the XOR of the two random bits. On the other hand, bit-commitment cannot be

accomplished even given a strong coin-flipping black-box [Ken99]. Therefore, coin-flipping is weaker than bit-commitment, and possibly more amenable to implementation using quantum information.

Classically, even weak coin-flipping cannot be implemented with information-theoretic security with a bias less than one half. That is, one player is always allowed to maximally cheat, and the protocol is no more secure than if that player were simply assigned the task of deciding the coin outcome. We shall sketch an argument here.

The first step is to note that every classical coin-flipping protocol, under the assumption that each side has unlimited computational power, can be transformed into a public-coin coin-flipping protocol (to be defined below). We assume without loss of generality that each message consists of a single bit. The basic idea is to inductively prove that the protocol can be run so that neither Alice nor Bob keep any private information. It is clearly true at the start. Assume it is true immediately before sending message n. The sender (say Alice) will send a message that depends on all previous messages, and some private randomness that she will obtain in this round. Part of that randomness is immediately revealed to Bob when Alice sends her message. The rest of it is not used, and so we can postpone obtaining it until some future round, which leaves Alice with no private state.

Given that neither Alice nor Bob keep any private state, the protocol can be cast in the form of a public-coin coin-flipping protocol: Alice and Bob alternate announcing random bits (with each bit having a bias possibly depending on the value of all previous bits). After n messages they should each have the same n bits, which they use as input to a public n-bit to 1-bit function to obtain the outcome of the coin-flipping protocol.

In a public-coin protocol the scenario of two dishonest players is well defined: each player may choose their message bits to maximize the likelihood of winning. But this scenario is simply a standard two-player game with no possibility of tying, and therefore one player (say Bob) must always be able to win. To complete the proof that all classical coin-flipping protocols have bias of one half, we simply note that if Bob can guarantee a victory against Alice when she is cheating, then he can surely guarantee a victory against Alice when she is honest.

By exploiting quantum information, better (though not perfect) protocols can be built. For strong coin-flipping, Ambainis [Amb01] and Spekkens and Rudolph [SR02a] independently proposed protocols that achieve a bias of $1/4$. Unfortunately, Kitaev [Kit03] (see also [ABDR04]) establishes a lower bound on the bias of $1/\sqrt{2} - 1/2$, which holds even if the players are allowed to output "abort." The proof bears a similarity to some of the techniques used classically to establish lower bounds on the bias of coin-flipping problems involving multiple players [BOL85]. In fact, multiplayer strong coin-flipping is also bounded in the quantum case [ABDR04].

The problem of whether weak coin-flipping can be achieved with arbitrarily small bias remains open. The best known protocol prior to the work in this thesis is by Spekkens and Rudolph [SR02b], which achieves a bias of $1/\sqrt{2} - 1/2 \simeq 0.207$. Other previous protocols include [GVW99, KN04]. The

best known lower bound is by Ambainis [Amb01] and states that the number of messages must grow at least as $\Omega(\log\log\frac{1}{\epsilon})$. In particular, it implies that no protocol with a fixed number of messages can achieve arbitrarily small bias.

In this thesis we continue the search for protocols for weak quantum coin-flipping. Whereas most previous protocols typically involve 3–5 messages, we focus on sets of protocols that can have an arbitrary number of messages, which are particularly important in light of Ambainis lower bound. The technique that allows us to study protocols with a large number of messages is Kitaev's description of coin-flipping as a semidefinite program [Kit03].

In particular, in Chap. 5 we identify a generalization of Spekkens and Rudolph's protocol that achieves a bias of 0.192 in the limit of arbitrarily large number of messages (though ten messages are sufficient for a bias of 0.193). This work was originally published as [Moc04b]. Though the derivation of the lower bound on the bias in this chapter is analytic, the protocols have a set of parameters that were originally optimized numerically. Later, an analytic derivation (published in [Moc05]) was found and is included in the last section of the chapter.

The rest of [Moc05] is contained in Chap. 6, which shows that the protocol with bias 0.192 is part of a large family of quantum protocols for weak coin-flipping that are based on classical public-coin coin-flipping protocols. The quantum protocols are obtained by replacing classical randomness with quantum entanglement and by adding a cheat detection test in the last round that verifies the integrity of this entanglement. This larger family contains a second generalization of the Spekkens and Rudolph protocol that achieves a bias of 1/6.

In the same chapter, the complete family is analyzed, and a set of optimal protocols is identified for every number of messages. These optimality results prove that a bias of 1/6 is the best that can be achieved within the family. A previous optimality result by Ambainis [Amb02] proved that $1/\sqrt{2}-1/2 \simeq 0.207$ is the best bias that can be achieved within a particular family of three-message protocols. However, we believe that the result in this thesis is the first lower bound for a family containing protocols with an arbitrary number of messages.

The problem of the optimal bias for quantum weak coin-flipping remains open, but we speculate that it may be possible to show that every quantum protocol can be reduced to one of the protocols discussed in Chap. 6. If such a conjecture were true, then the optimal bias would be 1/6.

Even if arbitrary small bias cannot be achieved using quantum information, there is the possibility of constructing protocols for either coin-flipping or bit-commitment with good cheat detection. The idea of a "cheat" outcome is essentially the same as the "abort" outcome discussed above, but the assumption now is that the protocol will be used as part of some larger transaction where getting caught cheating may have significantly worse consequences than losing a coin-flip.

Some aspects regarding the possibility of coin-flipping with cheat detection will be examined in Chap. 4, where the idea of cheat detection will also be defined more precisely. Protocols for quantum

weak coin-flipping with cheat sensitivity have been constructed by Spekkens and Rudolph [SR02b] and prevent cheating by a large amount without the risk of getting caught, though small amounts of cheating may remain undetected. Aharonov et al. [ATSVY00] have a protocol with quadratic cheat detection for either party, though the cheat detection can only be used on one party at a time. The best known cheat detecting protocol is by Hardy and Kent [HK04] where neither party may cheat without risking some finite probability of getting caught, though the precise functional form of cheat detection was not determined. We note that the last two are actually protocols for bit-commitment with cheat detection, though they can be transformed into protocols for coin-flipping using the reduction presented above.

The work presented in Chap. 4 is published as [Moc04c], and considers cheat detection curves similar to the ones discussed in [RS04]. It begins by showing that quantum coin-flipping can be composed in series to produce new coin-flipping protocols. Though it is known that serial composition of standard coin-flipping protocols cannot be used to decrease the final bias [SV86], serial composition of coin-flipping with cheat detection is a different story. However, we shall also argue that serial composition of coin-flipping is probably not a useful tool in designing weak coin-flipping protocols with arbitrarily small bias (yet another piece of evidence that this may not be possible). We also use serial composition to derive an upper bound on the amount of cheat detection that can exist in coin-flipping and bit-commitment protocols in the quantum information-theoretic setting.

In summary, this thesis presents applications of quantum information that achieve significantly better results that what can be done solely with classical information. Though the practical applications of the protocols in this thesis remain unclear, at a minimum they should serve as further examples of the differences between the classical world that we appear to live in, and the quantum world where we actually reside.

Chapter 2

Computing with anyons from non-solvable finite groups

In this chapter we present a constructive proof that anyons from a quantum gauge theory with a finite non-solvable group are capable of performing universal quantum computations. The proof is a generalization of the work done by Ogburn and Preskill in [OP99, Pre97] for anyons in the group A_5 (the smallest finite non-solvable group). These anyon computers are examples of topological quantum computers as envisioned by Kitaev [Kit97a, FKLW01].

The chapter is organized as follows: We begin by introducing some notation and reviewing the properties of the anyon model that will be used throughout this thesis. Sec. 2.2 presents the universal gate-set that will be employed to prove anyons can perform quantum computations. Sections 2.3 and 2.5 contain the meat of the chapter and discuss a concrete anyonic implementation of all the necessary gates. For pedagogical reasons, we first cover the easier subcase of simple perfect groups in Sec. 2.3, and then discuss the required generalizations for any non-solvable group in Sec. 2.5. In Sec. 2.4, we discuss how to make these computations fault-tolerant by performing leakage correction, and in Sec. 2.6 we show how to create the required anyon ancillas. Finally, we include the required mathematical proofs in Sec. 2.7.

2.1 Review

We begin by reviewing some of the braiding and fusion properties of the anyons that will be used in this chapter. Our review will be rather abridged, but more details can be found in the excellent review of discrete gauge theories [dWPB95] (and the original work [BvDdWP92]). The paper by Ogburn and Preskill [OP99, Pre97] also contains a good review with emphasis on the applications to quantum computing.

This section also establishes our notation for qudits and reviews the phase estimation circuit, a highly useful trick that will be used often.

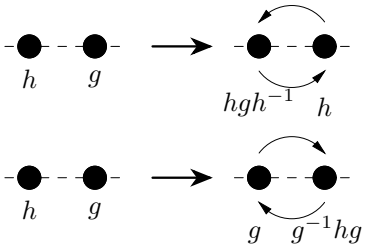


Figure 2.1: Exchanging two anyons.

2.1.1 Magnetic charges

The main players throughout this chapter will be the magnetic charges, also known as fluxes. For a field theory with an unbroken finite group G, there is one magnetic charge for each element $g \in G$. Quantum mechanically, we can have superpositions of these states, giving a one-particle Hilbert space spanned by $|g\rangle$ for all $g \in G$ (though strictly speaking, superpositions of charges in different conjugacy classes are meaningless, as will be explained in the next subsection).

Specifying the exchange properties of the charges involves making a choice of gauge. The easiest choice, which will be used in this thesis, is to keep all anyons ordered on a horizontal line. The exchange of particles, which can be clockwise or counterclockwise, is only allowed between adjacent pairs. In either case, the particle that passes below remains unchanged, while the particle that passes above gets conjugated. When the exchange is in the counterclockwise direction, the upper anyon gets conjugated by the flux of the lower one, whereas in the clockwise direction it gets conjugated by the inverse of the lower flux. This is depicted in Fig. 2.1.

One way to visualize these exchanges is to associate with each anyon a ray that is vertical in the plane, starting at the particle and proceeding upwards. Anyons are allowed to move freely through the plane, but every time an anyon crosses the ray of another particle, it gets conjugated by the flux of the owner of the ray (or by the inverse flux if crossing from left to right). Note that when a particle passes a group of anyons, it gets conjugated by the total flux of the anyons, which is given as the product from left to right of the individual fluxes.

Clearly, moving single anyons around can produce strange correlations throughout the system. However, moving a pair of anyons with a total flux that is trivial will not change the state of the system if the pair always passes below. This is why we will always be dealing with states of the form

$$\sum_g a_g |g\rangle \otimes |g^{-1}\rangle, \tag{2.1}$$

which correspond to a pair of anyons with trivial total flux. When dealing only with pairs of trivial total flux, we can swap any two pairs or bring any two pairs together without affecting the state of

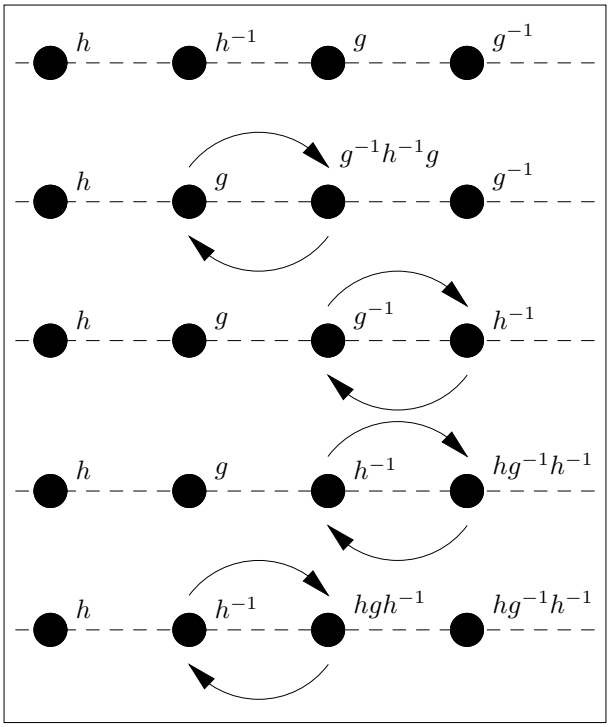


Figure 2.2: Conjugating a pair of anyons.

the rest of the system.

We do want to allow controlled interaction between pairs, though, and this is accomplished by a pass-through operation. The idea is to have one pair circle one anyon from the other pair. This will conjugate the fluxes of the pair that circles, but leave the other pair invariant. This operation is depicted using elementary exchanges in Fig. 2.2.

The net result of the pictured operation is

$$|h\rangle \otimes |h^{-1}\rangle \otimes |g\rangle \otimes |g^{-1}\rangle \longrightarrow |h\rangle \otimes |h^{-1}\rangle \otimes |hgh^{-1}\rangle \otimes |hg^{-1}h^{-1}\rangle, \tag{2.2}$$

which is a conjugation of the second pair by h. Conjugation by h^{-1} could be achieved by using counterclockwise exchanges in the picture.

For notational convenience, in this thesis we will generally only mention the flux of the left element of a pair. The above transformation will be written as

$$|h\rangle \otimes |g\rangle \longrightarrow |h\rangle \otimes |hgh^{-1}\rangle, \tag{2.3}$$

leaving the compensating fluxes implicit. While we will exclusively deal in this chapter with flux pairs with trivial flux, we will only explicitly refer to the second anyon when necessary to describe

the operations.

2.1.2 Electric charges and vacuum pairs

We now wish to focus on the operations of creating pairs from the vacuum and fusing pairs back into the vacuum. However, we must first briefly discuss the complete spectrum of particles, and that involves electric charges.

An electric charge is a particle with no flux that transforms as some non-trivial irreducible representation of the group G. A useful analogy is to think of the representation of G as the total spin of the particle. The internal state of the particle is then equivalent to the direction in which the spin is pointing.

The electric charge states can be labeled as $|R, V\rangle$, where R is a representation of G, and V is a vector that transforms in the representation R. The electric charges do not interact with each other, but when one of them circles a magnetic flux g, its state changes as

$$|R, V\rangle \longrightarrow |R, U_R(g)V\rangle, \tag{2.4}$$

where $U_R(g)$ is the matrix corresponding to g in the representation R. This is known as the Aharonov-Bohm effect.

While we can transform the state of an electric charge within the subspace of a representation, there are no operations (other than fusion, which destroys the particle) that can change the representation of a particle. Furthermore, the phase between states of different representations cannot be measured. We can therefore effectively describe the electric charges as having decohered into the different representations. In particle physics we would say that the different representations correspond to different superselection sectors.

The same thing happens to the magnetic charges. Different conjugacy classes live in different superselection sectors, so we can imagine that there is an automatic decoherence into different conjugacy classes. Superpositions of fluxes in different conjugacy classes are therefore meaningless.

The spectrum also contains particles with both electric and magnetic charge, which are called dyons. The only special feature is that the electric charge is a representation only of the subgroup of G that commutes with the flux. The aforementioned magnetic charges are just dyons with a trivial representation. The dyons also have superselection sectors that correspond to different conjugacy classes and representations.

The purpose of discussing the full spectrum, and the idea of superselection sectors, is to find out what kind of states we get when we create a pair of particles from the vacuum. The first thing to note is that each of the particles will instantly decohere into a specific conjugacy class and representation. Furthermore, because a pair created from the vacuum must have trivial total charge and flux, the

conjugacy classes must be inverses, and the representations must be conjugate representations.

Consider the case that the pair decoheres into plain magnetic charges, with the first one contained in the conjugacy class C. Because the combined state still has vacuum quantum numbers, the state must transform trivially when another flux is dragged around it. That is, it must be invariant under conjugation. There is only one such state:

$$|\mathrm{Vac}\,(C)\rangle = \frac{1}{\sqrt{|C|}} \sum_{g \in C} |g\rangle \otimes |g^{-1}\rangle. \tag{2.5}$$

The vacuum states for the other superselection sectors are also unique and have similar expressions. When a pair of anyons is created from the vacuum, it will start off in one of these states.

Another useful operation is to fuse two anyons together. Note that we are not talking about two anyon pairs, but rather two anyons, sometimes from the same pair, and sometimes from different pairs. The operation of fusion will turn the two particles into one, which must carry the total flux and charge of the two. It is also possible that the two anyons will have vacuum quantum numbers and will fuse back into the vacuum. In this case, no particle will be left behind, and their total mass will be transformed into some other medium, such as radiation. If $|\Psi\rangle$ is the combined state of the two anyons, and the first anyon is in the conjugacy class C, then the probability that the two will fuse into the vacuum is given by the standard rules of quantum mechanics:

$$P_{vacuum} = \left|\langle \mathrm{Vac}\,(C)|\Psi\rangle\right|^2. \tag{2.6}$$

After fusing two particles of different pairs, the fused particle may carry some flux. However, since the total flux of the original four particles was trivial, the total flux of all the remaining particles (including the product of the fusion) will be trivial as well. Therefore, it is possible to safely move the group of particles away from the bulk of the computation without disturbing our quantum state.

2.1.3 Qudits

Throughout this chapter it will be useful to perform computations with qudits rather than the usual qubits. We define our computational basis as the states $|i\rangle$ for $0 \leq i < d$, where we will assume that d is prime. The unitary Z and X gates can be defined as follows

$$Z|i\rangle = \omega^i |i\rangle, \tag{2.7}$$

$$X|i\rangle = |i+1\rangle, \tag{2.8}$$

where ω is a fixed non-trivial d^{th} root of unity, and sums are understood to be modulo d. The operators satisfy the commutation relation

$$ZX = XZ\omega. \tag{2.9}$$

As usual, the eigenstates of Z correspond to the computational basis. We can also introduce the eigenstates of X:

$$|\tilde{i}\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \omega^{-ij} |j\rangle, \tag{2.10}$$

which have the following transformations under the action of our unitary gates:

$$Z|\tilde{i}\rangle = |\widetilde{i-1}\rangle, \tag{2.11}$$

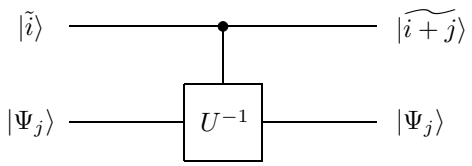
$$X|\tilde{i}\rangle = \omega^i |\tilde{i}\rangle. \tag{2.12}$$

2.1.4 Phase measurement

A very useful trick, used many times throughout this thesis, is Kitaev's phase estimation technique [Kit95]. In fact, we will only employ a special case of the technique, which we describe below.

Assume that we are working in a system with qudits, and we have an operator U with eigenvalues that are d^{th} roots of unity. We shall prove that being able to apply a controlled-U, and measure in the X basis, is equivalent to being able to measure the operator U.

Consider applying the circuit below to an eigenstate $|\Psi_j\rangle$ of U with eigenvalue ω^j:



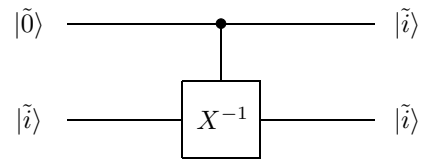
where the controlled-U^{-1} can be applied as $d-1$ controlled-U's. The circuit works as described because the controlled-U^{-1} leaves the bottom state invariant, but applies a Z^{-j} to the upper state. On a general state $|\phi\rangle = \sum_j c_j |\Psi_j\rangle$ expanded in terms of eigenvectors of U, the circuit produces the transformation

$$|\tilde{0}\rangle \otimes |\phi\rangle \longrightarrow \sum_j c_j |\tilde{j}\rangle \otimes |\Psi_j\rangle. \tag{2.13}$$

Clearly, a subsequent measurement of the first qudit in the X basis is equivalent to a non-destructive measurement of the original state in the U basis. We will use this technique in the next section to

measure the operators $X^a Z^b$.

In a later section we will employ the equivalent circuit



run in both the forwards and backwards directions, to change between the $|\tilde{i}\rangle$ states and the readily available $|\tilde{0}\rangle$ state that can be naturally produced from, and fused into, the vacuum.

2.2 A universal gate-set for anyons

A lot of the work in proving universality can be simplified by choosing a proper gate-set. We will employ a generalization of the gate-set used by Ogburn and Preskill [OP99, Pre97]. The gate-set, which involves measurements as well as unitary gates, can be applied to qudits when d is prime, which is the only case considered in this chapter.

The universal gate-set is:

1. Measure non-destructively Z
2. Measure non-destructively X
3. Apply Toffoli operators (to any set of three qudits)

where the qudit Toffoli is defined as

$$T|l, m, n\rangle = |l, m, lm + n\rangle. \tag{2.14}$$

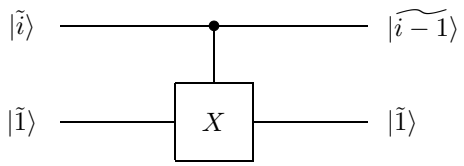
and all computations are done modulo d.

Though tangential to the main purpose of this chapter, the above gate-set is another answer to the question posed in [Shi02]. That is, given a Toffoli, what extra gates are required to complete a universal set? Of course, the answer provided by the above gate-set involves measurements in an integral way and is therefore different from the one proposed in [Shi02]. However, the above gate-set also addresses the question: Given classical computation (i.e., Toffoli and measurements of Z), what gates are needed to complete the universal set?

We now turn our attention to the proof of universality for the gate-set presented above. We note that Gottesman has already proven in [Got98] that for d prime, applying and measuring products of Z's and X's, plus a Toffoli, is universal for quantum computation. All we need to do in order to

prove universality is to show that we can apply and measure operators of the form $X^a Z^b$ using the above gates.

Measurements of X followed by measurements of Z can produce $|i\rangle$ ancillas for any i. Similarly, we can obtain $|\tilde{i}\rangle$ ancillas from measurements of Z followed by measurements of X. A controlled-sum can be made out of a Toffoli by fixing an input to a $|1\rangle$ ancilla. Because a controlled-sum is really a controlled-X, fixing the other input to $|1\rangle$ produces the X gate. On the other hand, a controlled-sum from a state to a $|\tilde{1}\rangle$ ancilla produces a Z on the state:



The general case of applying $X^a Z^b$ can be done by a series of X and Z gates. All that remains is to construct a method for measuring operators of the form $X^a Z^b$. First, we note that

$$\left(X^a Z^b\right)^d = \omega^{abd(d-1)/2} X^{ad} Z^{bd} = \begin{cases} 1 & d \text{ odd} \\ -1^{ab} & d = 2. \end{cases} \tag{2.15}$$

2.2.1 d odd case

The case $d = 2$ is rather complicated and will be handled separately. The general case d odd (remember we required d prime) is easy because the eigenvalues of $X^a Z^b$ are the d^{th} roots of unity just like those of X and Z. As discussed in the review of phase estimation, being able to apply a controlled-$X^a Z^b$, combined with measurements in the X basis (which includes preparation of X eigenstates) is sufficient to measure in the $X^a Z^b$ basis.

All that remains is to construct the controlled-$X^a Z^b$. That is, we need to be able to apply the gate

$$|n\rangle \otimes |\psi\rangle \longrightarrow |n\rangle \otimes \left(X^a Z^b\right)^n |\psi\rangle = |n\rangle \otimes X^{an} Z^{bn} \omega^{abn(n-1)/2} |\psi\rangle, \tag{2.16}$$

composed of a phase $|n, m\rangle \to \omega^{bnm+abn(n-1)/2}|n, m\rangle$ followed by controlled-sums. The controlled-sum is just a Toffoli with an input fixed to one. As for the phase, because we have a Toffoli, we have universal classical computation. We can thus compute $bnm + abn(n-1)/2$ in an ancilla, apply a Z to this ancilla, and then erase the computation.

2.2.2 $d = 2$ case

The $d = 2$ case is somewhat trickier because our gate-set is invariant under complex conjugation, and thus there is no way of distinguishing the two eigenstates of $ZX = iY$. We will solve this

problem by creating an ancilla that is an eigenstate of ZX, defining it to be the $+i$ eigenstate, and then using it to measure and build more eigenstates.

Assume we were given a state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \omega|1\rangle\right), \tag{2.17}$$

where $\omega^2 = -1$. Clearly, the state is equal to one of the two ZX eigenstates: $|\pm_Y\rangle = \left(|0\rangle \pm i|1\rangle\right)/\sqrt{2}$.

Using a controlled-ZX, which is built by the method described in the d odd case, we can produce copies of the state $|\Psi\rangle$. The idea, similar to the one used for phase estimation, is to apply the controlled-ZX from a state $|\tilde{0}\rangle$ to the state $|\Psi\rangle$. The target state is an eigenvector of ZX with eigenvalue ω, and therefore the relative phase is copied over to the first state:

$$|\tilde{0}\rangle \otimes |\Psi\rangle \longrightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + \omega|1\rangle\right) \otimes |\Psi\rangle = |\Psi\rangle \otimes |\Psi\rangle. \tag{2.18}$$

Notice that copying works independently of whether $|\Psi\rangle$ is the $+i$ or $-i$ eigenstate of ZX. Naturally, by subsequently applying a Z, we can also produce the orthogonal state $|\Phi\rangle = \left(|0\rangle - \omega|1\rangle\right)/\sqrt{2}$.

With our ancilla, we can also measure in this basis. This is done by applying a controlled-ZX to the ancilla from the state we want to measure:

$$|\Psi\rangle \otimes |\Psi\rangle \quad \longrightarrow \quad |\tilde{1}\rangle \otimes |\Psi\rangle, \tag{2.19}$$

$$|\Phi\rangle \otimes |\Psi\rangle \quad \longrightarrow \quad |\tilde{0}\rangle \otimes |\Psi\rangle, \tag{2.20}$$

and then measuring in the X basis.

As long as we are consistent in always using the same ancilla $|\Psi\rangle$, we will have broken the conjugation symmetry and found a way to label, create and measure eigenstates of ZX. Of course, we should keep many copies of the ancilla, which can be prepared from the original state. The operations above also allow us to error correct our set of ancillas by copying each, comparing the copies, and using majority voting to discard the damaged ancillas. Thus, even if there are some errors in preparation, or some of the ancillas decay over time, computation will still be feasible.

All that remains to be explained is how to create the first copy of $|\Psi\rangle$. Because a state with a density-matrix proportional to the identity can be written as

$$\rho = \frac{1}{2}I = \frac{1}{2}|+_Y\rangle\langle+_Y| + \frac{1}{2}|-_Y\rangle\langle-_Y|, \tag{2.21}$$

it is equivalent to having prepared an eigenstate of $ZX = iY$ chosen at random. The state $\rho = I/2$ can be produced by discarding one qubit of a bell state, and a bell state can be produced with a controlled-sum from a $|\tilde{0}\rangle$ ancilla to a $|0\rangle$ ancilla. Therefore, we have shown that we can produce

the initial eigenstate of ZX, and we have completed the proof that the gate-set presented at the beginning of this section is universal for quantum computation.

2.3 Universal computation for simple perfect groups

In this section we will prove that a set of anyons based on certain groups can perform universal quantum computations. Instead of dealing first with the general case of non-solvable groups, we will deal with the smaller set of groups that are both simple and perfect.

We remind the reader that non-solvable groups are those that contain a perfect subgroup, and a perfect group is any non-trivial group, whose commutator subgroup equals the full group: $[G, G] = G$. The property of simplicity means that the group has exactly two subgroups that are invariant under conjugation: the trivial group and the whole group. Because the commutator subgroup is invariant under conjugation, it should be clear that any simple non-abelian group is perfect. However, we shall refer to these groups as simple and perfect to remind the reader that we are dealing with a subcase of the general non-solvable case.

The set of simple perfect groups, which includes the groups A_n for $n > 4$, is powerful for computing because in some sense we can get from one non-trivial element to any other using operations that fix the identity. The general case of non-solvable groups will be deferred to Sec. 2.5, where we will show that a simple perfect group can be extracted from a non-solvable group.

2.3.1 Requirements for the physical system

Here we list the operations, ancillas and measurements that we assume are available on any realistic anyonic system, and which we will use to build our quantum gate-set:

1. We can braid or exchange any two particles.

2. We can fuse a pair of anyons and detect whether there is a particle left behind or whether they had vacuum quantum numbers.

3. We can produce a pair of anyons in a state that is chosen at random from the two particle subspace that has vacuum quantum numbers.

4. We have ancilla pairs of the form $|g\rangle \otimes |g^{-1}\rangle$ for any $g \in G$, where the individual anyons have trivial electric charge.

We remind the reader again that even though all our anyons are used in pairs of trivial total flux, we will generally only mention one of the anyons of the pair. These conventions also apply to ancillas, which means that we will refer to the $|g\rangle \otimes |g^{-1}\rangle$ state as an ancilla of flux g.

While the first three requirements are natural operations for a laboratory system, it is not clear where the ancillas would come from. Depending on the physical realization there may be many ways of obtaining the ancilla reservoir. We discuss one such scheme in Sec. 2.6.

2.3.2 Computational basis

Let G be a simple and perfect finite group. Let a and b be two non-commuting elements of G. Let d be the smallest integer such that $a^d b a^{-d} = b$. We can assume that d is prime, otherwise we could replace a by $a^{d/p}$ where p is some prime that divides d.

It turns out that every simple non-abelian group has even order. This was first conjectured by Burnside [Bur55] in 1911, and proven by Feit and Thompson [FT63] in 1963 (in fact, the complete classification of simple finite groups was completed in the early 1980s, see for instance [GLS94]). Using Sylow's theorems, the fact that every simple group has even order means that they all include a non-trivial element a such that $a^2 = 1$. Therefore, we could always work with a basis of qubits. However, we will present the general qudit case both for its elegance, and because in some instances a basis of qudits is more convenient.

We will work with a basis of qudits of trivial net flux

$$|n\rangle = |a^n b a^{-n}\rangle \otimes |a^n b^{-1} a^{-n}\rangle \tag{2.22}$$

for $0 \leq n < d$, where we have explicitly described both anyons of the pair.

It should be clear that we can initialize the computer by filling up the computational space with $|0\rangle$ ancillas. We turn now to the task of constructing the gates presented in Sec. 2.2.

2.3.3 Conjugation by a function

We begin by describing the technique of conjugation by a function, which is especially powerful for simple perfect groups. In Sec. 2.1.1 we showed that we could perform the transformation

$$|h\rangle \otimes |g\rangle \longrightarrow |h\rangle \otimes |hgh^{-1}\rangle, \tag{2.23}$$

where we conjugate the second anyon by the flux of the first, while the first anyon remains invariant. We can also conjugate an anyon by a product $h_1 h_2 \cdots h_n$

$$|g\rangle \longrightarrow |h_1 h_2 \cdots h_n g h_n^{-1} \cdots h_2^{-1} h_1^{-1}\rangle, \tag{2.24}$$

where the $\{h_i\}$ are fluxes of other anyons that remain unchanged throughout this process. The procedure is done by first conjugating by h_n, then by h_{n-1}, and proceeding leftward until we finally

conjugate by h_1.

The above procedure is not terribly useful if all the $\{h_i\}$ are fluxes of fixed ancillas because we could have equivalently conjugated by a single ancilla of flux $h = h_1 h_2 \cdots h_n$. However, some of the fluxes in the product could correspond to anyons that represent qudits of unknown state. In this case we can think of the above operation as conjugation by a function of the fluxes of certain qudits.

Let us consider what kind of functions can be applied in this way. Clearly we are speaking about functions that can be written as products of elements of G. The elements can include known constants if we use our ancillas to conjugate. We can also include the flux of a qudit, which will be of the form $a^i b a^{-i}$ if the qudit is in the computational basis (though this may not be the case when we are trying to correct leakage). Finally, we can include in the product the inverse of the flux of a qudit, as discussed in Sec. 2.1.1.

In conclusion, given n qudits with fluxes g_1 through g_n, and a function $f(g_1, \ldots, g_{n-1})$ of the first $n-1$ qudits, we can conjugate the last qudit by f

$$|g_n\rangle \longrightarrow |f(g_1, \ldots, g_{n-1}) g_n f(g_1, \ldots, g_{n-1})^{-1}\rangle, \tag{2.25}$$

provided that the function f can be written in product form. By product form, we mean that f is a product of the inputs $\{g_i\}$, their inverses $\{g_i^{-1}\}$, and fixed elements of G, each of which may appear more than once, or not at all. For example, a valid function would be $f(g_1, g_2) = a g_2 b g_1^{-1} c g_1^{-1} d$ with $a, b, c, d \in G$. Furthermore, this transformation does not change the flux of the first $n-1$ qudits, though it may entangle them with the last qudit.

2.3.4 Toffoli gate

To build the Toffoli gate we must be able to conjugate the third qudit by the function $f(g_1, g_2)$, which depends on the fluxes of the first two qudits as

$$f(a^i b a^{-i}, a^j b a^{-j}) = a^{ij} \tag{2.26}$$

and is arbitrary for values of g_1 and g_2 that are not in the computational basis. If the third qudit is in the state $a^k b a^{-k}$, conjugation by f produces the transformation

$$|a^k b a^{-k}\rangle \longrightarrow |a^{ij+k} b a^{-ij-k}\rangle, \tag{2.27}$$

which is the desired Toffoli gate.

Given the discussion in the previous subsection, we are left with the task of expressing the function f in product form. However, it turns out that for simple and perfect groups every function has such an expression:

Theorem 1. *If G is a simple and perfect finite group, then any function $f(g_1, \ldots, g_n) : G^n \to G$ can be expressed as a product of the inputs $\{g_i\}$, their inverses $\{g_i^{-1}\}$ and fixed elements of G, any of which may appear multiple times in the product.*

Not only does the above theorem prove that Toffoli gates are possible for any simple and perfect group, but it directly proves that any classical function can be computed.

The proof of the theorem, which is mostly constructive, is somewhat long and will be deferred to Sec. 2.7. However, to make this seem plausible to the casual reader, we would like to illustrate the basic steps needed to build a Toffoli gate for qubits.

The main idea behind the construction is that the function f is basically a logical AND of the inputs. A commutator makes a good logical AND because it equals the identity if either of its inputs are the identity. Furthermore, the commutator function can be expanded as a product of its inputs. Therefore, we would like the first input to take values 1 or c and the second input to take values 1 or d, with the requirement that d not commute with c, so that we can put them into a commutator.

Let g_1 denote the flux of the first qubit, and g_2 the flux of the second qubit. Each takes values $g_i \in \left\{b, aba^{-1}\right\}$. Define the new variables $g_i' = g_i b^{-1} \in \{1, c\}$, where $c \equiv [a, b] \equiv aba^{-1}b^{-1}$. It is sufficient to show that we can express the Toffoli function as a product of g_1', g_2', their inverses and fixed ancillas.

Choose an element d that does not commute with c and define $e \equiv [c, d]$. Imagine we could find two functions of one element that can be expressed in product form, such that

$$h_1(c) = d \qquad h_1(1) = 1, \tag{2.28}$$

$$h_2(e) = a \qquad h_2(1) = 1. \tag{2.29}$$

Using these functions, the Toffoli function can be written as

$$f(g_1, g_2) \quad = \quad h_2\Big(\Big[g_1',\ h_1(g_2') \Big] \Big), \tag{2.30}$$

which when expanded out is a product of the correct form.

The existence of the functions h_i, which is discussed in more detail in the full proof of the theorem, is a consequence of G being simple. For any element $c \in G$, the group generated by its conjugacy class $C(c)$ is a normal subgroup. Because G is simple, this subgroup must equal the full group. Therefore, every element $d \in G$ has an expression of the form $d = x_1 c x_1^{-1} x_2 c x_2^{-1} \cdots x_n c x_n^{-1}$ for some n and some elements $\{x_i\} \in G$. We can use the expression to construct h_1:

$$h_1(g) = x_1 g x_1^{-1} x_2 g x_2^{-1} \cdots x_n g x_n^{-1}, \tag{2.31}$$

and a similar construction builds h_2.

For a concrete example we can work with $G = A_5$. We begin by choosing an element a, which must satisfy $a^2 = I$, if we wish to work with qubits ($d = 2$). Because of the symmetries of the group, all choices are equivalent to $a = (12)(34)$. The next step would involve choosing an element b that does not commute with a, and an element d that does not commute with $c \equiv [a, b]$. While any choice can produce a Toffoli, the required h_1 function will be simplified if we can make c and d fall in the same conjugacy class. The same can be said for h_2 if $e \equiv [c, d]$ and a are in the same conjugacy class.

At this point, a little trial and error yield $b = (345)$ and $d = (234)$. The computational basis is now defined as

$$\begin{aligned} |0\rangle &= |b\rangle = |(345)\rangle, \\ |1\rangle &= |aba^{-1}\rangle = |(435)\rangle, \end{aligned} \tag{2.32}$$

and the remaining group elements are fixed as

$$\begin{aligned} c &= \left(aba^{-1}\right) b^{-1} = (435)(435) = (345), \\ e &= \left(cdc^{-1}\right) d^{-1} = (245)(324) = (25)(34). \end{aligned} \tag{2.33}$$

The h_i functions, which are the only non-constructive part of the proof, can be built as simple conjugations because of the choices we made earlier:

$$h_1(g) = h_2(g) = (521)g(125), \tag{2.34}$$

where both happen to be the same function by coincidence. Putting all the steps together we have a function

$$\begin{aligned} f\left(g_1, g_2\right) &= (521)\Big[g_1(435),\, (521)g_2(435)(125)\Big](125) \\ &= (521)g_1(435)(521)g_2(435)(125)(345)g_1^{-1}(521)(345)g_2^{-1}(125)(125) \\ &= (521)g_1(14352)g_2(124)g_1^{-1}(15342)g_2^{-1}(521), \end{aligned} \tag{2.35}$$

which can be applied with nine elementary conjugations.

2.3.5 Measuring Z

The basic idea behind measuring in the computational basis is that if we fuse a flux with another flux of the inverse group element, there is a finite chance that they will have vacuum quantum numbers

and disappear. On the other hand, if the product of the two fluxes is not unity then there must be a particle left behind to carry the remaining flux (i.e., the total flux is always conserved).

At this point it might be useful to remind the reader why a fusion of g with g^{-1} will not always turn into the vacuum. The short story is that the combined state is not invariant when another flux encircles them, implying that they have an electric charge component. The state that has vacuum quantum numbers is invariant under the effect of all fluxes, and hence is the sum of all the states in the conjugacy class of g, with the same phase. We can figure out the probability of fusion into the vacuum by calculating the overlap of the vacuum state with the state of two anyons to be fused. The result is

$$P = \left|\langle \mathrm{Vac}\,(C)| \left(|g\rangle \otimes |g^{-1}\rangle\right)\right|^2 = \frac{1}{|C(g)|} \tag{2.36}$$

where $C(g)$ is the conjugacy class of g, and the vacuum state was defined in Sec. 2.1.2.

Because one fusion will only probabilistically tell us the desired result, we should repeat the measurement many times to obtain a sufficient degree of accuracy. Besides, if we are working with qudits with $d > 2$ we need to test fusion with at least two different fluxes. We therefore need to have many copies of the state to be measured.

Because of the no cloning theorem, copying cannot be done exactly, but the transformation

$$\sum_i C_i|i\rangle \longrightarrow \sum_i C_i|i\rangle \otimes |i\rangle \otimes |i\rangle \otimes \cdots \otimes |i\rangle \tag{2.37}$$

means that we can measure each of the separate copies in the Z basis and expect to get the same answer. The above transformation can be done with a controlled-sum (Toffoli with one input fixed to $|1\rangle$) from the original state to a $|0\rangle$ ancilla. Repeating this controlled-sum with many target ancillas will produce the above entangled state.

To summarize, the procedure for measuring in the Z basis is first to create an entangled state using a controlled-sum. Then try to fuse each of the qudits with one of the inverses of the fluxes that are Z eigenstates. Eventually, one will disappear into the vacuum, and the inverse of the flux of that ancilla is the result. Even in the presence of errors, this measurement will have a good fidelity because the probability of failure is exponentially small in the number of fusions.

A final note is that, because we are always dealing with pairs of fluxes, what fusion really means is that we fuse the first anyon of our qubit with the first anyon of the ancilla.

2.3.6 Constructing the zero eigenvector of X

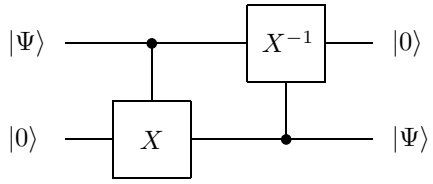
For the next gates, we are going to need a supply of states that are eigenvectors of X with zero eigenvalue:

$$|\tilde{0}\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |i\rangle. \tag{2.38}$$

We will produce them out of pairs of anyons with vacuum quantum numbers. As usual we will just discuss one member of the pair and assume that the equivalent operations are being performed on the other anyon.

One of the possible states that (when paired) have vacuum quantum numbers is the sum of fluxes in the conjugacy class of b. This is approximately what we want. Sadly, in general, a state created from the vacuum will be a mix of this desired state plus other states, including states that involve dyonic particles (particles with both electric and magnetic charge). We will have to filter through all this noise to get our X eigenstate.

The procedure that we will describe below is effectively an incomplete swap that has been extended to the full Hilbert space in a logical way. In the computational basis, the operations act as



which performs a swap provided that the second qudit started in the $|0\rangle$ state. Outside of the computational basis, though, the operations are chosen so that we can detect whether we obtained the desired $|\tilde{0}\rangle$ state or not.

We start with two qudit states, one created from the vacuum and one which is a $|0\rangle$ ancilla:

$$|\text{Vac}\rangle \otimes |0\rangle = \left(C|\tilde{0}\rangle + D|\Psi_\perp\rangle\right) \otimes |0\rangle, \tag{2.39}$$

where $|\Psi_\perp\rangle$ is a state orthogonal to the computational subspace. If the vacuum pair decohered into a superselection sector other than the one that contains the computational basis, the constant C will be zero. This will not be a problem as we will be able to detect this case and then start again from this step.

Using the theorem from Sec. 2.3.4, we can conjugate the $|0\rangle$ ancilla by a function of the flux of

the vacuum pair that has the following form:

$$\begin{aligned} f(a^i b a^{-i}) &= a^i, \\ f(\text{anything else}) &= I, \end{aligned} \tag{2.40}$$

which is essentially a controlled-sum that has been properly defined outside the computational basis.

The state of the combined system after conjugation will be

$$\frac{C}{\sqrt{d}} \sum_{i=0}^{d-1} |a^i b a^{-i}\rangle \otimes |a^i b a^{-i}\rangle + \sum_{i=0}^{d-1} D_i |\Psi_{i\perp}\rangle \otimes |a^i b a^{-i}\rangle, \tag{2.41}$$

where $\{D_i\}$ are some constants, and $\{|\Psi_{i\perp}\rangle\}$ are states perpendicular to the computational basis. Note that the states $|\Psi_{i\perp}\rangle$ for $i \neq 0$ are the ones that have flux $a^i b a^{-i}$ but have non-trivial charge. The state $|\Psi_{0\perp}\rangle$ includes all the other fluxes and charges. Depending on the superselection sector in which the vacuum state was created, many of the constants C and $\{D_i\}$ will be zero.

Now we conjugate by f^{-1} from the ancilla to the vacuum state yielding

$$C|b\rangle \otimes |\tilde{0}\rangle + \sum_{i=0}^{d-1} D_i |\Psi'_{i\perp}\rangle \otimes |a^i b a^{-i}\rangle, \tag{2.42}$$

where $\{|\Psi'_{0\perp}\rangle\} = \{|\Psi_{0\perp}\rangle\}$ and the states $\{|\Psi'_{i\perp}\rangle, i > 0\}$ have flux b but non-trivial charge.

Now we try to fuse the first qudit with an ancilla of flux b^{-1} and trivial charge. The only state that can fuse into the vacuum with the ancilla is $|b\rangle$, and this will happen with finite probability. Note that the ancilla can never vanish into the vacuum with a state with charge because there is no way of extending the basis to be invariant under the stabilizer group of the flux.

In the end, if the particles disappear into the vacuum, the ancilla is left in the desired X eigenstate. Otherwise, we repeat the procedure from the beginning until eventually the state appears.

2.3.7 Choosing a d^{th} root of unity

Before we continue building our gate-set, we have to address a problem that appears for $d > 2$, similar to the problem that occurred for $d = 2$ when proving that the gate-set is universal.

So far, we have defined everything in terms of ω, a non-trivial d^{th} root of unity. But there are $d-1$ of these, and there is a symmetry that interchanges them. We will have to break this symmetry by using an ancilla.

In particular, we need an ancilla that is an eigenstate of X with eigenvalue not equal to 1. We

will then define this state to be the $|\tilde{1}\rangle$ state in the X basis, i.e.,

$$|\tilde{1}\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} \omega^{-i} |i\rangle, \tag{2.43}$$

which has eigenvalue ω, thus fixing our root of unity. We then define the other X eigenstates by

$$|\tilde{n}\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} \omega^{-ni} |i\rangle, \tag{2.44}$$

and the operator Z by $|\tilde{n}\rangle \rightarrow |\widetilde{n-1}\rangle$.

How do we produce the first $|\tilde{1}\rangle$ in terms of which everything is defined? We start with a $|\tilde{0}\rangle$ (which is always well defined and which we know how to construct from the previous section), and we apply a controlled-X^{-1} (which is a classical function, and thus computable from the Toffoli) from this ancilla to a $|0\rangle$ ancilla, which produces the output

$$|\tilde{0}\rangle \otimes |0\rangle \longrightarrow \frac{1}{\sqrt{d}} \sum_i |\tilde{i}\rangle \otimes |\tilde{i}\rangle. \tag{2.45}$$

If we discard the second state, we will have a mixed state that is a combination of the different X eigenstates. This is equivalent to being handed an arbitrarily chosen X eigenstate, which we will call $|\tilde{i}\rangle$.

We can obtain copies of this state by applying a controlled-X^{-1} from a $|\tilde{0}\rangle$ ancilla to this state, which applies the transformation

$$|\tilde{0}\rangle \otimes |\tilde{i}\rangle \longrightarrow |\tilde{i}\rangle \otimes |\tilde{i}\rangle. \tag{2.46}$$

We can thus build arbitrarily many copies of the state. We still have to worry that this might be the $|\tilde{0}\rangle$ state. However, below in the section for measuring X, we will give a procedure to detect the $|\tilde{0}\rangle$, which does not rely on having $|\tilde{1}\rangle$ ancillas. If we determine that $i = 0$, we throw away all the copies and start over (this will only happen with probability $1/d$). Otherwise, we relabel our state as $|\tilde{1}\rangle$, fixing a value for ω.

Because we can copy the $|\tilde{1}\rangle$ state, and below we will also show how to measure it, we can build a reservoir of ancillas in this state, which will be used for all future computations. We can even use copying, comparing and majority voting to error correct our reservoir, thus allowing for computation even in the presence of noise.

2.3.8 Measuring X

The last gate needed for universality is the measurement of X. The basic idea is to fuse the pair of anyons that form the state to be measured. The $|\tilde{0}\rangle$ eigenstate will have some overlap with the vacuum and will vanish with probability $p = d/|C(b)|$, where $C(b)$ is the conjugacy class of b.

The other X eigenstates have zero probability of vanishing because $|\tilde{i}\rangle = \frac{1}{\sqrt{d}} \sum_i \omega^{-i} |a^i b a^{-i}\rangle$ is orthogonal to the vacuum for $i > 0$. To detect the state $|\tilde{i}\rangle$ we first apply a Z^i and then use the above fusion procedure. The Z gate can be applied as a controlled-sum with a $|\tilde{1}\rangle$ target as discussed in Sec. 2.2.

Of course, the above will require us to have many copies on which to measure, which means we need to perform the transformation

$$\sum_i C_i |\tilde{i}\rangle \longrightarrow \sum_i C_i |\tilde{i}\rangle \otimes |\tilde{i}\rangle \otimes |\tilde{i}\rangle \otimes \cdots \otimes |\tilde{i}\rangle, \tag{2.47}$$

which is done using a controlled-X^{-1} with a $|\tilde{0}\rangle$ ancilla as control and the state to be copied as target.

To perform the measurement non-destructively, we can fuse all but one of the copies of the state. Alternatively, using the Z gate and $|\tilde{0}\rangle$ ancillas, we can always produce the rest of the $|\tilde{i}\rangle$ states. The rest of the logic is similar to the Z measurement procedure.

Having completed the construction of the universal gates, we have proven that universal quantum computation can be performed with anyons from simple and perfect finite groups. We now turn to the question of whether these operations can be performed in a fault-tolerant fashion.

2.4 Leakage correction

In this section we will discuss both the motivation and the techniques needed to implement error correction and fault tolerance in the software of an anyonic computer. The main result will be the construction of a leakage correction circuit for anyons, which enables the use of the standard techniques for handling errors.

2.4.1 Motivation

Any quantum system that uses non-locality to protect its data will be susceptible to errors if a large number of its local components are damaged simultaneously. The probability for failure is generally exponentially small in the size of the system and is zero in the theoretical limit of an infinite system. However, all physical systems are finite. Furthermore, practical considerations may force a given setup to have a size such that the error of probability is small but non-negligible.

In the case of anyons, errors can occur due to quantum tunneling, which is an effect of the high-energy degrees of freedom that were frozen out to obtain a two-dimensional discrete gauge theory. The probability of this type of error goes as e^{-mL}, where m is the mass of the lightest particle that can mediate a charge interaction, and L is the separation between anyons.

Finite temperature effects are another source of error. These effects involve the creation of charge pairs from the vacuum. Because these pairs have trivial total charge, even if they braid with a computational anyon, the net charges of the collective three particle excitation will still be correct. However, if one of these particles separates from the group, or separately braids with another anyon, then errors will be introduced. The density of the thermal excitations goes as $e^{-\Delta/T}$, where Δ is the mass gap and T is the temperature.

A good anyonic quantum computer should therefore have $L >> m$ and $T << \Delta$. In some implementations, however, it may be more practical to accept a small error rate from the hardware, and then to correct it using standard quantum error correction techniques. For such cases, we present below the necessary steps needed to implement software based error correction for anyons.

While any of the error correcting codes can be used, most techniques require embedding a code space inside a Hilbert space on which we can do universal quantum computation. However, in the case at hand, our computational states are embedded in a Hilbert space (the states with arbitrary flux and charge) in which we cannot perform universal quantum computation. Therefore, before starting the recovery protocol, we must first deal with states that have leaked out of the computational subspace (the subspace in which we can perform universal computations).

2.4.2 Implementation

To deal with leakage errors we can construct a version of the swap-if-leaked gate described by Kempe et al. [KBDW01]. The idea behind the gate is to implement a projective measurement that can distinguish the computational subspace from its complement. If a state is found to be in the computational subspace, it is left alone. Otherwise, it is replaced with an arbitrary ancilla that is in the computational subspace. The ancilla will still be an error, but one that is correctable by standard quantum error correcting codes. In fact, the general methods of quantum error correction and fault-tolerant computation can be applied to anyons as long as we can reliably project leaked qudits into a state in the computational subspace.

We again focus on the case of simple and perfect groups, and defer the general leakage correction protocol to the next section. In the current formalism, the computational basis is the set of states of a pair of anyons with zero total magnetic charge, where each anyon has zero electric charge and a magnetic flux of the form $a^i b a^{-i}$ or its inverse.

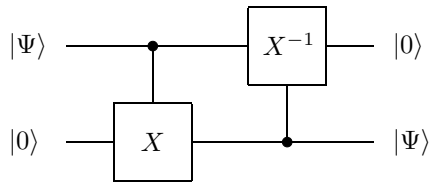
The first type of error that we will deal with is when the total magnetic flux of the pair is non-trivial. This is a particularly grievous error because, if we drag around a pair with a non-trivial net

flux, we could be introducing errors into all the other qudits. Furthermore, our assumption that we can perform the operation $h, g \to h, hgh^{-1}$ relied on the fact that the second pair had zero net magnetic flux, so it is important that we detect and fix this error first.

To detect a net flux, we take an ancilla $|g\rangle \otimes |g^{-1}\rangle$ and encircle it by the qudit we are performing the leakage correction on. The ancilla will get conjugated by the net flux of the qudit, and the qudit will get conjugated by the net flux of the ancilla, which should be zero. We then fuse the ancilla with a pair with opposite flux. If the net flux of the qudit is in the stabilizer of g, the fusion will have vacuum quantum numbers with a finite probability, whereas if the conjugation changed the flux of the ancilla, there will always be a particle left behind. If we repeat this many times with many different ancillas $|g\rangle \otimes |g^{-1}\rangle$, with good statistical confidence we will be able to tell if the net flux of the qudit is in the stabilizer of g. Because G has no center, the intersection of all stabilizers is the identity, and hence repeating the above with sufficiently many different elements g, we can detect a non-zero net magnetic charge.

If we detected a net flux, we replace the state with an ancilla in the state $|0\rangle$. Of course, we must be very careful when moving the damaged ancilla pair out of the region of qudits, so as not to damage other states. That is, when moving past other anyons, we always do so in the direction in which the damaged pair gets conjugated and the good qudits are unaffected.

In the case when the qudit passes the above test, then we have projected into the zero net flux subspace, but otherwise left the state unchanged. The next step is to deal with electric charge. Because it is very difficult to measure the electric charge of a single anyon, we will start with a fresh ancilla $|0\rangle$, made from two anyons neither of which have electric charge, and copy the state over. Once again we will be using the incomplete swap circuit



when acting on the computational basis. Of course, the heart of a leakage detection algorithm is how to extend the operations outside of the computational subspace. The procedure cannot be described simply by a circuit, and therefore we will present a way of completing the controlled-sum gate so that the above operation will always yield a state that is in the computational subspace.

The following procedure is almost identical to the one used to produce $|\tilde{0}\rangle$ states. This is because $|\tilde{0}\rangle$ states are obtained by taking a vacuum state and projecting to the computational basis, which is primarily leakage detection. The main difference is that when doing leakage detection, we only get one chance of using the qudit (because of the no-cloning theorem), but if the state leaked, it is

acceptable to replace it by anything in the computational basis. The latter is clearly not acceptable when creating $|\tilde{0}\rangle$ ancillas.

We will use the incomplete swap procedure for the second round of leakage detection. Recall that by this point we have projected the qudit into the zero net flux subspace. Take the qudit and a $|0\rangle$ ancilla, and conjugate the ancilla by a function of the qudit's flux:

$$\begin{aligned} f(a^i b a^{-i}) &= a^i \\ f(\text{anything else}) &= I. \end{aligned} \tag{2.48}$$

This is the same extension of a controlled sum that was used to produce $|\tilde{0}\rangle$ ancillas.

Afterward, we conjugate the original qudit by $f(g)^{-1}$, where g is the flux of the ancilla. Note that because we know at this point that the original qudit has net flux zero, the state of the ancilla will not exit the computational basis during this operation (though it might change within the computational basis if the original state had non-zero electric charge). The result of the past two controlled-sums is:

$$|\psi_{\|}\rangle \otimes |0\rangle + |\psi_{\perp}\rangle \otimes |0\rangle \longrightarrow |0\rangle \otimes |\psi_{\|}\rangle + \sum_{i=0}^{d-1} |\psi_{\perp i}\rangle \otimes |i\rangle, \tag{2.49}$$

where parallel and perpendicular refer to inside and outside the computational basis, and none of the ψ states are normalized. Finally, we replace the original pair with the ancilla pair and discard the original pair.

Clearly, the new state will be in the computational basis. Furthermore, if the original state was in the computational basis, then the new state will be equal to the old state, and unentangled with the old anyons.

Having complemented our gate-set with a leakage correction scheme, we have proven not only that we can do universal quantum computation with anyons, but that these computations can be made fault-tolerant.

2.5 Universal computation for non-solvable groups

We will now generalize the results of the previous section to any non-solvable group. Unfortunately, in our proofs for the simple perfect case, we made extensive use of the fact that we can compute any classical function simply by multiplying the inputs with ancillas. This is no longer true, even if we restrict ourselves just to perfect groups that are not simple. The quickest example is $A_5 \times A_5$, which is perfect, but has two normal subgroups given by each of the A_5 factors. Thus, if our two inputs are 1×1 and $g \times 1$, there is no expression made out of products in which the results differ in

the second factor.

The above example can easily be fixed by working within one A_5 subgroup. In general, though, even this is not possible, as not all perfect groups have a perfect and simple subgroup. However, the following theorem comes to the rescue:

Theorem 2. *If G is a non-solvable finite group, then there exists a normal subgroup P of G and a subgroup N, normal in P, such that P/N is perfect and simple.*

Once again we defer the proof to Sec. 2.7.

What the theorem tells us is that we want to work with cosets of N in P. That is, we would like to replace our old flux eigenstates with states that are labeled by elements in P/N and invariant under N. A good guess would be

$$|x\rangle = \frac{1}{\sqrt{|N|}} \sum_{n \in N} |x'n\rangle, \tag{2.50}$$

where x is an element of P/N, and x' is an element in the coset that x represents. More specifically, if $f : P \to P/N$ is the canonical epimorphism that maps elements to cosets, then we require that $f(x') = x$. The particular choice of x' has no effect on the above definition.

The above is a good guess but not quite right. A given coset may intersect many different conjugacy classes of G, each of which lies in a different superselection sector. Thus, we are effectively working with mixed states.

Remembering that we really want to keep our anyons in pairs of zero net flux, the right choice for the new states is

$$\rho_x = \frac{1}{|N|} \sum_{C \in \mathcal{C}(G)} \left[\left(\sum_{x' \in (C \cap f^{-1}(x))} |x'\rangle \otimes |x'^{-1}\rangle \right) \otimes \left(\sum_{x' \in (C \cap f^{-1}(x))} \langle x'| \otimes \langle x'^{-1}| \right) \right], \tag{2.51}$$

where again x is an element of P/N, and $\mathcal{C}(G)$ is the set of conjugacy classes of G.

These states have the nice property that when conjugated by any element $h' \in P$ (or equivalently, when a flux $h' \in P$ is dragged around them), the effect only depends on the coset $f(h')$ of h', and generates the transformation

$$\rho_g \longrightarrow \rho_{f(h')\, g\, f(h')^{-1}}. \tag{2.52}$$

Because of this, if we use the usual scheme of passing one pair of anyons in between another, and they are both prepared in states of the above form, the net effect is that the inner pair will get conjugated by the outer pair as

$$\rho_h \otimes \rho_g \longrightarrow \rho_h \otimes \rho_{hgh^{-1}}, \tag{2.53}$$

keeping the pair unentangled.

2.5.1 New requirements for the physical system

While the operations of braiding, fusion and vacuum pair creation described in Sec. 2.3.1 all seem like reasonable requirements to demand from the physical system, the requirement of flux ancillas is somewhat harder to justify.

In particular, take the case of a group that has a non-trivial center, which can occur even if the group is perfect. Consider two fluxes g and cg that differ by multiplication of an element c in the center. These two fluxes cannot be distinguished by conjugation, since $cgx(cg)^{-1} = gxg^{-1}$. Thus, it may be a difficult problem to distill these flux eigenstates from the vacuum.

A more reasonable assumption is to require the existence of ancillas only for the fluxes in the perfect subgroup. Another improvement might be to assume that we only have ancillas in the mixed states ρ_x defined above, where $x \in P/N$. These states might be easier to produce because they are obtained from the vacuum by first throwing away the anyons with flux not in P or with non-trivial charge, and then projecting to a definite coset of N in P. Therefore, we will replace our old requirement for the existence of flux ancillas by:

4' We have ancillas in the state ρ_x for any $x \in P/N$.

It would be highly desirable to be able to prove that requirements 1 through 4 are sufficient to create the states in 4'. Unfortunately, it appears that requirements 1-3 combined with 4' may neither be a subset nor a superset of requirements 1-4. Thus, in a sense, we are imposing a different set of requirements for this section. One ameliorating fact is that in the case when P is simple, the states ρ_x are just flux eigenstates. We therefore could have used requirement 4' for all sections of this chapter. We will not attempt to describe in Sec. 2.6 a protocol by which these generalized ancillas can be created, however.

2.5.2 Universal computation

As in Sec. 2.3.2, we choose two elements $a, b \in P/N$ such that $a^d b a^{-d} = b$ for some prime d, and $a^i b a^{-i} \neq b$ for $0 < i < d$. We then define our computational basis states as

$$\rho_i = \rho_{a^i b a^{-i}}, \tag{2.54}$$

which we define as eigenstates of the Z operator. The X operator is defined by $X(\rho_i) \equiv X\rho_i X^\dagger = \rho_{i+1}$, and its eigenstates can be obtained using the projection operators $P_i = \sum_{j=0}^{d-1} \omega^{-i} X^i/d$ by

$$\rho_{\tilde{i}} = d \times P_i \rho_0 P_i^\dagger = \frac{1}{d} \left(\sum_{j=0}^{d-1} \omega^{-i} X^i \right) \rho_0 \left(\sum_{j=0}^{d-1} \omega^{i} X^{i\dagger} \right). \tag{2.55}$$

At this point proving that universal quantum computation can be achieved is fairly straightforward and is almost identical to the discussion in Sec. 2.3. The major differences occur when we have to deal with states outside of the computational basis, that is, when creating $\rho_{\tilde{0}}$ states and when dealing with leakage correction. Both of these issues will be dealt with in the next subsection. As for the rest of the operations, we will only give a very brief discussion:

Because the ρ_x states have the same braiding properties as those of the fluxes of a group P/N (and in particular two Z eigenstates remain unentangled after braiding), the same method for producing a Toffoli applies to them.

Measuring Z is easy because the ρ_i states have support in orthogonal subspaces. The copy (using the Toffoli) and fuse with ancillas procedure will work just as well as before.

For the interested reader, we will carry out below some of the calculations needed to deal with X eigenstates and prove universality. Most of the results seem almost miraculous when expressed in the language of density operators. However, the reader should bear in mind that we are only using density operators to account for the different superselection sectors. If we just fixed a superselection sector for each particle, we would be dealing with pure states, and all of the proofs from the past section would carry through.

We begin by studying the action of the controlled-X^{-1} on X eigenstates:

$$\begin{aligned}
\rho_{\tilde{m}} \otimes \rho_{\tilde{n}} &= \frac{1}{d} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} \omega^{-im+jm} X^i \rho_0 X^{j\dagger} \otimes \rho_{\tilde{n}} \\
&\longrightarrow \frac{1}{d} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} \omega^{-im+jm} X^i \rho_0 X^{j\dagger} \otimes X^{-i} \rho_{\tilde{n}} X^{-j\dagger} \\
&= \frac{1}{d} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} \omega^{-im+jm} X^i \rho_0 X^{j\dagger} \otimes \omega^{-in} \rho_{\tilde{n}} \omega^{jn} \\
&= \frac{1}{d} \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} \omega^{-i+j(m+n)} X^i \rho_0 X^{j\dagger} \otimes \rho_{\tilde{n}} \\
&= \rho_{\widetilde{m+n}} \otimes \rho_{\tilde{n}},
\end{aligned} \tag{2.56}$$

which is equivalent to its action on pure states. Therefore, once we have $\rho_{\tilde{0}}$ states, we can use the same trick as before to break the symmetry and obtain a d^{th} root of unity. That is, we do a

controlled-X^{-1} from the $\rho_{\tilde{0}}$ with a ρ_0 target to create the state

$$\begin{aligned}
\rho_{\tilde{0}} \otimes \rho_0 &= \left(\frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1} X^i \rho_0 X^{j\dagger}\right) \otimes \left(\sum_{n=0}^{d-1}\sum_{m=0}^{d-1} P_n \rho_0 P_m^\dagger\right) \\
&\longrightarrow \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}\sum_{n=0}^{d-1}\sum_{m=0}^{d-1} X^i \rho_0 X^{j\dagger} \otimes \omega^{-in+jm} P_n \rho_0 P_m^\dagger \\
&= d\sum_{n=0}^{d-1}\sum_{m=0}^{d-1} \left(P_n \rho_0 P_m^\dagger\right) \otimes \left(P_n \rho_0 P_m^\dagger\right) \qquad (2.57)
\end{aligned}$$

and then discard (trace out) the first state to get the state $\rho = \sum_n P_n \rho_0 P_n = \sum_n \rho_{\tilde{n}}/d$, which gives us an unknown eigenstate of X as before. We then discard and repeat if we obtained the $\rho_{\tilde{0}}$ state, and otherwise we relabel the state as $\rho_{\tilde{1}}$.

Once the $\rho_{\tilde{1}}$ state is available, we can use a controlled-sum to produce the Z gate, which will allow us to produce any X ancilla including more $\rho_{\tilde{1}}$ states.

Finally, measuring X works by fusing the pair of anyons, because the $\rho_{\tilde{i}}$ are orthogonal to the vacuum for $i > 0$. The full measurement proceeds as before by copying and permuting states using the Z gate, and then fusing.

2.5.3 Leakage detection and $\rho_{\tilde{0}}$ generation

One final issue remains: How do we measure whether a state is in the computational subspace? Projecting onto the computational subspace is useful because a $\rho_{\tilde{0}}$ is just the projection of a vacuum state to the computational basis. Furthermore, this projection will allow us to perform leakage correction.

One of the new issues that arises for general non-solvable groups is that if we have a state in the computational basis, and we braid it with an electric charge carrying a non-trivial representation of the subgroup N, then the state will move outside the computational basis. The other issue is that the conjugacy class of an element in G might be larger than the conjugacy class of the element in P; though given that P is normal, the first set will be entirely contained in P.

Let us begin by examining how the leakage correction algorithm must be changed. The first step is to detect whether the net flux or charge of the pair of anyons we are working on has a non-trivial effect on the states ρ_x. The procedure is to braid the pair around the ancilla pair and then fuse the ancilla with another ancilla in the state $\rho_{x^{-1}}$. If the anyon pair has an effect on the ancilla states ρ_x, then the fusion statistics will be altered, and this will be detectable after many repetitions. If our state is found defective, we discard it as usual and replace it by a state in the computational basis. Otherwise, we move on to the next step. Note that if the anyon pair had a net flux in the subgroup N, or in some element outside of P that commutes with P, then the state will still advance to the

next round of error correction. However, this anomalous flux or charge will not affect the usual braiding properties.

The second round of error correction is a swap with an ancilla in the ρ_0 state. Note that using our universal classical computation in P, we can guarantee that if the original state was in P, then the ancilla ends up in the computational basis. However, if the original anyons are outside of P, we will get a state that is within P (because P is normal) but not necessarily in the computational subspace. The final step is to perform a swap with a second ancilla in the ρ_0 state, where now we know that the first ancilla had to be composed of anyons with no charge and fluxes only in P. This guarantees that the final state of the second ancilla is in the computational basis, and equals the original state if it did not leak, completing the leakage correction procedure.

To create $\rho_{\tilde{0}}$ we also use a swap, this time between a pair created from the vacuum and a ρ_0 ancilla. We then try to fuse the leftover vacuum state with a $\rho_{b^{-1}}$. If they fuse into the vacuum, then the ancilla is in a ρ_0 state. The logic is as follows: if the vacuum pair had electric charge when created, then the swap will not change the charge, and hence it cannot disappear into the vacuum. If the vacuum pair has no electric charge but is outside of P, then the ancilla is still guaranteed to be in P. Furthermore, when conjugating the vacuum state, we will be conjugating by an element in P. The vacuum state will end in a flux state outside of P, which is orthogonal to $\rho_{b^{-1}}$. Finally, if the vacuum pair is a pair of fluxes in P, then it will be of the form $\rho_{\tilde{0}}$, possibly superposed with other states ρ_x outside the computational basis. But the generalized swap can guarantee that a state in P outside of the computational basis will remain outside of the computational basis (just like in the simple perfect case). Only when the ancilla is in the state $\rho_{\tilde{0}}$ can the fusion into the vacuum occur.

The above procedure for producing $\rho_{\tilde{0}}$ ancillas completes the gate-set for non-solvable groups and proves the main result of this chapter: that anyons with fluxes in a non-solvable group can perform universal quantum computation.

2.6 Creating the ancillas

As discussed above, the requirement of a supply of calibrated flux ancillas needs further justification. In this section we will show that for a perfect and simple group, the requirements of braiding, fusion and vacuum pair creation can be supplemented by one extra measurement to allow the distillation of flux ancillas. We will not cover the general non-solvable case, though.

The new measurement involves determining whether a single anyon has trivial flux or not. Indeed, this measurement may even be done destructively. The plausibility of this measurement relies on the fact that non-zero flux charges are topologically non-trivial configurations that often have much higher masses than their electric charge counterparts. Naturally, dyons also have large masses and will be detected as having non-trivial flux.

Step 1: Creating electric ancilla pairs

The procedure for creating flux ancillas begins by creating single anyons with zero flux. These are obtained by creating a vacuum pair, measuring the flux of the first particle of the pair, and discarding the second one if the first one had non-trivial flux.

The next step is to create pairs of anyons, where each anyon has zero flux and unknown charge, but the pair has vacuum quantum numbers. Of course, if we could non-destructively distinguish trivial from nontrivial flux, we could skip this step, as the vacuum pairs always have vacuum quantum numbers.

Take two of the single electric charges we have produced. We are going to try to project this state onto the desired state with vacuum quantum numbers. Consider the process of creating a pair of anyons from the vacuum, braiding one of them around the pair of charges, and then fusing the vacuum pair. If the pair of charges had vacuum quantum numbers, then the vacuum pair will remain in the vacuum state throughout this process and fuse into the vacuum at the end with unit probability. On the other hand, if the pair of charges does not have vacuum quantum numbers, then there will be a finite probability that the pair created from the vacuum will leave a particle behind after fusion (since the vacuum is the only state that is left invariant by the action of every flux).

Repeated application of this process will be a projective measurement that determines whether the pair of charges has vacuum quantum numbers. If we project onto a vacuum pair, then we have found a good charge ancilla pair. If the pair does not project onto the vacuum state (because the two anyons do not transform in conjugate representations, or because we projected to a state orthogonal to the vacuum), then we pair them up with other charges and repeat the process. While slow, this process will eventually yield as many electric charge pairs with vacuum quantum numbers as needed.

Step 2: Identifying the magnetic charges

The electric ancilla pairs are useful because they can perform a non-destructive measurement of magnetic flux. The procedure is to take a member of the electric charge pair, drag it around the anyons or group of anyons whose total flux we want to measure, and then fuse it with its pair.

To describe the effect of the fluxes, let $R(g)$ be the representation of the first electric charge of the pair. Let $|n\rangle$ be an orthonormal basis for the space on which R acts, and let $|n^*\rangle$ be the dual basis for the conjugate representation R^* under which the second charge transforms. The effect of a flux g is then

$$\sum_n |n\rangle \otimes |n^*\rangle \longrightarrow \sum_n (R(g)|n\rangle) \otimes |n^*\rangle. \tag{2.58}$$

Just as before, if the total flux is non-trivial, there will be a good chance that the fusion of the

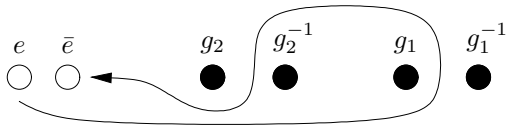


Figure 2.3: Using electric charges to check if $g_1 = g_2$.

electric charges will leave a particle behind. On the other hand, if the total flux is trivial, even if the total charge is not, the pair of electric charges will always fuse into the vacuum.

Repeated application of this procedure will determine whether the total flux is trivial or not. Furthermore, this procedure will at worst introduce decoherence in the flux basis, but will leave all flux eigenstates unchanged.

We can use this procedure to compare the fluxes of two anyons. In particular, consider two pairs created from the vacuum. Measure the total flux of the first anyon of the first pair combined with the second anyon of the second pair. If the combined flux is trivial, the first anyon of each pair has the same flux; otherwise the flux is different. The procedure is depicted in Fig. 2.3.

The above procedure allows us to sort the flux pairs into "bins" that depend on the total flux of the first anyon of the pair. We will get as many bins as elements of G, each containing an unlimited supply of vacuum pairs that carry the same flux in the first anyon of the pair. At this point, if the fluxes have not decohered in the flux basis, then we must have an entangled state involving all anyons in a given bin. Throwing away a single flux from each bin will produce the desired decoherence, just as it did when breaking the various symmetries in the main part of this chapter.

All that remains is to identify each bin with an element of G. Assume that we were given an assignment of an element of G to each bin. The assignment could be checked by using the following procedure. First, we note that any finite group G may be described by a set of elements $\{g_i\}$ and a set of relations of the form $g_{i_1} \cdots g_{i_n} = 1$, which they obey. To check that the assignment is correct, we just need to check all the relations (supplemented by the trivial one element relations $g_i^n g_i^m = g_i^{nm}$). These can be checked again with the electric charge ancillas, using a loop that circles each of the fluxes in the relation in the correct sequence.

To generate guesses, we could just randomly assign to each bin an element of g, which gives us a probability of success of at least $1/(|G|)!$. Of course, we can be a lot smarter, as the above procedure can help us figure out the powers of a given element (including its inverse) and even the elements in its conjugacy class. Thus the need for guesswork is minimal, and some of the choices correspond to different valid assignments (i.e., automorphisms) of the group.

Analysis of the produced ancillas

At this point we have almost produced the desired ancillas, with one caveat: the individual anyons do not have trivial charge (i.e., they may be dyons). However, all we have done to the pairs, after creating them from the vacuum, is to measure the flux of one of the anyons. That means that the electric charge portion of the state is still in the vacuum state. More technically, if the ancilla pair circles a flux that commutes with the flux of one anyon of the ancilla, then the state remains unaltered. This is the same behavior that the pure magnetic charges would have.

Some careful thought at this point shows that these states are good enough for the quantum computation procedure presented in the bulk of the chapter. Indeed, going back and repeating all the steps with these generalized ancillas would require very few modifications. The fusion to measure in the Z basis would now have a lower success probability, which is compensated by a higher rate of producing acceptable $|\tilde{0}\rangle$ states, but otherwise most gates remain unaltered. We have therefore succeeded in constructing an ancilla reservoir, which, while slightly different then the one initially desired, is useful for universal quantum computation.

2.7 Mathematicalia

In this section we prove the major mathematical theorems needed in the bulk of the chapter. We begin by stating the definitions of some of the mathematical terms used:

Perfect group: A non-trivial group G such that $[G, G] = G$. Note that $[G, G]$ is not the set of elements of the form $[g_1, g_2] \equiv g_1 g_2 g_1^{-1} g_2^{-1}$ but rather the group generated by these elements. Even if G is perfect, there may not be a commutator expression for every element.

Non-solvable group: A group that has a perfect subgroup.

Normal subgroup: A subgroup H of a group G such that $ghg^{-1} \in H$ for every $h \in H$ and $g \in G$.

Simple group: A group with no normal subgroups other than the whole group and the trivial group.

Before we get to our main theorem, we will prove a theorem that will allow us to deal with general non-solvable groups. We intend to show that we can extract from non-solvable groups a simple and perfect group. The simple perfect groups (which can also be described as the simple non-abelian groups) are the ones on which we can perform universal classical computation and are therefore important for this chapter.

We begin by defining the n^{th} derived subgroups by the relations $G^{(n)} = [G^{(n-1)}, G^{(n-1)}]$ and $G^{(1)} = [G, G]$. A solvable group is one for which $G^{(i)} = \{1\}$ for some i. A non-solvable group must have an i such that for every $j > i$, $G^{(i)} = G^{(j)}$ and $G^{(i)}$ is non-trivial. The group $G^{(i)}$ is perfect, thus the definition for solvable groups is consistent with the definition for non-solvable groups given above.

Furthermore, all the groups $G^{(n)}$ are normal subgroups of G. This can be proven by recalling the

property $g[g_1, g_2]g^{-1} = [gg_1g^{-1}, gg_2g^{-1}]$. The rest follows by induction because $G^{(1)}$ is normal in G, and $G^{(i)}$ is normal in G if $G^{(i-1)}$ is normal in G. We have therefore shown that every non-solvable group G has a perfect normal subgroup P.

Sadly, this subgroup is not necessarily simple. However, we can prove that every perfect group P has a normal subgroup N such that P/N is perfect and simple. We choose N to be a normal proper subgroup of P such that no other normal proper subgroup of P has more elements, which is well defined because P is finite. Let f be the canonical epimorphism $P \to P/N$, which maps elements into cosets. Because f is surjective we have $[P/N, P/N] = [f(P), f(P)] = f([P, P]) = f(P) = P/N$, which, combined with the fact that P/N is non-trivial, shows that P/N is perfect.

Finally, assume that P/N has a normal, non-trivial proper subgroup A. Then $B = f^{-1}(A)$ is a normal subgroup of P, because for any elements $b_1, b_2 \in B$ and $p \in P$, we have $f(b_1b_2) = f(b_1)f(b_2) \in A$ and $f(pb_1p^{-1}) = f(p)f(b_1)f(p)^{-1} \in A$. Furthermore, B is a proper subgroup of P, and $N = f^{-1}(1)$ is smaller than $B = f^{-1}(A)$, leading to a contradiction. Therefore, P/N is simple, and we have finished proving the following theorem:

Theorem 3. *If G is a non-solvable finite group, then there exists a normal subgroup P of G and a subgroup N, normal in P, such that P/N is perfect and simple.*

We now turn our attention to using our groups to compute classical functions. We shall prove that the set of functions that can be written in product form is complete, in the sense that it includes every function from $G^n \to G$, if G is simple and perfect (or equivalently simple and non-abelian). This was first proven in the mathematical literature by Maurer in 1965 [MR65]. In the computer science literature, a related result was proven by Barrington [Bar89]. In this thesis, we will provide our own constructive proof for the following theorem:

Theorem 4. *If G is a simple and perfect finite group, then any function $f(g_1, \ldots, g_n) : G^n \to G$ can be expressed as a product of the inputs $\{g_i\}$, their inverses $\{g_i^{-1}\}$ and fixed elements of G, any of which may appear multiple times in the product.*

Proof. Throughout this proof we will refer to the set of functions that can be expressed in the above form as "computable." Proving the above statement is equivalent to showing that all functions are computable. The proof consists of building a series of computable delta functions that map most elements to the identity and then expressing arbitrary functions as a product of these delta functions.

Step 1: Given a group element a not equal to the identity, let $C(a)$ denote its conjugacy class. Then the subgroup generated by the elements of $C(a)$ is equal to G. This is because the subgroup is a nontrivial, normal subgroup of G and G is simple.

Step 2: Fix two disjoint subsets A and B of G. Define a family of functions $\{\delta_c^{A,B}(g) : A \cup B \to G\}$

with elements labeled by $c \in G$:

$$\begin{aligned} \delta_c^{A,B}(a) &= 1 \qquad \forall\ a \in A \\ \delta_c^{A,B}(b) &= c \qquad \forall\ b \in B. \end{aligned} \tag{2.59}$$

If the function $\delta_c^{A,B}$ is computable for some $c \neq 1$, then every function in the family is computable. To prove this choose any $d \in G$. By Step 1 there is an expression for d as a product of elements in the conjugacy class of c (for instance, $d = g_1 c g_1^{-1} g_2 c g_2^{-1} c$). Then $\delta_d^{A,B}$ is obtained by substituting $\delta_c^{A,B}$ for c in the expression.

Step 3: Fix a set A, an element b not in A, and an element $x \neq b$. If a function $\delta_c^{A,B}$ is computable for some B such that $b \in B$, then there exists a computable function $\delta_c^{A',B'}$ with two new sets such that $A \cup \{x\} \subset A'$ and $b \in B'$. The function can be obtained from

$$\delta_e^{A',B'}(g) = [\delta_d^{A,B}(g), gx^{-1}] \tag{2.60}$$

using Step 2. The above equation assumes that we have extended the domain of $\delta_d^{A,B}$ to G, which can be done in a natural way once we have fixed a product representation for $\delta_d^{A,B}$. The element d was chosen to not commute with bx^{-1}. Such an element must exist because G is simple and non-abelian, and hence has no center. The element e is just $e = [d, bx^{-1}]$.

Step 4: The functions defined by $\delta_c^b(g) \equiv \delta_c^{A,B}$, with $A = G - \{b\}$ and $B = \{b\}$, are computable. To prove this start with $A_1 = \{1\}$ and $B_1 = \{b\}$. The function $\delta_c^{A_1,B_1}$ is computable because it is in the same family as $f(g) = g = \delta_g^{A_1,B_1}$. Then proceed by induction, using Step 3, on the elements in $G - \{b\}$ that are not included in A_i.

Step 5: For a fixed set of ordered elements $b_1, \ldots, b_i$ define a family of functions labeled by c:

$$\begin{aligned} \delta_c^{b_1 \ldots b_i}(g_1, \ldots, g_i) &= c \qquad g_1 = b_1, \ldots, \text{and } g_i = b_i \\ \delta_c^{b_1 \ldots b_i}(g_1, \ldots, g_i) &= 1 \qquad \text{otherwise.} \end{aligned} \tag{2.61}$$

The same proof in Step 2 shows that if any function of the family with $c \neq 1$ is computable, then the entire family is computable.

Step 6: Fix $i \in Z^+$ and elements $b_1, \ldots, b_{i+1} \in G$. If the function $\delta_c^{b_1 \ldots b_i}(g_1, \ldots, g_i)$ is computable, then so is the function $\delta_c^{b_1 \ldots b_{i+1}}(g_1, \ldots, g_i, g_{i+1})$. By Step 5 it is sufficient to be able to compute

$$\delta_e^{b_1 \ldots b_{i+1}}(g_1, \ldots, g_{i+1}) = \left[\delta_c^{b_1 \ldots b_i}(g_1, \ldots, g_i), \delta_d^{b_{i+1}}(g_{i+1})\right], \tag{2.62}$$

where the function $\delta_d^{b_{i+1}}(g_{i+1})$ is computable by Step 4, and d is chosen so that $e = [c, d] \neq 1$.

Step 7: Using induction on the number of inputs of the function, and starting from the base case

$\delta_c^{b_1}(g_1)$, it is clear that all the functions defined in Step 5 are computable.

Step 8: Every function is computable because

$$f(g_1,\ldots,g_i)=\prod_{b_1\in G}\cdots\prod_{b_i\in G}\delta^{b_1\ldots b_i}_{f(b_1,\ldots,b_i)}(g_1,\ldots,g_i)\,. \tag{2.63}$$

□

Chapter 3

Anyon computers with smaller groups

The previous chapter has shown that finite non-solvable groups produce anyons capable of universal quantum computation. However, the smallest finite non-solvable group is A_5, the even permutations of five objects, which has 60 elements. Unfortunately, anyons with a large symmetry group are less likely to be found in nature and are also harder to engineer. A more desirable symmetry group would be S_3, with only 6 elements. The purpose of this chapter is to study the feasibility of quantum computation with these smaller groups. In fact, it will be shown that the groups that are solvable but not nilpotent, which includes S_3 as the smallest case, produce anyons capable of universal quantum computation. The caveat, though, is that the constructions in this chapter require both electric and magnetic charges, whereas magnetic charges alone were sufficient in the non-solvable case. The use of electric charges complicates the procedure significantly and will occupy the bulk of the discussion.

This chapter will also elucidate the relationship between symmetry group structure and anyon computation power, and some of the connections with the theory of classical automata will be briefly discussed. The ideas of this chapter are based on an unpublished construction by Kitaev for the group S_3 [Kit02], and in particular his use of electric charges to obtain a magic state capable of completing the universal gate-set.

The organization of this chapter is as follows: Sec. 3.1 introduces some notation beyond what was used in the previous chapter. The next two sections prove the universality of anyons based on groups that are semidirect products of certain cyclic groups of prime order, which includes the important case of S_3. Sec. 3.2 constructs an abstract set of gates out of the fundamental anyon operations, whereas Sec. 3.3 proves that this gate-set is universal. In Sec. 3.4, the discussion is expanded to general finite groups, and the relationship between group structure and computational power is established. This section will also review the definitions of solvability and nilpotency. The main result of this chapter, which is the feasibility of universal quantum computation with anyons from groups that are solvable but non-nilpotent, is proven in Sec. 3.5. The discussion in Sec. 3.5 is

motivated by Sec. 3.2 and includes many of the same steps, but the details are significantly more complicated. Finally, Sec. 3.6 discusses a leakage correction scheme that can be applied to anyons, as well as many other quantum systems.

3.1 Review and notation

Most of the notation introduced in the previous chapter will be reused. We will also introduce some extra notation in order to deal with the electric charges and the special type of gate that they will produce.

In particular, we shall retain the notation for magnetic flux pairs, where the state $|g\rangle$, referred to as a state of flux g, will denote a pair of anyons one with flux g and one with flux g^{-1} (and hence trivial total flux). We will also extensively employ the "conjugations by a function" developed in the last chapter. We also retain the notation for qudits from the last chapter and leave implicit that all operations are to be done modulo d where appropriate.

3.1.1 Electric charge pairs

Recall that electric charges correspond to irreducible representations R of the group G and contain an internal state that transforms as a vector under R. In addition to pairs of magnetic charges, this chapter will often deal with pairs of electric charges, where the first charge transforms under the irreducible representation R, and the second charge transforms under the complex conjugate representation R^*. Of course, for some representations $R^* \simeq R$, which will not be a problem for what follows.

We introduce the bases $\{|i_R\rangle\}$ and $\{|j_{R^*}\rangle\}$ on which the representations act. The indices i, j take values from 1 to d_R, the dimension of the representation. We assume that the basis vectors are compatible in the sense that

$$\langle i_{R^*}|R^*(g)|j_{R^*}\rangle = \langle i_R|R(g)|j_R\rangle^*. \tag{3.1}$$

The combined state of the two charges is spanned by the vectors $|i_R\rangle \otimes |j_{R^*}\rangle$ and can be described by specifying a $d \times d$ matrix M

$$|M\rangle_R \equiv \frac{1}{\sqrt{d_R}} \sum_{i,j} M_{ij}|i_R\rangle \otimes |j_{R^*}\rangle, \tag{3.2}$$

where we have introduced a convenient normalization factor.

We will be interested in the braiding and fusion properties of these states. However, when two electric charges move past each other, even when they are not in pairs, their charges remain

unchanged. It is only the magnetic fluxes that have an effect on the electric charges. In particular, when a magnetic flux g goes around an electric charge, the flux remains invariant, but the charge transforms as if multiplied by g in the representation R. Starting with a state $|M\rangle_R$, if the flux circles the first electric charge, then it becomes

$$U(g) \otimes I|M\rangle_R = \frac{1}{\sqrt{d_R}} \sum_{i,j,k} R_{ik}(g) M_{kj} |i_R\rangle \otimes |j_{R^*}\rangle = |R(g)M\rangle_R \tag{3.3}$$

where $R(g)M$ is the matrix obtained by left multiplying M by the element g in the representation R. Similarly, if we act on the second charge, we obtain

$$\begin{aligned} I \otimes U(g)|M\rangle_R &= \frac{1}{\sqrt{d_R}} \sum_{i,j,k} M_{ik} R^*_{jk}(g) |i_R\rangle \otimes |j_{R^*}\rangle \\ &= \frac{1}{\sqrt{d_R}} \sum_{i,j,k} M_{ik} R^\dagger_{kj}(g) |i_R\rangle \otimes |j_{R^*}\rangle = |MR(g^{-1})\rangle_R \end{aligned} \tag{3.4}$$

where we have used the fact that R is unitary.

Note that, just as in the case of the magnetic charges, if we have a function $f(\{g_i\})$ of some anyon fluxes, written out in product form, then we can apply this function to our charges

$$|M\rangle_R \longrightarrow U(f) \otimes I|M\rangle_R = |R(f)\, M\rangle_R \tag{3.5}$$

by applying sequentially from right to left the elements of the product.

3.1.2 Superselection sectors, fusion and vacuum pairs

The rules for fusion and vacuum pair creation of magnetic charges (and their decomposition into superselection sectors) is the same as in the prior chapter. In the case of electric charges, the vacuum state for representation R is simply $|R(I)\rangle_R$, where $R(I)$ is the $d_R \times d_R$ identity matrix.

The fusion of two electric charges can only produce another electric charge (or the vacuum, which is the charge carrying the trivial representation). To calculate the possible products of fusion, note that fusion implies that a flux can no longer be braided around only one of the two electric charges. Mathematically, it is a restriction to the diagonal transformations

$$|M\rangle_R \longrightarrow U(g) \otimes U(g)|M\rangle_R = |R(g)MR(g^{-1})\rangle_R. \tag{3.6}$$

However, the above action of the group is not irreducible on this space. The vector space spanned by all possible states $|M\rangle_R$ decomposes into invariant subspaces. The invariant subspaces correspond to electric charges transforming under irreducible representations. The probability of obtaining each irreducible representation corresponds to the magnitude of the state vector projected down to the

appropriate invariant subspace. Furthermore, after fusion, it is no longer possible to measure the relative phase between the different representations and therefore decoherence effectively occurs in the representation basis.

The net result of fusion is a mixed state of different representations. Which representations occur is determined by the decomposition of $R(g) \otimes R^*(g)$ into irreducible representations. The probability of obtaining each of these representations is determined by the projection of M to the different invariant subspaces.

In particular, the trace of M is the unique invariant under conjugation by G (which is the content of Schur's lemma). Therefore the probability of fusion into the vacuum is

$$P_{\text{vac}} = |\langle R(1)|M\rangle_R|^2 = \left|\frac{\operatorname{Tr} M}{d_R}\right|^2 . \tag{3.7}$$

3.1.3 Requirements for the physical system

We list here the operations, ancillas and measurements that we assume are available on any realistic system, and which we will use to build our quantum gate-set:

1. We can braid or exchange any two particles.

2. We can fuse a pair of anyons and detect whether there is a particle left behind or whether they had vacuum quantum numbers.

3. We can produce a pair of anyons in a state that is chosen at random from the two particle subspace that has vacuum quantum numbers.

4. We have a supply of ancillas of the form $|g\rangle$ for any $g \in G$.

5. We have a supply of ancillas of the form $|R(I)\rangle_R$ for any irreducible unitary representation R.

The last two requirements are the only questionable ones, as it is not obvious how to produce this reservoir of calibrated electric and magnetic charges. In fact, since many of these ancillas will be destroyed during fusion, the reservoir will have to have a large number of ancillas of each type.

Note that the main difference between the constructions in this chapter, and the one used in producing computations with non-solvable groups in the last chapter, is that the latter case required no electric charge ancillas, which may be harder to produce. The production of calibrated flux and charge ancillas for the groups discussed in the present chapter, though similar to the case discussed in the previous chapter, will not be addressed here.

A final note is that the requirement of calibrated magnetic charge ancillas will have to be slightly modified in Sec. 3.5.3, in order to work with certain large groups.

3.1.4 Probabilistic projection onto $\mathcal{K}$

To conclude with the introduction of notation, we define a new type of gate called a probabilistic projection onto a subspace. The operation is essentially a projective measurement that distinguishes between a subspace $\mathcal{K}$ and its orthogonal complement. However, the operation has a one-sided probability of error, corresponding to a failure to notice the projection into $\mathcal{K}$.

For example, consider an operation that emits a photon if and only if the state is projected into the subspace $\mathcal{K}$. The photon is then received at a photodetector that has a probability $0 < p \leq 1$ of absorbing the photon. A photon will never be detected if the state was projected into the complement of $\mathcal{K}$; but even if the measurement projected into $\mathcal{K}$, the photodetector may remain silent.

To formalize the idea of a probabilistic projection, let $\mathcal{K}$ be a subspace of a Hilbert space $\mathcal{H}$, and let $P_{\mathcal{K}}$ be the projection onto $\mathcal{K}$. We define a probabilistic projection onto $\mathcal{K}$ as a two-outcome POVM with operators

$$F_0 = p_{PP} P_{\mathcal{K}}, \qquad F_1 = 1 - p_{PP} P_{\mathcal{K}}, \tag{3.8}$$

where $0 < p_{PP} \leq 1$. We say that we can do a probabilistic projection onto $\mathcal{K}$ if we can do the above operation for any fixed p_{PP}.

Furthermore, we demand that if outcome 0 is obtained when applying the operation to a state $|\Psi\rangle$, we obtain the state

$$|\Psi_0\rangle = \frac{P_{\mathcal{K}}|\Psi\rangle}{\sqrt{\langle\Psi|P_{\mathcal{K}}|\Psi\rangle}}. \tag{3.9}$$

On the other hand, if we get the result 1, we will consider the state damaged and trace it out of our computational system.

As an example consider

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle \right), \tag{3.10}$$

and let $\mathcal{K} = \{|0\rangle\}$. Applying a probabilistic projection to the first qubit, we obtain with probability $p_{PP}/2$ the state

$$|\Psi_0\rangle = |0\rangle \otimes |1\rangle, \tag{3.11}$$

and with probability $1 - p_{PP}/2$ we obtain the mixed state

$$\rho_1 = \frac{1}{2 - p_{PP}} \Big[(1 - p_{PP}) \, |1\rangle\langle 1| + |0\rangle\langle 0| \Big], \tag{3.12}$$

where we have already traced out the first qubit. Notice that if the probabilistic projection onto $|0\rangle$ is applied to both qubits simultaneously, it is possible to obtain the result 1 twice, but it is not possible to obtain the result 0 twice.

3.2 Base case: $G = \mathbb{Z}_p \times_\theta \mathbb{Z}_q$

Before tackling the general case of groups that are solvable but not nilpotent, we will describe the procedure for producing quantum computation using a special type of group based on the semidirect product. The construction for these groups is very similar to the general case, but can be described in more concrete terms. In particular, these groups are very useful in eliminating operations whose usefulness is unclear in the general case, but that have no computational power when reduced to this special case.

3.2.1 Algebraic structure

We will be interested in the groups $G = \mathbb{Z}_p \times_\theta \mathbb{Z}_q$, the semidirect product of the cyclic groups of order p and q. We assume that $p \neq q$ are both prime and that the function θ is non-trivial, which guarantees that G is not nilpotent.

The group can be described using two generators a and b, which satisfy the relations:

$$a^p = 1, \quad b^q = 1, \quad bab^{-1} = a^t, \tag{3.13}$$

where specifying an integer t between 0 and p is equivalent to specifying the function $\theta : \mathbb{Z}_q \to \mathrm{Aut}(\mathbb{Z}_p)$ used for the semidirect product. We will require that $t \neq 1$, which is equivalent to θ being non-trivial. Furthermore, consistency requires that

$$a = b^q a b^{-q} = a^{t^q} \implies t^q = 1 \bmod p, \tag{3.14}$$

which can always be solved for some t as long as q divides $p - 1$. We henceforth assume that p, q and t have been chosen in a self-consistent fashion.

The best example of one of these groups, and in fact the smallest non-abelian group, is S_3. This group can be expressed as $\mathbb{Z}_3 \times_\theta \mathbb{Z}_2$, with $t = 2$. We can choose a to be any order three element such as (123), and we can choose b to be any order two element such as (12).

The first example of such a group with odd order is $\mathbb{Z}_7 \times_\theta \mathbb{Z}_3$ with $t = 2$ or $t = 4$, both of which are equivalent. One of the most important features of this example is that not all the non-trivial powers of a are conjugate to one another. The elements a, a^2 and a^4 form one conjugacy class, whereas the elements a^3, a^5 and a^6 form another.

Both of the above examples will be revisited when we discuss group representations and fusion of electric charges.

3.2.2 Computational basis

We choose a qudit computational basis

$$|i\rangle = |a^i b a^{-i}\rangle, \tag{3.15}$$

for $0 \leq i < p$. Note that all these states are unique because $a^i b a^{-i} = a^{i(1-t)}b$, and a^{1-t} is a non-trivial generator of the group $\mathbb{Z}_p$. We are therefore using a complete conjugacy class for the computational subspace.

While the above choice of computational subspace may seem arbitrary, most other choices are either equivalent or less powerful. The conjugacy classes $a^i b^j a^{-i}$, for different non-trivial values of j, are all equivalent. Dyons with these fluxes are also equivalent since they are just the combination of the above states with electric charges that cannot be detected by braiding. Finally, the powers of a and pure electric charges are suboptimal as they are difficult to entangle (for more on this see the discussion on using nilpotent groups in Sec. 3.4).

Initializing a quantum computer in this basis is easy, as we have assumed the existence of flux ancillas in the state $|0\rangle$, which can be used as computational anyons. We therefore turn to the task of implementing gates on this space.

3.2.3 Operations involving braiding fluxes

We begin by characterizing the operations that can be achieved by braiding fluxes. Fix a target qudit that we will be conjugating, and assume that it is in the computational subspace. We can conjugate this qudit by the fluxes of arbitrary ancillas in the group. It can also be conjugated by the fluxes of other qudits, which we will also assume to have a definite flux in the computational subspace (as the effect of a superposition of fluxes can be inferred by linearity).

Let us begin with the case when only one qudit (in addition to the target) is involved. If the source qudit is in a state $|g\rangle$, then the target will get conjugated by an expression

$$f(g) = c_1 g c_2 g c_3 \dots c_n, \tag{3.16}$$

for some n, where the $\{c_j\}$ are fixed elements of G corresponding to the ancillas used. Of course, these elements represent the product of any ancillas that were used in series and can equal the identity if no ancillas were used.

Because of the structure of the group, all the fixed elements can be expanded as $c_i = a^{j_i} b^{k_i}$ for

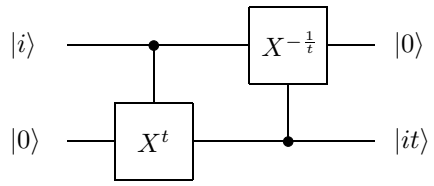
some integers j_i, k_i. Furthermore, since the source flux is in the computational basis, it can be written out as $g = a^x b a^{-x} = a^{x(1-t)} b$, for some x. Inserting these expressions, we get

$$f(g) = a^{j_1} b^{k_1} a^{x(1-t)} b a^{j_2} b^{k_2} \dots a^{j_n} b^{k_n}. \tag{3.17}$$

Using the group relation $ba^i = a^{it}b$, we can move all the b's to the right and combine factors to get

$$f(g) = a^{\alpha} a^{\beta x} b^{\delta}, \tag{3.18}$$

for some integers α, β and δ. The effect of each of these factors can be considered separately. Conjugating by a^{α} is just the application of the gate X^{α}. Conjugating by $a^{\beta x}$ is just a controlled-X from the source to the target, repeated β times. Finally, conjugating by b maps $|i\rangle$ to $|it\rangle$. This operation can be generated using a controlled-X gate and an ancilla $|0\rangle$:



where $-1/t$ is computed modulo p. Following the above circuit, we can either replace the original qudit with the ancilla, or use a swap, which can also be built out of controlled-X gates.

So far we have shown that the X and controlled-X gates generate the set of operations achieved by conjugations. However, we have yet to show that these operations are in fact included in the set of achievable operations. The X gate is rather trivial as it is a conjugation by an ancilla of flux a. The controlled-X is a conjugation by the function

$$f(g) = \left(g b^{-1}\right)^{1/(1-t) \bmod p} = \left(a^{x(1-t)} b\, b^{-1}\right)^{1/(1-t) \bmod p} = a^x, \tag{3.19}$$

where $1/(1-t)$ can be computed modulo p because we assumed $1 < t < p$.

The case involving many source qudits, all of which can be used to conjugate the target, is very similar to the above. The expression can be simplified by moving all the b's to the left and combining similar factors. In the end, the net effect will again be a series of X and controlled-X gates.

Finally, one may wonder about using an ancilla as an intermediate step. That is, first we take an ancilla (say, g'), conjugate it by some function (say, f) of some qudits, and then conjugate the target by the ancilla. However, the same effect can be achieved by conjugating the target first by f^{-1}, then by g' and finally by f. This procedure therefore provides no extra computational power.

The conclusion is that the operations achievable from braiding magnetic charges are exactly those generated by the X and controlled-X gates. In fact, the X gate is redundant as we have assumed the existence of $|1\rangle$ ancillas, which can be used as control qudits in a controlled-X.

3.2.4 Operations involving fusion of fluxes

Now we turn to the operations achieved by the fusion of magnetic fluxes. For these operations it will be sufficient to determine whether the two particles fused into the vacuum or not, thereby obtaining at most one bit of information from each fusion.

At this point we remind the reader that standard states consist of pairs of anyons, whose total flux is trivial. That is, the state $|g\rangle$ describes an anyon of flux g paired with an anyon of flux g^{-1}. There are therefore two basic choices for fusion: we can fuse the two anyons that compose a single pair with each other, or we can fuse one of them with an anyon from another pair, typically an ancilla. To avoid confusion, in the latter case we will always use the anyon of flux g (rather than g^{-1}) for the fusion.

The case of fusion with an ancilla will lead to a measurement in the Z basis. The fusion of anyons from the same pair will lead to a measurement in the X basis. However, we will delay the construction of the actual measurement gates until the next section. For this section, we will simply describe the fusions as abstract operations on the computational space by employing the construction of probabilistic projections.

The fusion of an anyon from a state $|\Psi\rangle$ with an anyon ancilla of flux b^{-1} is a probabilistic projection onto the subspace $\mathcal{K} = \{|0\rangle\}$. That is, an anyon of flux $a^i b a^{-i}$ can only fuse into the vacuum with a flux b^{-1} if $i = 0$ (modulo p as usual). When $i > 0$ there must be an anyon left over to carry the non-trivial total flux. When $i = 0$ the fusion can either produce the vacuum state or an anyon with non-trivial charge. The probability for fusion into the vacuum in this case is $1/p$. Furthermore, if we fuse into the vacuum we can replace the state with a $|0\rangle$ ancilla. Therefore the whole operation is a probabilistic projection onto $|0\rangle$ with $p_{PP} = 1/p$.

The fusion of two anyons from the same pair is a probabilistic projection onto the subspace $\mathcal{K} = \{|\tilde{0}\rangle\}$. Because the total magnetic flux of the pair is always trivial, the fusion product must be an electric charge. The charge corresponds to a representation of G given by the action of conjugation on the anyon fluxes. The state $|\tilde{0}\rangle$ transforms trivially and corresponds to the vacuum, whereas the states $|\tilde{i}\rangle$, for $i > 0$, are orthogonal to the vacuum and correspond to non-trivial representations. In fact, this procedure is a probabilistic projection with $p_{PP} = 1$. However, since the state is destroyed during fusion, to complete the projection we must be able to produce $|\tilde{0}\rangle$ states. This will be discussed below.

The other choices for fusion are equivalent to a combination of one of the above measurements and an X or controlled-X gate. Fusing with a flux of the form $a^i b^{-1} a^{-i}$ is equivalent to first applying

a X^{-i} gate and then performing a fusion with b^{-1}. A fusion with any other flux can never produce the vacuum if the qudit is in the computational subspace. Finally, one can consider fusion of anyons from two different qudits. If the state of the two qudits is $|i\rangle \otimes |j\rangle$, the fusion will only produce the vacuum state if $i = j$. Therefore, the operation can be simulated by a controlled-X^{-1}, followed by the fusion of the target with a b^{-1} flux.

The conclusion so far is that fusion of magnetic charges provides us with two new operations: the probabilistic projections onto the subspaces $|0\rangle$ and $|\tilde{0}\rangle$, which will eventually become measurements in the Z and X bases. The only operation that has not been considered is using the products of fusion for further operations or fusions. This subject will be briefly touched upon after discussing fusion of electric charges.

3.2.4.1 Production of $|\tilde{0}\rangle$ states

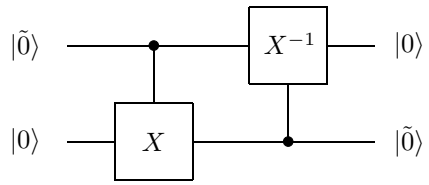
To conclude the discussion on fusion of fluxes, we present the construction of $|\tilde{0}\rangle$ states, which were needed to complete the probabilistic projection onto $|\tilde{0}\rangle$.

Just as the state $|\tilde{0}\rangle$ naturally fuses into the vacuum, it is also naturally produced from the vacuum. Unfortunately, producing a pair of anyons from the vacuum is just as likely to produce the vacuum state for one of the other superselection sectors as it is to produce the state $|\tilde{0}\rangle$. Therefore, after producing a vacuum state we must measure its superselection sector. Vacuum pairs that are produced in the computation subspace (magnetic charge in the conjugacy class of b) will be kept as $|\tilde{0}\rangle$ states, and the rest will be tossed out.

Since measurements are done by fusion, which is a destructive procedure, we must copy the vacuum state before measuring the conjugacy class. The procedure starts with a pair created from the vacuum and a $|0\rangle$ ancilla:

$$|\text{Vac}\rangle \otimes |0\rangle \tag{3.20}$$

and applies to it a swap, made out of the conjugation-based controlled-X:



where the circuit depicts the result for the case when the vacuum pair was created in the computational superselection sector, in which case $|\text{Vac}\rangle = |\tilde{0}\rangle$.

In the case when the vacuum state was not created in the computational superselection sector, the effect of the conjugations will be different. However, since the conjugations are performed using

braiding, which never changes the superselection sector, the vacuum state can only be transformed into a state that is orthogonal to $|0\rangle = |b\rangle$.

After applying the above controlled-X gates, we attempt to fuse an anyon from what was the vacuum state with an ancilla of flux b^{-1}. If they fuse into the vacuum, this implies that the vacuum state was created in the computational superselection sector, and the above circuit worked correctly. The ancillas $|0\rangle$ will have been transformed properly into a $|\tilde{0}\rangle$ ancilla, which can be used for computation. In the case when the fusion does not produce a vacuum state, the swap probably did not produce the desired state, so we discard it and start over.

To summarize, we now have a source of $|\tilde{0}\rangle$ ancillas, which can be used as the last step needed to complete the probabilistic projection onto $|\tilde{0}\rangle$.

3.2.5 Representations and fusion of electric charges

Thus far, we have only considered operations involving magnetic fluxes. These operations led to a controlled-X and measurements in the X and Z bases. However, these gates do not form a universal gate-set. We must therefore consider operations involving electric charges as well.

The electric charges transform as irreducible representation of the group G. To obtain the spectrum of electric charges, as well as their braiding and fusion rules, we must therefore discuss the representation theory of G.

It is easy to see that the commutator subgroup G' of groups of the form $G = \mathbb{Z}_p \times_\theta \mathbb{Z}_q$ is just $G' = \mathbb{Z}_p$. The representation theory of G can be obtained by inducing representations from G'. Starting from the trivial representation on G', the induced representations are the one-dimensional representations where $a \to 1$ and b is a q^{th} root of unity.

The rest of the irreducible representations have dimension q and are obtained by inducing from the non-trivial representations of $\mathbb{Z}_p$. The induced representations are all irreducible though not necessarily distinct. In fact, they can be easily described in their natural basis as

$$a \to \begin{pmatrix} \omega & & & \\ & \omega^t & & \\ & & \ddots & \\ & & & \omega^{t^{q-1}} \end{pmatrix}, \quad b \to \begin{pmatrix} 0 & 1 & & \\ & 0 & & \\ & & \ddots & 1 \\ 1 & & & 0 \end{pmatrix},$$

where ω is the p^{th} root of unity of the representation from which we are inducing. The matrix for a is diagonal, whereas b is the permutation matrix with entries 1 above the diagonal.

Even though the representation theory for these particular groups is easy, we will use abstract language to describe the fusion rules, which will make the connection to the general case clearer.

Take any non-abelian irreducible representation, and consider a pair of electric charges in the

state $|R(a^i)\rangle_R$. What representations do we get if we fuse the two charges? The product of fusion is invariant under the action of a:

$$U(a) \otimes U(a)|R(a^i)\rangle_R = |R(a)R(a^i)R(a^{-1})\rangle_R = |R(a^i)\rangle_R \tag{3.21}$$

and therefore represents the commutator subgroup G' by the identity. This implies that the representation is abelian! In particular, it is easy to see that the one-dimensional subspaces

$$|[\gamma^j]\rangle_R \equiv |\text{diag}\left(\gamma^j, \gamma^{2j}, \ldots, \gamma^{qj}\right)\rangle_R \tag{3.22}$$

with $\gamma^q = 1$ are the spaces corresponding to the representations $a \to 1$, $b \to \gamma^j$.

We will be interested in the quantum amplitude that a state $|R(a^i)\rangle_R$ fuses into the $b \to \gamma^j$ representation. This quantity will be denoted by the fusion amplitude

$$F_{i\to j} \equiv \left\langle \left[\gamma^j\right] \middle| R(a^i)\right\rangle_R = \frac{1}{q}\sum_{k=1}^{q} \gamma^{-kj}\omega^{it^{(k-1)}}, \tag{3.23}$$

with $0 \le i < p$ and $0 \le j < q$.

Let $|\Psi\rangle$ be an arbitrary state entangled with an electric charge pair

$$|\Psi\rangle = \sum_{i=0}^{p-1} |\Psi_i\rangle \otimes |R(a^i)\rangle_R \tag{3.24}$$

where the $|\Psi_i\rangle$ denotes (unnormalized) states of the rest of the system. The fusion amplitudes allow $|\Psi\rangle$ to be rewritten as

$$|\Psi\rangle = \sum_{i=0}^{p-1}\sum_{j=0}^{q-1} F_{i\to j}|\Psi_i\rangle \otimes |\left[\gamma^j\right]\rangle_R. \tag{3.25}$$

The basis $|[\gamma^j]\rangle_R$ labels the total charge of the two anyons that comprise the electric charge pair. A fusion of the two electric charges, followed by a measurement of the resulting fusion product, will be a measurement in this basis.

Note that the basis $|[\gamma^j]\rangle_R$ only spans the diagonal subspace of $|M\rangle_R$. However, this is the subspace containing all the states $|R(a^i)\rangle_R$. The subspaces spanned by $|R(b^j a^i)\rangle_R$, for some fixed $j > 0$, are mapped unchanged into the space of a single higher-dimensional irreducible representation and are therefore not useful for our purposes.

While the representation R does not appear explicitly in the fusion coefficients, it enters implicitly in the above expression as the choice for p^{th} root of unity ω. Though we could use the notation ω_R, this will not be necessary as we will generally work with only one higher-dimensional irreducible

representation.

The most important feature of the $F_{i\to j}$ coefficients is that $|R(a^0)\rangle_R$ is the vacuum state and therefore

$$F_{0\to j} = \delta_{j,0}, \tag{3.26}$$

which can be verified by direct calculation. Another important property is that

$$|F_{i\to j}| > 0 \tag{3.27}$$

for all $i > 0$. The proof involves showing that a linear relation of roots of unity only vanishes if it is a combination of the obvious regular polygon relations (which is proven in [Sch64]).

A final interesting property is that

$$F_{it^k\to j} = \gamma^{-jk} F_{i\to j}, \tag{3.28}$$

which is a consequence of

$$|R(a^{it^k})\rangle_R = |R(b^k a^i b^{-k})\rangle_R = U(b^k)\otimes U(b^k)|R(a^i)\rangle_R. \tag{3.29}$$

3.2.6 Examples

3.2.6.1 S_3

The group S_3 has three irreducible representations, the trivial (identity) representation (where $a\to 1$, $b\to 1$), the sign of the permutation (where $a\to 1$, $b\to -1$) and a two-dimensional one:

$$a\to \begin{pmatrix}\omega & 0\\ 0 & \bar{\omega}\end{pmatrix}, \quad b\to \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \tag{3.30}$$

where ω is a non-trivial cube root of unity. The fusion amplitudes are

$$\begin{aligned} F_{0\to 0} &= 1, \quad F_{1\to 0} = -\frac{1}{2}, \quad F_{2\to 0} = -\frac{1}{2},\\ F_{0\to 1} &= 0, \quad F_{1\to 1} = -i\frac{\sqrt{3}}{2}, \quad F_{2\to 1} = i\frac{\sqrt{3}}{2}. \end{aligned} \tag{3.31}$$

The best way to visualize these coefficients is to start with a state $|\tilde{0}\rangle$ and a pair of electric charges in the vacuum state of the two-dimensional representation: $|R(I)\rangle_R$. Then entangle with a

controlled-sum to get

$$\frac{1}{\sqrt{3}}\sum_j |j\rangle|R(a^j)\rangle_R = \frac{1}{\sqrt{3}}\sum_j |j\rangle \left| \begin{pmatrix} \omega^j & 0 \\ 0 & \bar{\omega}^j \end{pmatrix} \right\rangle_R . \tag{3.32}$$

Fusion of the electric charge pair produces either the vacuum (trivial representation) or a charge transforming under the sign representation. The probability of getting each is

$$\begin{aligned} P_{vac} &= \sum_j \left| \frac{1}{\sqrt{3}} F_{j\to 0} \right|^2 = \frac{1}{2}, \\ P_{sgn} &= \sum_j \left| \frac{1}{\sqrt{3}} F_{j\to 1} \right|^2 = \frac{1}{2}, \end{aligned} \tag{3.33}$$

and the state of the magnetic charges afterwards is one of

$$\begin{aligned} |\Psi_{vac}\rangle &= \frac{1}{\sqrt{6}}\left(2|0\rangle - |1\rangle - |2\rangle\right), \\ |\Psi_{sgn}\rangle &= \frac{1}{\sqrt{2}}\left(|1\rangle - |2\rangle\right). \end{aligned} \tag{3.34}$$

These are obtained by multiplying the initial state by the appropriate F coefficients and renormalizing to unit magnitude. In the case of the second state we also introduced an extra global phase of i, which is related to the arbitrary choice of phase of the $|[\gamma^j]\rangle_R$ states.

3.2.6.2 $\mathbb{Z}_7 \times_\theta \mathbb{Z}_3$

The group $\mathbb{Z}_7 \times_\theta \mathbb{Z}_3$ has five irreducible representations. Three of them are one dimensional and set $a \to 1$ and b to a cube root of unity. The other two are three dimensional and are complex conjugates of each other.

The main new feature of this group is that the non-trivial powers of a are not all conjugate to one another. This leads to more complicated fusion coefficients. For example:

$$\begin{aligned} F_{0\to 1} &= 0, \\ F_{1\to 1} &= \frac{1}{3}A, \quad F_{2\to 1} = \frac{\gamma^2}{3}A, \quad F_{3\to 1} = \frac{\gamma}{3}B, \\ F_{4\to 1} &= \frac{\gamma}{3}A, \quad F_{5\to 1} = \frac{\gamma^2}{3}B, \quad F_{6\to 1} = \frac{1}{3}B, \end{aligned} \tag{3.35}$$

with

$$\begin{aligned} A &= \gamma^2\omega + \gamma\omega^2 + \omega^4 = \mathrm{e}^{2\pi i\frac{17}{21}} + \mathrm{e}^{2\pi i\frac{13}{21}} + \mathrm{e}^{2\pi i\frac{12}{21}}, \\ B &= \gamma^2\omega^{-1} + \gamma\omega^{-2} + \omega^{-4} = \mathrm{e}^{2\pi i\frac{11}{21}} + \mathrm{e}^{2\pi i\frac{1}{21}} + \mathrm{e}^{2\pi i\frac{9}{21}}, \end{aligned} \tag{3.36}$$

where we have chosen $\gamma = e^{2\pi i/3}$ and $\omega = e^{2\pi i/7}$. Notice how A is close in magnitude to 3 whereas B is close in magnitude to 1.

3.2.7 Operations involving electric charges

Now it is time to apply the discussion in the previous subsections to build a useful operation out of electric charges. While there seems to be a wealth of strange ancillas that could be produced using electric charges, most of them have complicated relative amplitudes or phases that are hard to use in a constructive proof of universal computation. We will therefore focus our attention on producing an operation that arises naturally from the fusion amplitudes: the projection onto the subspace orthogonal to $|0\rangle$.

Consider a qudit in the state

$$|\Psi\rangle = \sum_{i=0}^{p-1} \psi_i |i\rangle, \tag{3.37}$$

where the coefficients $\{\psi_i\}$ could either be complex numbers, or could represent the state of the rest of the system if the qudit is entangled with other qudits.

We append to the qudit an electric charge pair $|R(I)\rangle_R$ in the vacuum state of a non-abelian representation R. Using braiding, we can right multiply the state of the electric charge by some function f of the qudits flux:

$$|\Psi\rangle \otimes |R(I)\rangle_R \longrightarrow \sum_{i=0}^{p-1} \psi_i |i\rangle \otimes |R(f(i))\rangle_R. \tag{3.38}$$

We have shown in Sec. 3.2.3 that the most general function is of the form $f(i) = a^\alpha a^{\beta i} b^\delta$. Choosing $\delta \neq 0$ turns out not to be useful, and choosing $\alpha \neq 0$ can be used to get projections to the spaces orthogonal to $|i\rangle$ for $i > 0$, but this can be achieved as well with an X gate. We will therefore focus on $f(i) = a^{\beta i}$ so that we obtain the state

$$\sum_{i=0}^{p-1} \psi_i |i\rangle \otimes |R(a^{\beta i})\rangle_R = \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} F_{\beta i \to j} \psi_i |i\rangle \otimes |[\gamma^j]\rangle_R. \tag{3.39}$$

A fusion of the electric charge pair, followed by a measurement of the resulting electric charge (the feasibility of which will be the subject of Sec. 3.2.8 below), leads to a state that is proportional to

$$\sum_{i=0}^{p-1} F_{\beta i \to j} \psi_i |i\rangle, \tag{3.40}$$

where j now labels the result of the measurement in the basis $|[\gamma^j]\rangle_R$.

Because of the property $F_{0\to j} = \delta_{j,0}$, if the measurement result is $j \neq 0$, we will have projected into the space orthogonal to $|0\rangle$. Unfortunately, we will have also introduced undesired relative phases and amplitudes. The trick will be to balance these out.

Consider repeating the above procedure $p-1$ times, with β taking values from 1 to $p-1$. Furthermore, assume that in each case the fusion results in $j = 1$. The resulting state will be, up to normalization:

$$\propto \sum_{i=1}^{p-1} \left(\prod_{\beta=1}^{p-1} F_{\beta i\to 1} \right) \psi_i |i\rangle \propto \sum_{i=1}^{p-1} \psi_i |i\rangle, \tag{3.41}$$

where we have used the fact that multiplication by i, modulo p, is just a rearrangement of the values of β.

The above procedure is a probabilistic projection onto $\mathcal{K} = |0\rangle^\perp$. As usual, if we do not obtain $j = 1$ as the result of each measurement, we just discard the state being projected.

What is the probability of success of the above procedure? The probability for obtaining $j = 1$ on the first try is

$$P_{j=1} = \sum_{i=1}^{p-1} |F_{i\to 1}\psi_i|^2 \geq \min_{i>0} \left(|F_{i\to 1}|^2 \right) \sum_{i=1}^{p-1} |\psi_i|^2 . \tag{3.42}$$

On subsequent measurements, the state has previously been projected to $|0\rangle^\perp$ and renormalized. Therefore the probability of success for each trial is simply bounded by

$$P_{j=1} \geq \min_{i>0} \left(|F_{i\to 1}|^2 \right). \tag{3.43}$$

The total probability of success is just the product of these quantities. In particular, the probability p_{PP} associated with the probabilistic projection can be bounded by

$$p_{PP} \geq \min_{i>0} \left(|F_{i\to 1}|^2 \right)^{p-1} > 0, \tag{3.44}$$

where we used the fact that $|F_{i\to j}| > 0$ for $i > 0$.

Of course, the above is an underestimation of the probability of obtaining a good projection. For example, if all the results j were equal to some fixed $j > 1$, the same argument would show that a correct projection was obtained. Furthermore, there are many other ways in which the relative phases and amplitudes can cancel out. A classical computer, with knowledge of the values of $F_{i\to j}$, can keep repeating the procedure until such a cancellation occurs. The computer would also be required to stop after a long sequence of $j = 0$ results, in which case the state would have been

projected onto $|0\rangle$.

In the end, as long as p_{PP} is fixed and finite, we have produced the desired probabilistic projection to the space $|0\rangle^{\perp}$. Different values of p_{PP} will just affect the complexity of an algorithm as a multiplicative constant. Furthermore, for the small groups that are likely to appear in the laboratory, p_{PP} should be reasonably large. For example, in the case of $G = S_3$, p_{PP} can be made exponentially close to one in the number of measurements.

It should be noted that because we are working with qudits of dimension $d = p$, and the semidirect product requires $p > q \geq 2$, the above projection will always be a non-trivial operation. In fact, it will always be powerful enough to complete a universal gate-set.

At this point, all that remains to be done is to prove the universality of the gates constructed from the basic anyon operations. This will be the subject of Sec. 3.3. However, before closing this section, we shall discuss some issues regarding the measurability of electric charges and look at some alternative operations that could have been employed.

3.2.8 On the measurement of one-dimensional representations

The feasibility and accuracy of the probabilistic projection onto $|0\rangle^{\perp}$ depend crucially on being able to identify electric charges carrying one-dimensional representations. However, these charges have a special property that makes them hard to identify: when only using braiding, a one-dimensional representation is indistinguishable from the vacuum!

The reason behind the above difficulty is that one-dimensional representations of a group G are constant on conjugacy classes of G. Therefore, a magnetic charge that is braided around one of these electric charges will have its state change by an overall phase. These global phases are not measurable in quantum mechanics.

Of course, an interference experiment would produce a measurement of the charge. The standard double-slit experiment, with the electric charge located in between the slits, will produce a pattern on the screen that depends on the representation of the electric charge. However, during the experiment, the anyon will be in a superposition of spatial positions, which is no longer protected from decoherence by topology. Since the interference experiment can be repeated many times without affecting the electric charge, this may not necessarily be a problem. However, it does involve working in a regime where the anyons can be treated as waves rather than particles.

On the bright side, these electric charges can also be detected by fusion, assuming the availability of electric charge ancillas with one-dimensional representations. Their fusion rules are particularly simple because these states have a one-dimensional internal Hilbert space. Furthermore, their fusions always produce unique results. If $\gamma(g)$ and $\gamma'(g)$ are two one-dimensional representations of a group G, then the fusion of the electric charges carrying these representations produces a charge of representation $\gamma''(g) = \gamma(g)\gamma'(g)$. A charge will only fuse into the vacuum when fused with its

conjugate representation. Therefore, after a series of fusions that end up producing the vacuum state, we can determine the representation of the original electric charge.

In fact, for groups with $q = 2$ such as S_3, there is a further simplification. In these groups there are only two one-dimensional representations: the vacuum and the sign representation. Since the fusion of $|R(a^i)\rangle_R$ produces a one-dimensional charge, if it does not fuse into the vacuum, then it must have produced the sign charge. Therefore, for these groups, we do not even require one-dimensional electric charge ancillas.

3.2.9 Other possibilities

In this section, we will briefly discuss one last possibility for producing useful operations: using the products of fusion. Though not strictly needed to complete a universal gate-set, this subsection is an interesting study of alternative operations and the effects of decoherence during fusion.

At first sight, it appears that the projection onto $|0\rangle^\perp$ can be done without using electric charges with the following procedure: first fuse one anyon from the state to be measured with a b^{-1} flux. Only the $|0\rangle$ state can fuse into the vacuum. If an anyon remains, fuse again with a b flux to restore it to its previous state, and pair it with its old partner. Repeating the procedure multiple times (because the $|0\rangle$ could turn into an electric charge rather than the vacuum) yields the desired projection.

There are, however, two problems with the above construction. The first, and smaller, problem is that when fusing with b^{-1} or with b we could be turning our magnetic charges into dyons. For groups of the form $\mathbb{Z}_p \times_\theta \mathbb{Z}_q$ the dyonic electric charges are all one dimensional, however, and will therefore have no effect on braiding, as discussed in the previous section. The probabilities of fusion into the vacuum will be reduced, and therefore so will the respective projection probabilities, but they will still remain non-zero. In fact, a careful examination of the operations constructed so far shows that they work with a probabilistic mixture of dyons and regular magnetic charges.

The second and larger problem, though, is decoherence. The fluxes $a^i b a^{-i} b^{-1} = a^{i(1-t)}$ belong, in general, to different conjugacy classes and therefore different superselection sectors. When the quantum state is encoded in this form, it is susceptible to decohere into the different superselection sectors.

When does this decoherence occur? It occurs during fusion. In general, fusion takes two n-dimensional Hilbert spaces $\mathcal{H}$ and maps them to one: $\mathcal{H}_1 \times \mathcal{H}_2 \to \mathcal{H}_3$. But quantum mechanics is unitary; therefore, what must really be happening is a mapping to a tensor product of $\mathcal{H}$ and the environment: $\mathcal{H}_1 \times \mathcal{H}_2 \to \mathcal{H}_3 \times \mathcal{E}$. When two states are mapped onto new states that are orthogonal in the environment subspace, decoherence occurs.

How do we know if states will have orthogonal environment components after fusion? If two states belong to the same superselection sectors, they are related by symmetry, which protects them

from decoherence. This may not be the case when they come from different superselection sectors, though.

For example, consider the states $|a^i\rangle \otimes |a^j b\rangle$ for i and j between 0 and $p-1$, where the kets will denote single anyons in this paragraph and the next. States of different j are all in the same conjugacy class, but states of different i are grouped into conjugacy classes of q elements (except for $i = 0$, which is its own conjugacy class). In total, we are talking about p^2 states.

These states fuse into the states with flux $a^k b$ for $0 \leq k < p$. The resulting states may also have one of q electric charges. In total, we fuse into a space containing pq states. Since $pq < p^2$, what must be happening is that different conjugacy classes are mapped to states that are orthogonal in the environment subspace.

Note that the decoherence seems to occur when fusing out of a state made up of different superselection sectors. However, fusion is the only operation that could have measured the relative phase between the sectors, and it clearly does not. Therefore, it is acceptable to assume that the decoherence occurs as soon as states are mapped into different superselection sectors.

Returning to the question of alternative implementations of the projection onto $|0\rangle^\perp$, it is clear that the procedure described above does not achieve its goals without causing decoherence in the general case. However, in the special case when $q = p - 1$, the non-trivial powers of a form one conjugacy class. Therefore, the above trick can produce a projection onto $|0\rangle^\perp$ using only magnetic charges. Of course, $q = p - 1$ only holds for $G = S_3$.

For other groups, the operation could become useful if we could tell into which superselection sector the state decohered, producing a probabilistic projection onto a smaller space. The smaller projections may also be computationally powerful. However, since we have completed a universal gate-set without the results of this subsection, we shall work on proving universality from the previously constructed gates, rather than pursuing this matter further.

3.3 Gate-set universality

The goal of this section is to prove the universality of the following qudit gate-set, which includes measurements:

1. Controlled-X,
2. Probabilistic projection onto $|0\rangle$,
3. Probabilistic projection onto $|\tilde{0}\rangle$,
4. Probabilistic projection onto $|0\rangle^\perp$,

where we assume that the qudits are of dimension $d > 2$, with d prime. The first requirement on d is needed to make the gate-set universal, whereas the second one will allow us to relate this gate-set to Gottesman's gate-set [Got98]. The above gate-set must be supplemented by a controlling computer capable of universal classical computation.

The above gates were selected as those arising naturally from the anyons based on the groups $\mathbb{Z}_p \times_\theta \mathbb{Z}_q$. The proof of universality of the above gate-set is the last step needed to show that universal quantum computation is feasible with these anyons.

The proof of universality will proceed in two steps. In the first step we will turn the second and third gates into proper measurements in the Z and X bases. Most of the methods of the first step were described while building computation with non-solvable anyons in Chap. 2, though we include the discussion for completeness. The second step involves using the probabilistic projection onto $|0\rangle^\perp$ to construct magic states that complete the universal gate-set. This is the new element needed to achieve universality with solvable anyons.

3.3.1 Non-destructive measurement of Z and X

By the end of this subsection we will have constructed measurements in the Z and X bases. These measurements will be non-destructive in the sense that if result i was obtained, the measured qudit will be in state $|i\rangle$ or $|\tilde{i}\rangle$, respectively. Because the measurements in question are complete, the non-destructive requirement can be achieved by having ancillas for every eigenstate of X and Z, and then using the controlled-X to swap the ancillas into the computational space.

The construction begins by producing a set of basic ancillas. Along the way we will also produce the X and Z unitary gates.

3.3.1.1 $|0\rangle$ and $|\tilde{0}\rangle$ ancillas

Clearly, given $|0\rangle$ ancillas we can use the third gate to produce $|\tilde{0}\rangle$ ancillas. Similarly, given $|\tilde{0}\rangle$ ancillas we can use the second gate to produce $|0\rangle$ ancillas. Therefore, if the initial state of the quantum computer overlaps with either state, we can produce both kinds of ancilla.

Usually, the initial state of the quantum computer is $|0\rangle$. However, by using the controlled-X gate, in combination with the projections onto $|0\rangle$, we can obtain these states no matter what the qudits are initialized to. The procedure is just to apply a controlled-X^{-1} (equivalent to $d-1$ controlled-X gates) to two qudits, and then to project the target to the $|0\rangle$ space. If the initial state had some overlap with any of the states $|i\rangle \otimes |i\rangle$, then this produces the desired ancillas. Furthermore, even if we allow states that are initially entangled, once we involve more than d qudits, at least one pair must have an overlap with the diagonal states. Therefore, $|0\rangle$ states can always be produced.

Henceforth, we shall assume an ample supply of $|0\rangle$ and $|\tilde{0}\rangle$ ancillas.

3.3.1.2 $|1\rangle$ states, $|\tilde{1}\rangle$ states; X gates, Z gates

The next step is to produce $|1\rangle$ and $|\tilde{1}\rangle$ ancillas. The importance of these ancillas is that they will break the symmetry currently present in the one-qudit Hilbert space.

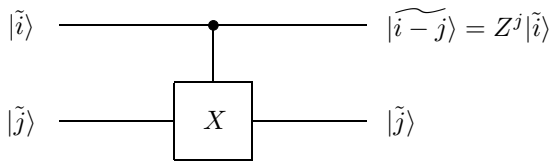
There are two symmetries in the Hilbert space that are not fixed by the basic four gates of our set. The first symmetry is a relabeling $|ix\rangle \rightarrow |i\rangle$, calculated modulo d, for some $0 < x < d$. The second, is the relabeling $Z^y \rightarrow Z$, for integer $0 < y < d$. For fixed x, the second symmetry is a relabeling of our d^{th} root of unity ω by $\omega^y \rightarrow \omega$ and a relabeling $|\widetilde{jy}\rangle \rightarrow |\tilde{j}\rangle$.

Therefore, given an ancilla in a state $|x\rangle$, with $x > 0$, we can just rename it so that it becomes a $|1\rangle$ ancilla. Similarly, given an ancilla in a state $|\tilde{y}\rangle$, $y > 0$, we can relabel it as $|\tilde{1}\rangle$. In fact, both can be done simultaneously in a consistent fashion, even if we do not know the values of x and y.

The initial states $|x\rangle$ and $|\tilde{y}\rangle$ can be obtained from two maximally mixed states. The maximally mixed states can be described either as a state $|x\rangle$ with x chosen at random, or a state $|\tilde{y}\rangle$ with y chosen at random. Therefore, two maximally mixed states serve our purpose as long as we do not obtain $x = 0$ or $y = 0$. These two bad cases will be detected below, in which case the process can be restarted with two new mixed states.

To produce the maximally mixed states we apply a controlled-X with $|\tilde{0}\rangle$ as source and $|0\rangle$ as target. The result is a maximally entangled state, which can be turned into a maximally mixed state by discarding one of the two qudits. Two of these mixed states will serve as our ancillas.

Given our two ancillas, which we have now labeled $|1\rangle$ and $|\tilde{1}\rangle$, we can build X and Z gates, which are consistent with the new labeling. The X gate is clearly just a controlled-X with a $|1\rangle$ state as control, whereas the Z gate is just a controlled-X with a $|\tilde{1}\rangle$ as target. The less familiar second construction is just a specific case of the following circuit:



At this point, if we were unlucky enough to get $x = 0$ or $y = 0$, then one of the transformations X or Z will be the identity operator. This can easily be checked by applying them to $|0\rangle$ or $|\tilde{0}\rangle$ ancillas and then using the available probabilistic projections.

The X and Z gates can also be used to produce a reservoir of $|1\rangle$ and $|\tilde{1}\rangle$ ancillas that will be consistent with the original states. Two elements in the reservoir can also be compared, for example, by applying a Z built from one ancilla followed by a Z^{-1} built from the other. Therefore, even if the states were to decay over time, by using majority voting the damaged states can be weeded out.

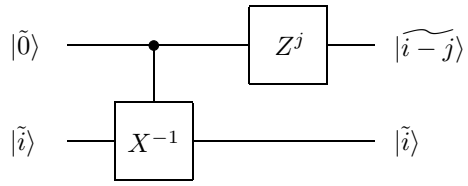
In some cases, the one-qubit Hilbert spaces do have natural $|1\rangle$ or $|\tilde{1}\rangle$ states, which implies a natural way of measuring or obtaining such states. For those systems, either the natural ancillas

or the arbitrary ones constructed above can be used. For example, for the anyons $|1\rangle = |aba^{-1}\rangle$. However, choosing a different $|1\rangle$ state is equivalent to choosing a different element a.

3.3.1.3 Measurements of Z and X

At this point all the elements are in place to produce measurements in either the Z basis or the X basis.

The key element of the X basis measurement is the circuit



applied to a $|\tilde{0}\rangle$ ancilla and the state to be measured. If the above circuit is repeated many times, each time with a different $|\tilde{0}\rangle$ ancilla, and with j varying from 0 to $d-1$, we obtain the transformation

$$\sum_i \beta_i |\tilde{i}\rangle \longrightarrow \sum_i \beta_i |\tilde{i}\rangle \otimes |\tilde{i}\rangle \cdots \otimes |\widetilde{i-1}\rangle \otimes |\widetilde{i-1}\rangle \cdots \otimes |\widetilde{i-d+1}\rangle. \tag{3.45}$$

A probabilistic projection onto $|\tilde{0}\rangle$ can then be applied to each qudit. If one of the qudits of the form $|\widetilde{i-j}\rangle$ projects onto the space $|\tilde{0}\rangle$, then the outcome of the measurement is j.

Note that because of the one-sided error model of the probabilistic projection, an erroneous measurement result can never be obtained, no matter how small p_{PP} is. The worst possible outcome is that after all the qudits have been measured, no conclusion can be reached. Of course, a standard, small, two-sided probability of error can also be made exponentially small by using enough qudits in the above measurement.

The measurement in the Z basis proceeds similarly, where the transformation

$$\sum_i \alpha_i |i\rangle \longrightarrow \sum_i \alpha_i |i\rangle \otimes |i\rangle \cdots \otimes |i-1\rangle \otimes |i-1\rangle \cdots \otimes |i-d+1\rangle \tag{3.46}$$

is performed using the X and controlled-X gates, followed by a probabilistic projection onto $|0\rangle$.

Finally, the above measurements can be performed non-destructively, by projecting all but one of the qudits. Alternatively, the eigenstates of X and Z can be directly constructed from these gates and $|0\rangle$ or $|\tilde{0}\rangle$ eigenstates.

3.3.2 Completing the gate-set

So far, we have only shown that our gates can realize operations in the Clifford group. In order to achieve universal quantum computation we need to complete the gate-set with an operation outside

the Clifford group.

It was shown in Chap. 2 that the Toffoli gate, combined with measurements in the X and Z bases, is universal for quantum computation. Therefore, a successful construction of the Toffoli out of our gate-set will prove it universal. The Toffoli gate will be constructed out of the previously described operations, together with the thus far unused probabilistic projection onto $|0\rangle^{\perp}$.

In addition to producing measurement gates, probabilistic projections are particularly useful for preparing magic states, which are ancillas whose use allows us to perform new gates such as the Toffoli. In particular, we shall show that we can produce the two magic states

$$\begin{aligned} |\phi_{M1}\rangle &= \frac{1}{d}\sum_{i,j}|i\rangle\otimes|j\rangle\otimes|ij\rangle, \\ |\phi_{M2}\rangle &= \frac{1}{d}\sum_{i,j}\omega^{\delta_{i,0}\delta_{j,0}}|i\rangle\otimes|j\rangle, \end{aligned} \tag{3.47}$$

where $\delta_{i,j}$ is the Kronecker delta function. The first of these states produces the Toffoli gate up to some errors in the Clifford group. The second magic state allows us to correct these errors, and in fact, allows the construction of the complete Clifford group even without the use of the first magic state.

We shall begin by discussing how to use each of the magic states and then afterwards turn to the task of describing their construction out of the available operations.

3.3.2.1 Using $|\phi_{M1}\rangle$

The magic state $|\phi_{M1}\rangle$ and its use in producing the Toffoli gate was first introduced by Shor [Sho96] and generalized to qudits in [Got98]. We shall give a brief description of its use in order to give an account of the exact Clifford group operations needed in the last step as corrections.

The procedure begins with a general state

$$|\Psi\rangle = \sum_{a,b,c}\psi_{a,b,c}|a\rangle\otimes|b\rangle\otimes|c\rangle, \tag{3.48}$$

to which an ancilla $|\phi_{M1}\rangle$ is appended. A controlled-X^{-1} is applied to the first data qudit with the first ancilla qudit as control. Similarly, a controlled-X^{-1} is applied to the second data qudit from the second ancilla qudit, and a controlled-X is applied to the third ancilla qudit, from the third data qudit. The first two data qudits are then measured in the Z basis, and the third data qudit is measured in the X basis. If the results of the measurements are α, β and γ, respectively, then the remaining qudits are left in the state

$$\sum_{a,b,c}\psi_{a,b,c}\,\omega^{\gamma c}\,|a-\alpha\rangle\otimes|b-\beta\rangle\otimes|(a-\alpha)(b-\beta)+c\rangle. \tag{3.49}$$

The corrections begin by applying an $X^{\alpha} \otimes X^{\beta} \otimes X^{-\alpha\beta}$ gate followed by a controlled-X^{β} from the first qudit to the third qudit and a controlled-X^{α} from the second qudit to the third qudit. The state then becomes

$$\sum_{a,b,c} \psi_{a,b,c}\, \omega^{\gamma c} \, |a\rangle \otimes |b\rangle \otimes |ab + c\rangle. \tag{3.50}$$

All that is needed to complete the Toffoli gate is a $Z^{-\gamma}$ gate applied to the third qudit and a phase $\omega^{\gamma ab}$ applied to the first two qudits. Unfortunately, we must first build the latter transformation out of the second magic state.

3.3.2.2 Using $|\phi_{M2}\rangle$

Once again, the magic state is appended to a pair of qudits. Now controlled-X gates are applied with the data qudits as source and the ancilla qudits as targets. Then the ancilla qudits are measured in the computational basis. The outcomes α and β will be uniformly distributed, and at the end we will have produced the transformation

$$\sum_{a,b} \psi_{a,b} |a\rangle \otimes |b\rangle \longrightarrow \sum_{a,b} \psi_{a,b}\, \omega^{\delta_{a,\alpha}\delta_{b,\beta}} \, |a\rangle \otimes |b\rangle. \tag{3.51}$$

This procedure randomly and uniformly chooses a computational basis state and multiplies it by a phase of ω. Repeated application of this transformation will eventually yield any of the d^{d^2} states of the form

$$\sum_{a,b} \psi_{a,b}\, \omega^{f(a,b)} \, |a\rangle \otimes |b\rangle, \tag{3.52}$$

where f is an arbitrary integer-valued function. This process is effectively a classical random walk on a d^2 dimensional periodic lattice with d^{d^2} nodes, where each use of a magic state is equivalent to taking one step. Because the lattice is finite, after a polynomially large number of steps the probability of not having arrived at least once at any one of the above states becomes exponentially small.

The final correction needed to complete the Toffoli gate was the phase transformation to the state with $f(a, b) = \gamma ab$ and can therefore be realized using many copies of the second magic state. All that remains to prove universality is to describe the production of the magic states.

3.3.2.3 Making the magic states

The final piece of the puzzle is the production of the magic states using the probabilistic projection onto $|0\rangle^{\perp}$.

Probabilistic projections onto a subspace are particularly powerful for making magic states because it can be assumed that they successfully project into the subspace every time. That is, if the probabilistic projection does not project onto the desired subspace, the state is tossed out, and the procedure is restarted from the beginning. Therefore, the probabilistic projection onto $|0\rangle^{\perp}$ effectively takes a state and removes the $|0\rangle$ component of the state:

$$\sum_{i=0}^{d-1} \alpha_i |i\rangle \longrightarrow A \sum_{i=1}^{d-1} \alpha_i |i\rangle, \tag{3.53}$$

where A is some normalization constant. In fact, by combining this projection with the X gate, we can remove any of the components $|i\rangle$.

The main strategy for this section is to construct a series of ancilla states of increasing complexity, until finally the desired magic states are obtained. At this point, we have a supply of ancillas of the form $|i\rangle$ and $|\tilde{j}\rangle$ for any i and j. From the $|\tilde{0}\rangle$ state we can also make the ancilla $(|0\rangle + |1\rangle)/\sqrt{2}$ by removing all $|i\rangle$ for $i > 1$ with the probabilistic projection.

The next step is to produce entangled two-qudit ancillas. Given a supply of ancillas of the form $|\Psi\rangle = \sum_i \psi_i |i\rangle$ we shall produce ancillas of the form

$$|\Psi'\rangle = \left(\psi_0 |0\rangle \otimes |1\rangle + \sum_{i=1}^{d-1} \psi_i |i\rangle \otimes |0\rangle \right) = \sum_{i=0}^{d-1} \psi_i |i\rangle \otimes |\delta_{i,0}\rangle. \tag{3.54}$$

The procedure begins with the state

$$|\Psi\rangle \otimes \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) = \frac{1}{\sqrt{2}} \sum_{i=0}^{d-1} \sum_{j=0}^{1} \psi_i |i\rangle \otimes |j\rangle, \tag{3.55}$$

which in general has $2d$ non-zero coefficients. We need to remove d of these coefficients to obtain the state $|\Psi'\rangle$.

The procedure, done once for each k from 1 to $d-1$, is the following: First, apply a controlled-X^k with the left qudit as source and the right qudit as target. Then, the right qudit is projected onto $|0\rangle^{\perp}$, and finally the controlled-X^k is undone. For each k, we remove the components $|0\rangle \otimes |0\rangle$ and $|-1/k\rangle \otimes |1\rangle$. The operation $-1/k$ is modulo d as usual, and ranges over all integers between 1 and $d-1$ because d is prime. Therefore, given a supply of $|\Psi\rangle$ ancillas, we can probabilistically convert some of them into a supply of $|\Psi'\rangle$ ancillas.

Note that the above procedure works even if the coefficients ψ_i represent the state of other qudits, as long as these are ancilla qudits that can be tossed out if the projection procedure fails. In the

same spirit, given ancillas of the form

$$|\Phi\rangle = \sum_{i=0}^{1}\sum_{j=0}^{1} \phi_{i,j}|i\rangle \otimes |j\rangle, \tag{3.56}$$

we can produce the three-qudit ancillas

$$|\Phi'\rangle = \sum_{i=0}^{1}\sum_{j=0}^{1} \phi_{i,j}|i\rangle \otimes |j\rangle \otimes |\delta_{i,0}\delta_{j,0}\rangle. \tag{3.57}$$

The procedure again involves appending $(|0\rangle + |1\rangle)/\sqrt{2}$ to the ancilla $|\Phi\rangle$, which now generically has eight non-zero coefficients, and removing four of them. This is done with a set of controlled-X gates with the third qudit as target, followed by a probabilistic projection of the third qudit onto $|0\rangle^{\perp}$, followed by the inverse controlled-X gates. If we use two controlled-X^{-1} gates controlled by the first two qudits, respectively, the projection will remove the components with labels $|0\rangle|0\rangle|0\rangle$, $|1\rangle|0\rangle|1\rangle$ and $|0\rangle|1\rangle|1\rangle$. In addition, using two controlled-$X^{(d-1)/2}$ gates, we remove $|1\rangle|1\rangle|1\rangle$ and $|0\rangle|0\rangle|0\rangle$ (again). These are the four states that need to be removed to produce the ancilla $|\Phi'\rangle$.

The above two procedures allow us to finally produce the desired magic states. Starting with $|\tilde{0}\rangle \otimes |\tilde{0}\rangle$, we apply the first procedure to each ancilla and then apply the second procedure to the appended qudits. The resulting state is

$$\frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1} |i\rangle \otimes |j\rangle \otimes |\delta_{i,0}\rangle \otimes |\delta_{j,0}\rangle \otimes |\delta_{i,0}\delta_{j,0}\rangle. \tag{3.58}$$

If the last three qudits are measured in the X basis, and the results are 0, 0 and 1, respectively, then we will have produced the magic state $|\phi_{M2}\rangle$.

In fact, measuring in the X basis and only accepting if the result is zero is a convenient way to unentangle the system with temporary qudits. Therefore, the previously described procedures can be combined into the probabilistic transformation

$$\sum_{i=0}^{d-1}\sum_{j=0}^{d-1} \psi_{i,j}|i\rangle \otimes |j\rangle \longrightarrow \sum_{i=0}^{d-1}\sum_{j=0}^{d-1} \psi_{i,j}|i\rangle \otimes |j\rangle \otimes |\delta_{i,n}\delta_{j,m}\rangle, \tag{3.59}$$

where the first state is either transformed into the second state with some nonzero probability, or else it is damaged. The above transformation has only been discussed so far for $n = m = 0$, but a trivial use of X gates before and after the transformation will allow any n and m.

Starting with $|\tilde{0}\rangle \otimes |\tilde{0}\rangle \otimes |0\rangle$, repeated application of the above procedure can produce

$$\begin{aligned} \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}|i\rangle\otimes|j\rangle\otimes|0\rangle \quad &\longrightarrow \quad \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}|i\rangle\otimes|j\rangle\otimes|0\rangle\bigotimes_{n=0}^{d-1}\bigotimes_{m=0}^{d-1}|\delta_{i,n}\delta_{j,m}\rangle \\ &\longrightarrow \quad \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}|i\rangle\otimes|j\rangle\otimes|ij\rangle\bigotimes_{n=0}^{d-1}\bigotimes_{m=0}^{d-1}|\delta_{i,n}\delta_{j,m}\rangle, \end{aligned} \tag{3.60}$$

where the second step involves only controlled-X gates from the extra qudits to the third qudit. Erasing the extra qudits with a measurement in the X basis, and retaining only when all results are zero, produces the desired magic state $|\phi_{M1}\rangle$.

The construction of the magic states out of the probabilistic projection onto $|0\rangle^{\perp}$ completes the description of the Toffoli gate. Though the procedures of this section are far from optimal in terms of resources, they are sufficient to demonstrate universality. In particular, this completes the proof that universal quantum computation is feasible with anyons from groups of the form $\mathbb{Z}_p \times_\theta \mathbb{Z}_q$.

3.4 Computational power of magnetic charges

In this section, we will be interested in classifying the computational power that can be achieved by braiding anyonic magnetic charges of a finite group. The range of operations that can be achieved by braiding is closely related to the structure of the group to which the magnetic charges belong. In particular, the possibility of realizing the operations of controlled-X and Toffoli (equivalently a doubly-controlled-X) are respectively related to the group properties of nilpotency and solvability. These standard properties of group theory will also be defined below.

There are certain important assumptions that go into the discussion in this section. First, we assume that each qubit is carried by a pair of anyons. Furthermore, we choose a computational basis corresponding to the states of definite flux (e.g., $|0\rangle = |g\rangle$ for some $g \in G$). We remind the reader at this point that the state $|g\rangle$ corresponds to an anyon of magnetic charge g paired with a compensating anyon of charge g^{-1} whose only purpose is to allow the pair to move through the system without introducing undesired correlations. Finally, we will restrict the discussion to operations that can be achieved by braiding magnetic charges. The consequences of lifting these restrictions will be discussed near the end of this section.

Let the fluxes corresponding to the zero and one states be the elements b, $b' \in G$, respectively. If we desire a coherent superposition between the zero and one states, they must be in the same conjugacy class, and therefore $b' = aba^{-1}$ for some non-trivial $a \in G$. This is summarized by:

$$|0\rangle = |b\rangle, \quad |1\rangle = |b'\rangle = |aba^{-1}\rangle. \tag{3.61}$$

Even if the basis in use is a qudit basis, with additional states, we will only concern ourselves with states that have support on the above two basis vectors.

Consider now a pair of these states. We are interested in the operations that can be achieved by conjugating the second state by the flux of the first state, with the help of ancillas. Let $g \in \{b, b'\}$ be the flux of the first state. The most general conjugation possible is by a function of the form

$$\begin{aligned} f(g) &= c_1 g c_2 g c_3 g \cdots g c_n \\ &= \left(d_1 g' d_1^{-1}\right)\left(d_2 g' d_2^{-1}\right)\cdots\left(d_{n-1} g' d_{n-1}^{-1}\right) d_n, \end{aligned} \tag{3.62}$$

for some fixed elements $\{c_i\} \in G$. In the second line, the expression has been rewritten in terms of $g' = gb^{-1}$ and new elements $\{d_i\} \in G$, which can easily be determined in terms of $\{c_i\} \in G$. For example, $d_2 = c_1 b c_2$.

The power of the second line is that it expresses the conjugation as a composition of two basic operations. The first is a conjugation by an ancilla with flux d_n and is independent of the state of the first qubit. The second is conjugation by a product of conjugates of g', which was defined so that if $g = b$ then $g' = 1$ and the product of its conjugates is trivial. In the other case, if $g = b'$ then $g' = [a, b] \equiv aba^{-1}b^{-1}$, and the operation is conjugation by a product of conjugates of $[a, b]$.

We define $\mathcal{C}_G(x)$ as the conjugacy class of x in G, and $\mathcal{C}_G^{\#}(x)$ as the group generated by the elements in $\mathcal{C}_G(x)$. The operations discussed so far are conjugation by fixed elements in G, and controlled conjugation by elements in $\mathcal{C}_G^{\#}([a, b])$.

The most natural controlled operation is the logical controlled-X gate, which acts as a controlled conjugation by a. Naturally, if $a^2 \neq 1$, then we could arrive at the qudit state $|2\rangle = |a^2 b a^{-2}\rangle$. However, our interest lies in proving that certain groups cannot produce a controlled-X, in which case it is sufficient to prove that a controlled conjugation by a is unfeasible.

It seems that a requirement for a controlled conjugation by a is the existence of elements a, b such that $a \in \mathcal{C}_G^{\#}([a, b])$. There is a potential loophole in the argument, though, because different qubits could use different basis fluxes. The target qubit could use b_2 as the zero state, and $a_2 b_2 a_2^{-1}$ as the one state. If $a_2 \in \mathcal{C}_G^{\#}([a_1, b_1])$ then the controlled-X would be possible. Considering many qubits requires a sequence of non-trivial elements $\{a_i\}$ and $\{b_i\}$, which satisfy, at a minimum, the conditions:

$$a_{i+1} \in \mathcal{C}_G^{\#}([a_i, b_i]). \tag{3.63}$$

The above equations are related to the series of subgroups of G, defined by

$$G^{((j+1))} = \left[G^{((j))}, G\right], \tag{3.64}$$

with base case $G^{((0))} = G$. By definition, if $a_i \in G^{((j))}$ then $[a_i, b_i] \in G^{((j+1))}$. Furthermore, since the group $G^{((j+1))}$ is normal in G, the requirement on a_{i+1} reads

$$a_{i+1} \in \mathcal{C}_G^{\#}\left([a_i, b_i]\right) \subset G^{((j+1))}. \tag{3.65}$$

Of course, $a_1 \in G^{((0))} = G$. Therefore, repeating the above argument shows that a controlled-X requires $a_i \in G^{((i-1))}$ with $a_i \neq 1$ for every $i \geq 1$.

Given that G is finite, and $G^{((j+1))} \subset G^{((j))}$, the series must converge after a finite number of subgroups to some final subgroup $G^{((\infty))}$. The final subgroup can either be trivial or non-trivial. The groups with $G^{((\infty))} = \{1\}$ are called nilpotent. The conclusion thus far is that nilpotent groups cannot implement a controlled-X by braiding. The inverse of this statement—i.e., that groups that are not nilpotent can implement the controlled-X—will be shown in Sec. 3.5.

3.4.1 Conjugations with multiple sources

A similar analysis can be used to study the relationship between group structure and gates produced using multiple qubits as sources of conjugation. Clearly any group that is not nilpotent can produce a series of controlled-X gates with different sources. However, certain groups are capable of producing much more powerful gates such as the Toffoli, which is universal for classical computation.

In the rest of this section we shall prove that groups that are solvable cannot produce a Toffoli gate, or equivalently universal classical computation, by braiding magnetic charges. This connection between universality for classical computation and non-solvability had been previously identified by Barrington [Bar89] in 1989. Though we shall mostly be interested in groups that are solvable, this result will place limits on the power that we can expect to obtain from braiding magnetic charges.

Just as above, the most general conjugation with m sources is the conjugation by a function of the form

$$f(g_1, \ldots, g_m) = \left(d_1 g'_{i_1} d_1^{-1}\right)\left(d_2 g'_{i_2} d_2^{-1}\right) \cdots \left(d_{n-1} g'_{i_{n-1}} d_{n-1}^{-1}\right) d_n, \tag{3.66}$$

where $g'_i = g_i b^{-1}$ and the indices i take values from 1 to m. For brevity, we assume that all qubits are expressed in the same basis, though the general case would not be very different.

The Toffoli gate is simply a conjugation by a function $f_T(g'_1, g'_2)$, such that

$$f_T(c^k, c^l) = a^{kl}, \tag{3.67}$$

which has been expressed as a function of g'_i and where we introduced $c \equiv [a, b]$. In order to produce the Toffoli gate using conjugation alone, we must be able to express the above equation in the form of Eq. (3.66) with $m = 2$. We shall show that this is not possible for a solvable group.

For $m = 2$, Eq. (3.66) is a product of conjugates of g_1' and g_2'. We can rewrite it by moving all the conjugates of g_1' to the left, and all the conjugates of g_2' to the right. In the center we will pick up factors of the form $[d_i g_1' d_i^{-1}, d_j g_2' d_j^{-1}]$ and commutators of commutators, and so on. In the end we will obtain:

$$f(g_1', g_2') = f_1(g_1') f_C(g_1', g_2') f_2(g_2') d_n, \tag{3.68}$$

where d_n is a constant element of G, f_i is a product of conjugates of g_i', and $f_C(g_1', g_2')$ is the factor with all the commutators. The function f_C has the property that $f_C(g_1', 1) = f_C(1, g_2') = 1$.

Setting $f = f_T$ implies the conditions

$$\begin{aligned} 1 &= f(1,1) = d_n, \\ 1 &= f(c,1) = f_1(c) d_n, \\ 1 &= f(1,c) = f_2(c) d_n, \\ a &= f(c,c) = f_1(c) f_C(c,c) f_2(c) d_n, \end{aligned} \tag{3.69}$$

which imply $f_C(c, c) = a$.

However, f_C has the additional property that, if N is a normal subgroup of G containing c, then $f_C(c, c) \in [N, N]$. Furthermore, since $c = [a, b]$, the requirement on c needed to express the Toffoli function in product form is

$$c \in N \Longrightarrow c \in [N, N], \tag{3.70}$$

for any normal subgroup N. This condition is related to the series of subgroups defined by

$$G^{(j+1)} = \left[G^{(j)}, G^{(j)} \right], \tag{3.71}$$

again with base case $G^{(0)} = G$. Just as before, this series must converge to a final subgroup $G^{(\infty)}$. The groups where $G^{(\infty)} = \{1\}$ are known as solvable. Any group that is nilpotent is also solvable.

Because the subgroups $G^{(j)}$ are all normal in G, the requirement of Eq. (3.70) can only be satisfied if c, which by definition cannot be 1, is contained in $G^{(\infty)}$. We have therefore shown that if the group is solvable, then the function f_T cannot be expressed in product form, and therefore we cannot conjugate by it. This is true even if the target of conjugation is in a known state, which implies that even if we had used the target as a source of conjugations as well (i.e., by using it to conjugate ancillas, and then using the ancillas) the Toffoli gate would still not be feasible by using only braiding of anyons from a solvable group.

The fact that the Toffoli gate can be produced for non-solvable groups is a consequence of the

Abelian	Nilpotent	Solvable	Example	Computational Power
yes	yes	yes	$\mathbb{Z}_2$	I
no	yes	yes	Q	X
no	no	yes	S_3	Controlled-X
no	no	no	A_5	Toffoli

Table 3.1: Computational power achieved by conjugation for different groups.

results of [Bar89] (also proved in the last section of Chap. 2) and will not be discussed here. In fact, the computational model discussed in this section resembles the non-uniform deterministic finite automata presented in [BST90]. For non-solvable groups, the two models are almost identical. Nonetheless, for solvable groups, the magnetic charges presented in this section have significantly less computational power because the zero and one states have to be represented by group elements in the same conjugacy class.

3.4.2 Summary of computational power

The results discussed so far have been summarized by Table 3.1. For each type of group, it describes the computational operations that can be achieved through braiding of magnetic fluxes, as well as an example. The examples are the smallest group in the class, with the exception of the abelian case where the trivial group could also be listed. For the non-abelian, nilpotent case there are two examples with eight elements: the dihedral group D_4, and the quaternionic group Q, which is listed in the table and has elements $\pm 1, \pm i, \pm j, \pm k$.

The most basic case is when G is abelian, in which case it is also nilpotent and solvable. Clearly conjugation can only produce the identity transformation. In fact, every superselection sector consists of a one-dimensional Hilbert space, and therefore quantum information cannot even be stored in abelian anyons in a topologically protected manner.

At the other extreme are anyons from non-solvable groups. Universal classical computation can be accomplished through braiding, and universal quantum computation can be obtained by completing the gate-set with measurements in the X and Z bases. The complete construction for this case was discussed in the previous chapter.

Anyons from groups that are solvable but not nilpotent can also be used for universal quantum computation, but the construction is more complicated. A controlled-X can be constructed from flux braiding, and measurements in the X and Z bases can be constructed in a manner similar to the non-solvable groups. However, to complete a universal gate-set, fusions of electric charges must be employed. The proof of universality, along with the details of the gates, will be the subject of the rest of this chapter.

Finally, anyons from groups that are nilpotent seem insufficient for universal computation. In

the constructions for the non-nilpotent groups, the only operation that can produce entanglement between multiple qudits is the controlled-X or Toffoli gates obtained by braiding fluxes. However, for nilpotent groups, braiding fluxes does not seem to yield an operation capable of producing entanglement. Either a new type of operation, or a different basis must be used. Simple modifications to the basis, such as encoding a qudit on multiple anyons, are of no help. However, there are countless strange bases that are hard to discredit. For example, a lattice of electric charges could serve as a Hilbert space, with magnetic charges used to create or measure entanglement among the charges. Therefore, while the prospects of universal computation with nilpotent anyons seem bleak, the question remains open.

3.5 Solvable non-nilpotent groups

In this section, we will prove that anyons based on a finite group that is solvable but not nilpotent are sufficient for universal quantum computation. The first step will be to decompose an arbitrary group G that is solvable but non-nilpotent, into a form similar to the previously studied $\mathbb{Z}_p \times_\theta \mathbb{Z}_q$ groups. The proof of universality will then be a small generalization of the ideas presented in Sec. 3.2.

3.5.1 Group decomposition

Let G be a group as above, and define $H \equiv G^{((\infty))}$ in terms of the series discussed in Sec. 3.4. Because G is non-nilpotent H is non-trivial, and because G is solvable $H \neq G$. Furthermore, H is normal in G, and G/H is nilpotent. The second fact is due to

$$(G/H)^{((i+1))} = \left[(G/H)^{((i))}, (G/H)\right] = \left[G^{((i))}, G\right]/H = G^{((i+1))}/H, \tag{3.72}$$

and therefore $(G/H)^{((\infty))} = H/H = \{1\}$.

Any nilpotent group can be written as the direct product of its Sylow p-groups, which are groups whose order is a prime power. Therefore

$$G/H = K_{q_1} \times K_{q_2} \times \cdots \times K_{q_l}, \tag{3.73}$$

where K_q denotes a group of order q^m for some prime q and integer m. We further define NK_{q_i} to be the lifting of K_{q_i} to the full group G that is $NK_{q_i}/N = K_{q_i}$. Note that to maintain consistency with the notation in Sec. 3.2, the primes involved in these p-groups are labeled by the letter q.

Having fully characterized G/H, we turn to the study of H itself. Let N be the largest normal subgroup of G that also satisfies $N \subset H$ and $N \neq H$. If more than one subgroup satisfies the above requirements, then let N be any such subgroup. Because H is finite, there must be at least one

maximal subgroup.

We shall prove that $H/N = \mathbb{Z}_p^n$ for some prime p and integer n. The basic idea is that working modulo N, H/N is a normal subgroup of G/N. Furthermore H/N has no proper subgroups that are normal in G/N. In particular, this implies that H/N is abelian, because its commutator subgroup is a normal subgroup of G/N. Note that the possibility that the commutator subgroup of H/N is equal to H/N is excluded because H/N is solvable.

For any $x \in H/N$ consider $\mathcal{C}^{\#}_{G/N}(x)$, the group generated by the conjugates of x in G/N. This is a subgroup of H/N and is normal in G/N. Therefore

$$\mathcal{C}^{\#}_{G/N}(x) = H/N, \quad \forall x \in H/N, \tag{3.74}$$

which implies that all elements in H/N, with the exception of the identity, have the same order. That is because conjugates of x have the same order as x, and a product of elements of order k, in an abelian group, must have order less than or equal to k. This concludes the proof that $H/N = \mathbb{Z}_p^n$.

Thus far, we have the following tower of groups

$$N \subset H \subset HK_{q_i} \subset G, \tag{3.75}$$

where N, H and HK_{q_i} are all normal in G, and the group HK_{q_i} can be any of the groups found above.

Because $(G/N)^{((\infty))} = H/N = \mathbb{Z}_p^n$, the group G/N is also solvable and non-nilpotent. However, its structure is simpler than that of the full group G. We shall therefore be interested in working modulo N and shall denote groups modulo N by a tilde. That is

$$\tilde{G} = G/N, \quad \tilde{H}K_{q_i} = HK_{q_i}/N, \quad \tilde{H} = H/N = \mathbb{Z}_p^n. \tag{3.76}$$

The final step is to study the relationship between $\tilde{H}$ and the groups $\tilde{H}K_{q_i}$. By construction, we know $[\tilde{G}, \tilde{H}] = \tilde{H} = Z_p^n$, but what about $[\tilde{H}K_{q_i}, \tilde{H}]$? Because both $\tilde{H}K_{q_i}$ (for any i) and $\tilde{H}$ are normal in $\tilde{G}$, $[\tilde{H}K_{q_i}, \tilde{H}]$ is normal in $\tilde{G}$ and, furthermore, is contained in $\tilde{H}$. But N was defined to be the largest proper subgroup of H that was normal in $\tilde{G}$. Therefore $\tilde{H}$ has no proper subgroups that are normal in $\tilde{G}$, and $[\tilde{H}K_{q_i}, \tilde{H}]$ must be either the trivial group or all of $\tilde{H}$.

If $q_i = p$ then $\tilde{H}K_{q_i}$ is a p-group, and therefore nilpotent. This means that $[\tilde{H}K_p, \tilde{H}] \neq \tilde{H}$ and by the previous paragraph $[\tilde{H}K_p, \tilde{H}] = \{1\}$. The rest of the groups $\tilde{H}K_{q_i}$ can either commute or not with $\tilde{H}$. However, because $[\tilde{G}, \tilde{H}] = \tilde{H}$, at least one of them must not commute. Fix an i such that $[\tilde{H}K_{q_i}, \tilde{H}] = \tilde{H}$, and define $K = K_{q_i}$, $\tilde{H}K = \tilde{H}K_{q_i}$ and $q = q_i$. This will be the group to take the place of $\mathbb{Z}_q$.

We would like to show that there exists an element $b \in \tilde{H}K$, such that $[b, \tilde{H}] = \tilde{H}$. Let X be the

stabilizer of $\tilde{H}$ in $\tilde{H}K$, that is, the largest subgroup of $\tilde{H}K$ such that $[X, \tilde{H}] = 1$. Clearly $H \subset X$ and $X \neq \tilde{H}K$. Because $\tilde{H}K/X$ is nilpotent, it has a non-trivial center. Let $b \in \tilde{H}K$ be any element that projects modulo X to one of the non-trivial elements in the center. We will show that $[b, \tilde{H}]$ is normal in $\tilde{G}$, which implies $[b, \tilde{H}] = \tilde{H}$. The proof is that modulo X (which is normal in $\tilde{G}$), every element $g \in \tilde{G}$ commutes with b. Therefore $gbg^{-1} = bx$ for some $x \in X$ and

$$g\,[b,h]\,g^{-1} = bxh'x^{-1}b^{-1}h'^{-1} = bh'b^{-1}h'^{-1} \in \left[b, \tilde{H}\right], \tag{3.77}$$

for any $h \in \tilde{H}$, where $h' = ghg^{-1} \in \tilde{H}$.

To summarize, working modulo N, we have the following tower of subgroups:

$$\tilde{H} \subset \tilde{H}K \subset \tilde{G}, \tag{3.78}$$

with $\tilde{H} = Z_p^n$ for some prime p. Furthermore, $\tilde{H}K/\tilde{H} = K$ is a subgroup of order a power of q, for some prime q not equal to p. Finally, $\exists b \in \tilde{H}K$ such that $[b, \tilde{H}] = \tilde{H}$.

Note that this notation is consistent with the one used in Sec. 3.2. That is, if $G = \mathbb{Z}_p \times_\theta \mathbb{Z}_q$, then $N = \{1\}$, $H = \mathbb{Z}_p$, $K = \mathbb{Z}_q$, and the definitions of p, q and b would be consistent.

3.5.2 Examples

There are a few good examples to keep in mind that illustrate the potential new complications arising from groups with more structure than $\mathbb{Z}_p \times_\theta \mathbb{Z}_q$.

The first example is $A_4 = \mathbb{Z}_2^2 \times_\theta \mathbb{Z}_3$. The group can be described as $a_1 = (12)(34)$, $a_2 = (23)(41)$, $b = (123)$ with

$$\begin{aligned} a_1^2 = a_2^2 = 1, \qquad & a_1 a_2 = a_2 a_1, \\ b^3 = 1, \qquad & b a_1^i a_2^j b^{-1} = a_1^j a_2^{i+j}. \end{aligned} \tag{3.79}$$

For this group $N = \{1\}$, $H = \mathbb{Z}_2^2$ and $K = \mathbb{Z}_3$. Its most important feature is that $p = 2$, which was not previously possible. Because $p = 2$ implies working with qubits, these groups will be have to be handled specially.

The next example is $G = (\mathbb{Z}_3^2) \times_\theta (\mathbb{Z}_3 \times \mathbb{Z}_2)$. Let a_1, a_2 be the generators of $\mathbb{Z}_3^2$, let b be the generator of $\mathbb{Z}_2$ and let x be the generator of the remaining $\mathbb{Z}_3$. The semidirect product is defined by the conjugations

$$b a_1^i a_2^j b^{-1} = a_1^{-i} a_2^{-j}, \quad x a_1^i a_2^j x^{-1} = a_1^{-j} a_2^{i-j}. \tag{3.80}$$

For this group $H = \mathbb{Z}_3^2$ because $[G, G] = [G, H] = H$. The subgroup generated by $a_1 a_2^{-1}$ is normal

in G and therefore $N = \mathbb{Z}_3$. Finally $H/N = \mathbb{Z}_3$ and $K = \mathbb{Z}_2$. Note that x commutes with H modulo N, as discussed in the last section.

The final pair of examples illustrates the case where K is non-abelian. The examples are $\mathbb{Z}_3^2 \times_\theta Q$ and $\mathbb{Z}_3^2 \times_\theta D_4$. Labeling the generators of $\mathbb{Z}_3^2$ by a_1 and a_2, the semidirect product for $\mathbb{Z}_3^2 \times_\theta Q$ is defined by

$$ia_1^x a_2^y i^{-1} = a_1^y a_2^{-x}, \qquad ja_1^x a_2^y j^{-1} = a_1^{x+y} a_2^{x-y}, \tag{3.81}$$

where ± 1, $\pm i$, $\pm j$, $\pm k$ are the standard quaternionic elements. For $\mathbb{Z}_3^2 \times_\theta D_4$ the semidirect product is defined by

$$\beta a_1^x a_2^y \beta^{-1} = a_1^y a_2^{-x}, \qquad \gamma a_1^x a_2^y \gamma^{-1} = a_1^y a_2^x, \tag{3.82}$$

where the relations $\beta^4 = \gamma^2 = 1$ and $\gamma\beta\gamma = \beta^{-1}$ define D_4.

In both of the above cases $p = 3$, $q = 2$, $N = \{1\}$ and $H = \mathbb{Z}_3^2$. However, for $\mathbb{Z}_3^2 \times_\theta Q$ the non-trivial elements of H are conjugate to one another, and none of the non-trivial elements of Q commute with any of the non-trivial elements of H. The $\mathbb{Z}_3^2 \times_\theta D_4$ case divides H into three conjugacy classes (including the identity). Furthermore, each of the elements of the form $\beta^i\gamma$ commute with two non-trivial elements of H. These differences will become important when discussing the operations involving electric charges.

3.5.3 N-invariant ancillas

The first lesson from the above analysis is that we should work modulo N. That is, we want flux states labeled by elements of $\tilde{G} = G/N$, that are invariant under N. The idea of N-invariant states was already discussed in the previous chapter when generalizing simple non-abelian anyons to non-solvable ones, and therefore the discussion below will be brief.

A basis for the N-invariant magnetic fluxes is just $|g\rangle$ for $g \in \tilde{G}$. The braiding and fusion properties of these states behave almost exactly as if the full group were $\tilde{G}$ and these states were flux eigenstates. The only difference is that when fusing two anyons from pairs with opposite fluxes, the probability of disappearing into the vacuum is lower.

Even producing anyons from the vacuum behaves correctly with respect to N-invariance. Pairs produced from the vacuum are naturally invariant under the full group G. Normally, when braiding with other states, this invariance will be broken. However, if the vacuum pair only interacts with N-invariant states, then the invariance under the group N will remain.

At this point we will change our requirements for the physical system. Instead of requiring a reservoir of flux ancillas for every element of G, we will require a reservoir of N-invariant flux ancillas

for every element of $\tilde{G}$. This is likely a reasonable modification, as it appears that the latter ancillas are no harder to produce than the original ones.

It should be noted that, when working modulo N, the electric charges need no modification. That is, because N is normal in G, any representation of $\tilde{G}$ extends to a representation of G that is invariant under N. Furthermore, fusing two N-invariant electric charges must produce a new N-invariant electric charge. Therefore, working with N-invariant electric charges simply involves working with a subset of the charges of the group G.

Given the above caveats, we can effectively replace the group G with the group $\tilde{G} = G/N$, which will be done without further comment for the rest of this section.

3.5.4 Computational basis

We will begin by defining an extended computational basis and will discuss the operations that can be performed on this extended subspace. Toward the end of this section, a subset of these states will be singled out as the true computational basis.

Let $a_1, \ldots, a_n$ be a set of generators for $\tilde{H} = \mathbb{Z}_p^n$, and recall the definition of the element $b \in \tilde{G}$. The extended computational basis consists of the states

$$|i_1, \ldots, i_n\rangle \equiv |a_1^{i_1} \cdots a_n^{i_n} b a_n^{-i_n} \cdots a_1^{-i_1}\rangle, \tag{3.83}$$

where each of the i's takes values from 0 to $p-1$.

To prove that the states are all distinct consider the map from $H \to H$ defined by

$$[g, \cdot] : h \to [g, h]\,. \tag{3.84}$$

Because $\tilde{H}$ is abelian, this map is an homomorphism for any $g \in \tilde{G}$. In particular, since $[b, \tilde{H}] = \tilde{H}$, the homomorphism defined by $[b, \cdot]$ is surjective and has trivial kernel. That is, no element of $\tilde{H}$ commutes with b. But

$$hbh^{-1} = h'bh'^{-1} \Rightarrow (h'^{-1}h)b(h'^{-1}h)^{-1} = b, \tag{3.85}$$

for any elements $h, h' \in \tilde{H}$, which can only be true if $h = h'$.

3.5.5 Basic operations

The generalized controlled-X is the transformation

$$|i_1, \ldots, i_n\rangle \otimes |j_1, \ldots, j_n\rangle \longrightarrow |i_1, \ldots, i_n\rangle \otimes |i_1 + j_1, \ldots, i_n + j_n\rangle. \tag{3.86}$$

It can be implemented as a conjugation of the second anyon by a function of the flux of the first

anyon such that

$$f(hbh^{-1}) = h, \tag{3.87}$$

for any $h \in \tilde{H}$. Because the map $[b, \cdot]$ defined above is just a permutation of the elements of $\tilde{H}$, it has a finite period (say l). The desired function is

$$f(g) = \Big[\Big[\big[gb^{-1}, b\big], b\Big] \cdots, b\Big], \tag{3.88}$$

which consists of $l-1$ nested commutators. The final commutator needed to complete the period is the one formed in the expression gb^{-1} when g has the form hbh^{-1}.

At this point, one may wonder how does working modulo a normal subgroup N affect the discussion regarding the computability of the controlled-X gate. The controlled-X can only be implemented because $\tilde{G}$ is non-nilpotent. In a sense, $\tilde{G}$ was constructed to be as small as possible, but still maintain the property of being non-nilpotent. On the other hand, if a group G is nilpotent to begin with, then any subgroup or quotient group will also be nilpotent, and no controlled-X can be constructed using braiding.

Using the same techniques as in Sec. 3.2.4, anyon fusions can be used to perform measurements. Fusion with $|b^{-1}\rangle$ ancillas produces a probabilistic projection onto $|0, \ldots, 0\rangle$. Fusing the two anyons that form a qudit is a probabilistic projection onto

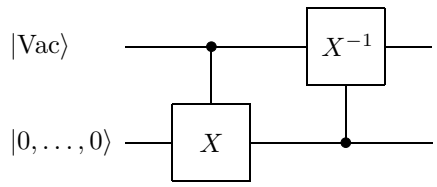
$$|\tilde{0}, \ldots, \tilde{0}\rangle \equiv \frac{1}{\sqrt{p^n}} \sum_{i_1=0}^{p-1} \cdots \sum_{i_n=0}^{p-1} |i_1, \ldots, i_n\rangle. \tag{3.89}$$

As usual, to complete the probabilistic projection, these fusions must be supplemented by a reservoir of $|0, \ldots, 0\rangle$ and $|\tilde{0}, \ldots, \tilde{0}\rangle$ ancillas. The first case is trivial, because the existence of these ancillas has been assumed as one of the physical requirements of the system. The production of $|\tilde{0}, \ldots, \tilde{0}\rangle$ ancillas is more complicated and will occupy the rest of the subsection.

The procedure to distill $|\tilde{0}, \ldots, \tilde{0}\rangle$ states begins with a pair created from the vacuum and a $|0, \ldots, 0\rangle$ ancilla:

$$|\text{Vac}\rangle \otimes |0, \ldots, 0\rangle. \tag{3.90}$$

Using only braiding, an incomplete swap is applied to the state:



Once again, the circuit denotes the action of the conjugations on the computational basis, but their extension to the full Hilbert space needs to be discussed. After applying the necessary braidings to perform the circuit, the top state is fused with a $|b^{-1}\rangle$ ancilla. If the fusion does not produce the vacuum state, the final product is discarded and the procedure restarted from the beginning. Since conjugations cannot change the superselection sector, the only case that needs to be considered is when the vacuum state is created in the superselection sector that contains the computational subspace (i.e., the conjugacy class of b). In this superselection sector the vacuum state has the form

$$|\text{Vac}\rangle \propto |\tilde{0},\ldots,\tilde{0}\rangle + |\Psi^{\perp}\rangle, \tag{3.91}$$

where $|\Psi^{\perp}\rangle$ is a state in the space spanned by vectors of the form $|gbg^{-1}\rangle$ that are not contained in the computational basis.

Because we want to guarantee that after the controlled-X the state $|0,\ldots,0\rangle$ remains in the computational subspace, we need the conjugation function to satisfy

$$\begin{aligned} f(\tilde{G}) &\in \tilde{H}, \quad \text{and} \\ f(hbh^{-1}) &= h \quad \forall h \in \tilde{H}. \end{aligned} \tag{3.92}$$

The second requirement can be satisfied by choosing f as a sequence of commutators as in Eq. (3.88), as long as the number of commutators is one minus a multiple of l (the period of $[b,\cdot]$). Furthermore, the result after i commutators must be contained in $\tilde{G}^{((i))}$. Because the series is finite, $\tilde{G}^{((j))} = \tilde{G}^{((\infty))} = \tilde{H}$ for some finite j, and the first requirement can also be satisfied by defining f to be a long enough sequence of commutators. Both requirements can be satisfied simultaneously by correctly choosing the number of commutators in the expression, and this completes the definition of the first controlled-X.

The second controlled-X can be a regular controlled-X because in this case the control is known to be in the computational subspace. In the end, the vacuum state will be conjugated by an element of $\tilde{H}$, and therefore can only have flux b if it was originally in the computational subspace.

Having completed the construction of the $|\tilde{0},\ldots,\tilde{0}\rangle$ ancillas, all that is required to complete a universal set of gates is an analog of the probabilistic projection onto $|0\rangle^{\perp}$ constructed out of fusions of electric charges.

3.5.6 Using electric charges

The ideal goal for this section would be the construction of the probabilistic projection onto the state $|0,\ldots,0\rangle^{\perp}$. Unfortunately, this is not possible for most groups. However, we will produce a pair of gates that have an equivalent computational power.

The first gate involves a non-trivial subgroup $\tilde{\Lambda} \subset \tilde{H}$, to be defined later, which could equal all of $\tilde{H}$. Note that this subgroup defines a subspace of the computational space spanned by

$$|\lambda b \lambda^{-1}\rangle, \tag{3.93}$$

for all elements $\lambda \in \tilde{\Lambda}$, which will also be denoted by $\tilde{\Lambda}$. The probabilistic projection onto $\tilde{\Lambda}$ will be the first gate.

The second gate is the probabilistic projection onto $|0,\ldots,0\rangle^{\perp} \cap \tilde{\Lambda}$. This second gate can be thought of as an application of the first gate, followed by a probabilistic projection onto $|0\rangle^{\perp}$ that only works on states contained in $\tilde{\Lambda}$. For the moment, we will assume that the first gate can be implemented and will work on the construction of the second gate.

The basic building block for this section involves working with the state to be measured $|\Psi\rangle$ and an electric charge pair in the vacuum state $|R(I)\rangle_R$ of some non-abelian representation R. The state to be measured is contained in the computational basis and can therefore be expanded as

$$|\Psi\rangle = \sum_{h\in\tilde{H}} \psi_h |hbh^{-1}\rangle, \tag{3.94}$$

where, as in Sec. 3.2.7, the coefficients $\{\psi_h\}$ could be numbers or could denote the state of the rest of the system.

Using braiding, the state $|\Psi\rangle$ can be entangled with the electric charges. In particular, if $\phi(g)$ is a function constructed as a product of g and fixed elements of $\tilde{G}$, then the following transformation can be realized:

$$|\Psi\rangle \otimes |R(I)\rangle_R \longrightarrow \sum_{h\in\tilde{H}} \psi_h |hbh^{-1}\rangle \otimes |R(\phi(h))\rangle_R. \tag{3.95}$$

Note that the state of the electric charge can depend on $\phi(h)$ rather than $\phi(hbh^{-1})$ by composing with the function defined in Eq. (3.88). That is $\phi(f(hbh^{-1})) = \phi(h)$.

Now the electric charge pair is fused together, and the resulting particle is measured. More specifically, in accordance with the discussion in Sec. 3.2.8, we just check whether the resulting particle belongs to some one-dimensional representation labeled γ. If the charge γ is detected, then the electric charge will have disentangled with the state being measured because its internal Hilbert space is one dimensional. Furthermore, because each one-dimensional representation occurs only

once in the decomposition of $R \otimes R^*$, the state will be unentangled with the environment as well. The proof of the latter property uses Schur's lemma and the fact that if $|M_1\rangle_R$ and $|M_2\rangle_R$ always fuse into representation γ then $|M_1 M_2^\dagger\rangle_R$ will always fuse into the vacuum.

The result of the complete operation, when the outcome γ is obtained, is the transformation

$$|\Psi\rangle \longrightarrow \sum_{h \in \tilde{H}} F_{\phi(h)\to\gamma} \psi_h |hbh^{-1}\rangle, \tag{3.96}$$

where the state after the measurement has been left unnormalized. The coefficients $F_{h\to\gamma}$ depend implicitly on the original representation R and will be defined carefully below.

The above procedure can be repeated many times for different functions $\phi(g)$. If on each occurrence the outcome γ is obtained, the resulting (unnormalized) state will be

$$\sum_{h \in \tilde{H}} \left(\prod_{\phi \in \Phi} F_{\phi(h)\to\gamma} \right) \psi_h |hbh^{-1}\rangle, \tag{3.97}$$

where Φ is the set of functions used. As usual, if the outcome γ is not obtained on each instance, the state is discarded, and the probabilistic projection reports a projection onto the complement.

We assume that all functions in the set Φ are products of conjugates of the input, and therefore $\phi(I) = I$ for any $\phi \in \Phi$. Because $|R(I)\rangle_R$ is the vacuum state, it will always fuse back into the vacuum. Therefore, if γ is a non-trivial representation then $F_{I\to\gamma} = 0$ and the above operation projects out the $|0, \ldots, 0\rangle$ state.

At this point we have almost constructed a probabilistic projection onto $|0, \ldots, 0\rangle^\perp \cap \tilde{\Lambda}$. The states outside of $\tilde{\Lambda}$ can be removed using the probabilistic projection onto $\tilde{\Lambda}$, which for the moment we assume can be implemented. Therefore, the desired gate will be complete if the coefficients

$$\prod_{\phi \in \Phi} F_{\phi(\lambda)\to\gamma} \tag{3.98}$$

are non-zero and equal for every non-trivial $\lambda \in \tilde{\Lambda}$. The requirement of equality is accomplished if the orbits under the functions in Φ, of all non-trivial $\lambda \in \tilde{\Lambda}$, are equal.

More specifically, let Φ be a set of maps from $\tilde{\Lambda}$ to $\tilde{\Lambda}$ that fix the identity. We say that Φ is balanced on $\tilde{\Lambda}$ if it satisfies the relation

$$\#(\lambda_1 \to \lambda') = \#(\lambda_2 \to \lambda') \quad \forall\ \lambda_1, \lambda_2, \lambda' \in \tilde{\Lambda} - \{I\}, \tag{3.99}$$

where $\#(\lambda \to \lambda')$ denotes the number of elements $\phi \in \Phi$ such that $\phi(\lambda) = \lambda'$. The requirement that Φ be balanced guarantees that the expressions in Eq. (3.98) are equal for every λ. Of course, for the coefficients to be non-zero, we must prove separately that the value of $F_{\lambda\to\gamma}$ is non-zero for every

non-trivial $\lambda \in \tilde{\Lambda}$.

The goal for the rest of this section is, therefore, to find: a subgroup $\tilde{\Lambda}$ of $\tilde{H}$, an irreducible representation R of $\tilde{G}$, and a one-dimensional representation γ of $\tilde{G}$ such that:

1. The probabilistic projection onto $\tilde{\Lambda}$ can be implemented.

2. $F_{\lambda\to\gamma} \neq 0$ for every non-trivial $\lambda \in \tilde{\Lambda}$.

3. There exists a set of maps from $\tilde{\Lambda}$ to $\tilde{\Lambda}$, that is balanced on $\tilde{\Lambda}$, and can be expressed as

$$\phi(g) = \prod_i g_i g g_i^{-1}, \tag{3.100}$$

   for some elements $\{g_i\} \in \tilde{G}$.

3.5.6.1 Choosing $\tilde{\Lambda}$

There are groups, such as $\mathbb{Z}_3^2 \times_\theta D_4$, for which there is no choice of R and non-trivial γ such that $F_{h\to\gamma} \neq 0$ for all non-trivial $h \in \tilde{H}$. It is therefore advantageous to choose $\tilde{\Lambda}$ as small as possible. Furthermore, a small $\tilde{\Lambda}$ will also help when proving the existence of a set of functions balanced on $\tilde{\Lambda}$.

Let a be a non-trivial element of $\tilde{H}$, and consider the set of functions of the form

$$\phi(g) = \prod_i g_i g g_i^{-1}, \tag{3.101}$$

such that $\phi(a) = I$. The kernel of each of these functions is a subgroup of $\tilde{H}$ that contains the element a. We define $\tilde{\Lambda}$ as the intersection of all these kernels.

Because there is a finite set of maps from $\tilde{H}$ to $\tilde{H}$, we can find a finite set of functions $\{\phi_i\}$, in the form of Eq. (3.101), satisfying

$$\begin{aligned} \lambda \in \tilde{\Lambda} \quad &\Longrightarrow \quad \forall i \;\; \phi_i(\lambda) = I, \\ h \notin \tilde{\Lambda} \quad &\Longrightarrow \quad \exists i \;\; \phi_i(h) \neq I. \end{aligned} \tag{3.102}$$

A probabilistic projection onto $\tilde{\Lambda}$ can be constructed using controlled conjugations on an ancilla $|b\rangle$:

$$\sum_{h\in\tilde{H}} \alpha_h |hbh^{-1}\rangle \otimes |b\rangle \longrightarrow \sum_{h\in\tilde{H}} \alpha_h |hbh^{-1}\rangle \otimes |\phi_i(h) b \phi_i(h)^{-1}\rangle, \tag{3.103}$$

and then using fusion to make sure that the ancilla remains in the $|b\rangle$ state. Repeating the procedure for each ϕ_i produces the desired projection.

To build the set of functions that are balanced on $\tilde{\Lambda}$, let Φ be the set of functions in the form of Eq. (3.101) such that $\phi(a) \in \tilde{\Lambda} - \{I\}$. We shall prove that this is the desired set of functions.

Let $\lambda \in \tilde{\Lambda}$ be non-trivial and let ϕ be any map in Φ. The value of $\phi(\lambda)$ must be non-trivial and contained in $\tilde{\Lambda}$. Otherwise, it would be possible to construct a map in product form such that a is in its kernel but λ is not, contrary to the definition of $\tilde{\Lambda}$. In fact, the functions in Φ are just automorphisms of $\tilde{\Lambda}$ and form a group with multiplication given by function composition. Furthermore, because $\mathcal{C}^{\#}_{\tilde{G}}(a) = \tilde{H}$, for any non-trivial $\lambda \in \tilde{\Lambda}$ there exists a function $\phi_{a\to\lambda} \in \Phi$ such that $\phi_{a\to\lambda}(a) = \lambda$. If $\lambda' \in \tilde{\Lambda}$ is a third non-trivial element, then for every function $\phi \in \Phi$ such that $\phi(\lambda) = \lambda'$ there is a function $\phi'(a) = \lambda'$ given by $\phi' = \phi \circ \phi_{a\to\lambda}$. Therefore, Φ is balanced on $\tilde{\Lambda}$.

3.5.6.2 The amplitudes $F_{h\to\gamma}$

To choose R and γ we first need to examine and define $F_{h\to\gamma}$ more carefully. Since we are mostly interested in whether $F_{h\to\gamma}$ is zero or non-zero, we will generally work with its magnitude squared, which has the simple expression

$$|F_{h\to\gamma}|^2 = \left| P_\gamma |R\,(h)\rangle_R \right|^2, \tag{3.104}$$

where P_γ is the projector onto the space that will turn into the representation γ after fusion. This subspace is just the subspace that transforms as γ under conjugation. It can be projected out using the orthogonality of characters (and matrix entries for non-abelian representations):

$$P_\gamma|\Psi\rangle = \frac{1}{|\tilde{G}|}\sum_{g\in\tilde{G}} \bar{\gamma}^g\; U(g)\otimes U(g)|\Psi\rangle, \tag{3.105}$$

where $\bar{\gamma}$ is the conjugate representation. Note that the values of the representation γ on $g \in \tilde{G}$ will be denoted by γ^g, as a reminder that it is always a power of some root of unity that we shall also denote by γ.

Combining the expressions for the projector and the electric charge state we obtain

$$\begin{aligned} |F_{h\to\gamma}|^2 &= \left| \frac{1}{|\tilde{G}|}\sum_{g\in\tilde{G}} \bar{\gamma}^g |R\left(ghg^{-1}\right)\rangle_R \right|^2 \\ &= \frac{1}{d_R|\tilde{G}|^2}\left| \sum_{g\in\tilde{G}} \bar{\gamma}^g R\left(ghg^{-1}\right) \right|^2, \end{aligned} \tag{3.106}$$

where d_R is the dimension of representation R. In the second line, the magnitude squared of the matrix is given by $|M|^2 = \mathrm{Tr}\, MM^\dagger$, which is equivalent to the sum of the magnitude squared of the entries of the matrix.

Because $\tilde{H}$ is abelian, the representation R can be diagonalized on $\tilde{H}$ so that the diagonal

entries are one-dimensional representations of $\tilde{H}$. These representations can be labeled by an index i running along the diagonal of the matrices R, and described by functions $\omega_i^h : \tilde{H} \to \mathbb{C}$. With the new notation:

$$|F_{h\to\gamma}|^2 = \frac{1}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}}\bar{\gamma}^g\,\omega_i^{ghg^{-1}}\right|^2, \tag{3.107}$$

where the representation R is now implicit in the definition of the representations $\{\omega_i\}$.

Finally, let $\tilde{S}$ be the stabilizer of $\tilde{H}$ in $\tilde{G}$, that is, the subgroup of G that commutes with every element of $\tilde{H}$. Clearly, it is a normal subgroup of $\tilde{G}$, and $\tilde{H} \subset \tilde{S}$. Furthermore, we had argued that if $q_i = p$ then $K_{q_i} \in \tilde{S}$. Therefore $|\tilde{G}/\tilde{S}|$ is not divisible by p.

Since the function $F_{h\to\gamma}$ will be zero unless we choose a representation such that $\gamma^{\tilde{S}} = 1$, we shall assume this from now on, and write

$$|F_{h\to\gamma}|^2 = \frac{|\tilde{S}|^2}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}/\tilde{S}}\bar{\gamma}^g\,\omega_i^{ghg^{-1}}\right|^2. \tag{3.108}$$

We are now guaranteed that γ corresponds to powers of an n^{th} root of unity such that p does not divide n. The terms in the above expression have the form

$$\sum_{i=0}^{p-1} c_i\omega^i, \tag{3.109}$$

where the coefficients c_i are sums of n^{th} roots of unity. By [Sch64], the expression will be zero if and only if the n coefficients c_i are all equal.

Using the above notation it is easy to show two properties of the amplitudes $F_{h\to\gamma}$. If $|F_{h\to\gamma}| \neq 0$ then

$$\left|F_{h^j\to\gamma}\right|^2 = \frac{|\tilde{S}|^2}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}/\tilde{S}}\bar{\gamma}^g\,\left(\omega_i^{ghg^{-1}}\right)^j\right|^2 \neq 0, \tag{3.110}$$

as long as p does not divide j. Note that in general $|F_{h\to\gamma}| \neq |F_{h^j\to\gamma}|$. The fact that was used above is that $|F_{h\to\gamma}| \neq 0$ implies that at least two coefficients of different powers of ω must be different. Replacing ω by a power of itself just permutes the coefficients c_i in Eq. (3.109).

The second property is easier to prove in the form of Eq. (3.107) and states that given $|F_{h\to\gamma}| \neq 0$

then

$$\begin{aligned} |F_{xhx\to\gamma}|^2 &= \frac{1}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}}\bar{\gamma}^g\,\omega_i^{gxhxg^{-1}}\right|^2 \\ &= \frac{1}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}}\bar{\gamma}^{gx^{-1}}\,\omega_i^{ghg^{-1}}\right|^2 \\ &= |F_{h\to\gamma}|^2 \neq 0, \end{aligned} \tag{3.111}$$

for any $x \in \tilde{G}$. The second line involves a relabeling of the summation variable, whereas the third line is true because γ is a group homomorphism and $\bar{\gamma}^{-x}$ is just an overall phase.

Together, the two properties imply that, if $|F_{h\to\gamma}|^2$ is non-zero, then so are the amplitudes $|F_{h'\to\gamma}|^2$ for any non-trivial $h' = gh^ig^{-1}$. Unfortunately, even after adding the identity element, this set is in general not a group. Furthermore, it remains to be shown that the amplitude is non-zero for at least one h.

3.5.6.3 Finding a non-zero amplitude

It is possible to indirectly show that, for every element $h \in \tilde{H}$, there is a pair of representations R and γ meeting our requirements, such that $|F_{h\to\gamma}|^2 \neq 0$.

The basic idea is to consider the regular representation of $\tilde{G}$. Let $\mathcal{H}_{\tilde{G}}$ be the Hilbert space spanned by the vectors

$$|g\rangle_{\tilde{G}} \tag{3.112}$$

for $g \in \tilde{G}$. For the moment, these are just abstract vectors in a Hilbert space, and therefore we use the above notation to distinguish them from the anyon magnetic charges.

The group $\tilde{G}$ has both a left and a right action on this vector space, which transforms as the regular representation in both cases. More generally, we could say that there is an action of the group $\tilde{G}\times\tilde{G}$ on this vector space given by

$$|g\rangle_{\tilde{G}} \longrightarrow |g_1 g g_2^{-1}\rangle_{\tilde{G}} \tag{3.113}$$

for any element $g_1 \times g_2 \in \tilde{G}\times\tilde{G}$.

Let $\mathcal{H}_R$ be the Hilbert space spanned by the vectors of the form $|M\rangle_R$, where R in an irreducible representation of $\tilde{G}$. These spaces are also representations of $\tilde{G}\times\tilde{G}$ and, in fact, are irreducible. The space $\mathcal{H}_{\tilde{G}}$ decomposes as a sum of irreducible representations of $\tilde{G}\times\tilde{G}$ as

$$\mathcal{H}_{\tilde{G}} = \bigoplus_R \mathcal{H}_R \tag{3.114}$$

with each irreducible representation R appearing exactly once. Fusion corresponds to a further decomposition into the irreducible representations of the diagonal group $\tilde{G}$.

The state $|I\rangle_{\tilde{G}}$ transforms as the identity under the diagonal group and can therefore be written as a sum of states $|R(I)\rangle_R$ for different representations R. Hence a state $|h\rangle_{\tilde{G}}$ can be written as a sum of states $|R(h)\rangle_R$. If the state $|h\rangle_{\tilde{G}}$ has a non-zero projection to a representation γ of the diagonal group, then we know that $|F_{h\to\gamma}| \neq 0$ for at least one irreducible representation R.

More explicitly, the projection is

$$P_\gamma |h\rangle_{\tilde{G}} = \frac{1}{|\tilde{G}|} \sum_{g\in\tilde{G}} \bar{\gamma}^g |ghg^{-1}\rangle_{\tilde{G}}. \tag{3.115}$$

To make it non-zero, it is sufficient to choose γ to be constant over the stabilizer, S_h, in $\tilde{G}$ of h. This is still possible, even with our requirements that γ be one-dimensional and non-trivial, because $S_h/\tilde{H}$ is a proper subgroup of the nilpotent group $\tilde{G}/\tilde{H}$. Proper subgroups of nilpotent groups are always contained in proper normal subgroups because the normalizer of the proper subgroup is always a larger group (and eventually the operation of replacing a subgroup with its normalizer must yield a normal subgroup). This concludes the proof that, for any non-trivial $h \in \tilde{H}$, there exists a choice of γ and R such that $|F_{h\to\gamma}| \neq 0$.

In fact, for any two non-trivial elements $\lambda_1, \lambda_2 \in \tilde{\Lambda}$, the same representation γ is useful because $S_{\lambda_1} = S_{\lambda_2}$. However, it is not clear that it is possible to pick R such that both $|F_{\lambda_1\to\gamma}| \neq 0$ and $|F_{\lambda_2\to\gamma}| \neq 0$. This is illustrated by working with the group $\mathbb{Z}_5^2 \times_\theta (\mathbb{Z}_2 \times \mathbb{Z}_3)$, where certain choices of γ consistent with the above discussion lead to zero amplitudes for at least one non-trivial element of $\tilde{\Lambda}$, no matter which R is used. On the other hand, the same example does have simultaneous choices of R and γ that satisfy all our requirements. It is unclear to the author whether it is possible, for any group $\tilde{G}$, to choose R and γ such that $|F_{\lambda\to\gamma}| \neq 0$ for all non-trivial elements $\lambda \in \tilde{\Lambda}$ simultaneously.

3.5.6.4 Alternative $\tilde{\Lambda}$

What happens if R and γ cannot be chosen so that $|F_{\lambda\to\gamma}| \neq 0$ over all non-trivial elements $\lambda \in \tilde{\Lambda}$? While none of the examples in this chapter have this problem, if such a case arises, we could try to shrink $\tilde{\Lambda}$. In particular, if $\tilde{\Lambda} = \mathbb{Z}_p$, then the problem is solved. That is, because we can always choose the representations so that the amplitude is non-zero for some element, and then it is guaranteed to be non-zero for the powers of that element as well.

The set of functions balanced on $\tilde{\Lambda} = \mathbb{Z}_p$ can be easily constructed as simply $\phi(g) = g^i$ for $0 < i < p$. However, the probabilistic projection onto $\tilde{\Lambda}$ is more difficult. It can be achieved if we are willing to relax the error model of the probabilistic projections. That is, we use an approximate probabilistic projection, where the probabilities and projected states are close to the desired results. While the results will be exponentially close in the number of successful fusions, they will only be

polynomially close in the number of actual fusions, and therefore the machinery of fault tolerant quantum computation must be employed. Computation with the approximate gate will still be feasible, but one of the advantages of topological quantum computation, that is, the exactness of gates, will be lost.

To construct this approximate projection, consider the amplitude for the fusion of the electric charges into the vacuum, denoted by $F_{h\to I}$. It is the same quantity that has been dealt with thus far, only with the representation γ replaced by the identity representation. These quantities have the expression

$$\begin{aligned} |F_{h\to I}|^2 &= \frac{1}{d_R|\tilde{G}|^2}\sum_{i=1}^{d_R}\left|\sum_{g\in\tilde{G}}\omega_i^{ghg^{-1}}\right|^2 \\ &= \frac{1}{d_R|\mathcal{C}_{\tilde{G}}(h)|^2}\sum_{i=1}^{d_R}\left|\sum_{h'\in\mathcal{C}_{\tilde{G}}(h)}\omega_i^{h'}\right|^2, \end{aligned} \tag{3.116}$$

where $\mathcal{C}_{\tilde{G}}(h)$ is the conjugacy class of h in $\tilde{G}$. The amplitudes satisfy the properties

$$0 < |F_{h\to I}|^2 < |F_{I\to I}|^2 \tag{3.117}$$

for any non-trivial $h \in \tilde{H}$. The first inequality comes from the fact that we are summing p^{th} roots of unity and the number of summands is not divisible by p. The second inequality comes from the fact that $\omega_i^{h'}$ must be non-constant over the conjugacy class of h. The equation

$$I = \prod_{h'\in\mathcal{C}_{\tilde{G}}(h)} h', \tag{3.118}$$

is true because the right-hand side commutes with all of $\tilde{G}$, and therefore must be the identity. Because the number of factors on the right is not divisible by p, ω_i cannot be constant over the conjugacy class unless it is the identity. Furthermore, since the conjugacy class generates $\tilde{H}$, and R is non-trivial, one of the ω_i must not be the identity. This proves the second inequality of Eq. (3.117).

The standard procedure of entangling a state with an electric charge pair, which is then fused, can then be used. The state is now kept if the pair fuses into the vacuum, which always has a non-zero probability of occurring. The basis state that was entangled with $|R(I)\rangle_R$ will have its amplitude increased relative to the other basis states. Using braiding to achieve a function of the form $f(h) = ha^i$, for some element $a \in \tilde{\Lambda}$ and different values of i, we can make the basis states in $\tilde{\Lambda}$ consisting of powers of a have an arbitrarily large amplitude relative to the other states. Even if $|F_{h\to I}|$ varies significantly over the non-trivial elements of $\tilde{H}$, we can use the old $\tilde{\Lambda}$ projector and functions in Φ to balance out the non-trivial elements while increasing the amplitude of the state with $f(h) = I$. After many repetitions, the basis states with flux $a^i b a^{-i}$ can be made to have an

amplitude much larger then all the other states. This completes the construction of the approximate probabilistic projection onto the new $\tilde{\Lambda}$ for the special cases when we require $\tilde{\Lambda} = \mathbb{Z}_p$.

3.5.7 Putting it all together

At this point we have shown the existence of an extended computational space, with elements labeled by $\tilde{H} = \mathbb{Z}_p^n$, on which we can perform the generalized controlled-X, and probabilistic projections onto $|0, \ldots, 0\rangle$ and $|\tilde{0}, \ldots, \tilde{0}\rangle$. Furthermore, there exists a non-trivial subgroup $\tilde{\Lambda} \subset \tilde{H}$, such that we can implement probabilistic projections onto $\tilde{\Lambda}$ and $|0, \ldots, 0\rangle^{\perp} \cap \tilde{\Lambda}$.

To define the real computational subspace, choose a non-trivial element $a \in \tilde{\Lambda}$, and define

$$|i\rangle \equiv |a^i b a^{-i}\rangle, \tag{3.119}$$

for $0 \leq i < p$. This subspace corresponds to the subgroup $\{a^i\} \subset \tilde{\Lambda}$ of powers of a.

A probabilistic projection onto the real computational space, corresponding to $\{a^i\}$, can be achieved in two steps. The first step is to apply the probabilistic projection onto $\tilde{\Lambda}$. The second step is repeated for each $\lambda \in \tilde{\Lambda}$ that is not in $\{a^i\}$. For fixed λ, we use an ancilla to conjugate by λ^{-1}, then do the probabilistic projection onto $|0, \ldots, 0\rangle^{\perp} \cap \tilde{\Lambda}$ and then conjugate by λ using another ancilla:

$$\begin{aligned} \sum_{x \in \tilde{\Lambda}} \alpha_x |x b x^{-1}\rangle &\longrightarrow \sum_{x \in \tilde{\Lambda}} \alpha_x |\lambda x b x^{-1} \lambda^{-1}\rangle \\ &\longrightarrow C \sum_{x \in \tilde{\Lambda}, x \neq \lambda} \alpha_x |\lambda x b x^{-1} \lambda^{-1}\rangle \\ &\longrightarrow C \sum_{x \in \tilde{\Lambda}, x \neq \lambda} \alpha_x |x b x^{-1}\rangle, \end{aligned} \tag{3.120}$$

where the probabilistic projection was assumed to succeed in the second step, and therefore the state is renormalized by the constant C. The net effect of one such operation is to project out the state $|\lambda b \lambda^{-1}\rangle$. If all the projections succeed, then we will have projected the original state into the computational basis, completing the probabilistic projection onto $\{a^i\}$.

For the case of qudits with $d = p > 2$ we are now done. The generalized controlled-X behaves as a controlled-X when restricted to act on the computational space. A probabilistic projection onto $|0\rangle$ is just the probabilistic projection onto $|0, \ldots, 0\rangle$ because $|0, \ldots, 0\rangle = |0\rangle$. The probabilistic projection onto $|\tilde{0}, \ldots, \tilde{0}\rangle$ behaves as a probabilistic projection onto $|\tilde{0}\rangle$ because

$$\langle \tilde{i} | \tilde{0}, \ldots, \tilde{0}\rangle \propto \delta_{i,0} \tag{3.121}$$

with the caveat that we must use the projection onto the computational basis to turn the $|\tilde{0}, \ldots, \tilde{0}\rangle$

ancillas into $|\tilde{0}\rangle$ ancillas. Finally, the probabilistic projection onto $|0,\ldots,0\rangle^{\perp} \cap \tilde{\Lambda}$ reduces to a probabilistic projection onto $|0\rangle^{\perp}$ when acting on states in the computational subspace. These are the gates that were proven universal for quantum computation in Sec. 3.3.

3.5.7.1 The case $p = 2$

Special treatment must be given to the case when $p = 2$, that is, when working with qubits. Though all the gates constructed above are valid for $p = 2$, the gate-set is not universal. The problem is that the probabilistic projection onto $|0\rangle^{\perp} = |1\rangle$ does not provide any additional computational power beyond the probabilistic projection onto $|0\rangle$.

Just as in Sec. 3.3.2, the gate-set can be made universal given a supply of the magic states:

$$\begin{aligned} |\phi_{M1}\rangle &= \frac{1}{2}\sum_{i,j} |i\rangle \otimes |j\rangle \otimes |ij\rangle, \\ |\phi_{M2}\rangle &= \frac{1}{2}\sum_{i,j} \omega^{\delta_{i,1}\delta_{j,1}} |i\rangle \otimes |j\rangle, \end{aligned} \tag{3.122}$$

where the second state can be produced from the first one by measuring the third qudit in the X basis.

The production of the magic state $|\phi_{M1}\rangle$ is the step that requires a projection constructed from the fusion of electric charges. Given our choice of $a \in \tilde{\Lambda}$ above, assume that $bab^{-1} \in \tilde{\Lambda}$. This must be the case if $\tilde{\Lambda}$ was defined as the intersection of kernels of functions. Clearly, we can apply a controlled conjugation by a, and therefore, additionally, the controlled conjugation by bab^{-1} and by $abab^{-1}$. Note that $a \neq bab^{-1}$ because b was chosen to not commute with a.

We begin with the $|\tilde{0}\rangle \otimes |\tilde{0}\rangle \otimes |\tilde{0}\rangle$ state and append a $|0\rangle = |b\rangle$ ancilla. We then conjugate it to obtain

$$\frac{1}{\sqrt{8}} \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} |i\rangle \otimes |j\rangle \otimes |k\rangle \otimes |f_{i,j,k} b f_{i,j,k}^{-1}\rangle, \tag{3.123}$$

where

$$f_{i,j,k} = a^{1-i} \left(bab^{-1}\right)^{1-j} x^k, \tag{3.124}$$

with x to be determined in a moment. A probabilistic projection onto $|0,\ldots,0\rangle^{\perp}$ is then applied to the last ancilla, and the conjugations are undone.

If the projection succeeds we will have projected out two of the initial eight basis states, depending on the value of $x \in \{a, bab^{-1}, abab^{-1}\}$. In all cases, the state $|1\rangle \otimes |1\rangle \otimes |0\rangle$ is removed, and for each of the three values of x, one of the other undesirable basis states is removed. Repeating the procedure once for each value of x produces the desired magic state $|\phi_{M1}\rangle$. Note that the above

procedure succeeds because $a^2 = 1$, and a commutes with bab^{-1}.

What happens if bab^{-1} is not in $\tilde{\Lambda}$? This is the case when fusions of electric charges into the vacuum must be used. In particular, instead of projecting out the undesirable basis states, we increase the amplitude of the desired basis states and obtain an ancilla that is exponentially close to the desired magic state. The procedure is almost unchanged, except that the function involved is

$$f_{i,j,k} = a^i \left(bab^{-1}\right)^j x^{1-k}, \tag{3.125}$$

and the function $f'_{i,j,k} = a^i \left(bab^{-1}\right)^j$ must also be used to adjust the relative amplitude of $|1\rangle\otimes|1\rangle\otimes|1\rangle$ with respect to the other desired states.

In either case, we have now shown that the case of qubits can be dealt with in a similar fashion to the general qudit case, and therefore, we have completed the construction of universal quantum computation for anyons based on solvable non-nilpotent groups.

3.6 Leakage correction

Before concluding this chapter, it is important to address the issue of fault tolerance. A physical system with anyons will have sources of errors due to the finite separation of anyons and non-zero temperature as discussed in the previous chapter. While the probability of error is exponentially small in the distance and temperature, it is in general non-zero. These errors could be especially relevant if anyons are used as long-term quantum memory, in which case error correcting codes must be employed.

While most of the machinery of error correcting codes can be applied directly to anyons, it requires that states with errors remain within the computational subspace (that is, the subspace on which universal quantum computation can be done). For our model of computation, this is only a small subspace corresponding to anyons that are magnetic charges with fluxes such as $a^i b a^{-i}$ and are arranged in pairs of trivial total flux. Note that only the magnetic charges need error correction as they are the ones in which the quantum state is stored.

All that is required to perform quantum error correction is to be able to replace qudits that have "leaked out" of the computational subspace with arbitrary states that are in the computational subspace. This step can then be followed by the standard error correcting step, which will remove the errors. The leakage correction step is equivalent to the swap-if-leaked gate described by Kempe et al. [KBDW01].

In the previous chapter a leakage correction scheme was presented for non-solvable anyons. While a similar scheme could be constructed for the solvable anyons discussed in the present chapter, it will be easier to present a generic leakage correction scheme that can also be applied to anyons.

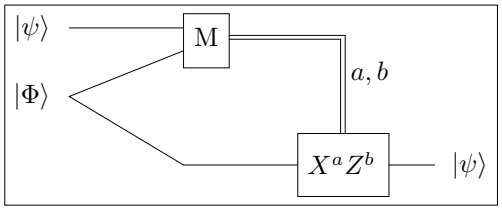


Figure 3.1: Leakage correction circuit.

The scheme is simply to teleport a computational qudit to a fresh qudit. The standard steps, shown in Fig. 3.1, are first to create the entangled ancilla $|\Phi\rangle = \sum_i |i\rangle \otimes |i\rangle/\sqrt{d}$, and then measure the computational qudit and the first ancilla qudit in the basis $|a, b\rangle = X^a Z^b \otimes I|\Phi\rangle$, obtaining outcome a, b. The correction gate $X^a Z^b$ is then applied to the second ancilla qudit, which now becomes part of the computational space. All these operations can be performed using the anyon gates discussed so far.

If the original qudit was in the computational space, then its state will be flawlessly transfered into the new qudit (in our case, a fresh anyon pair). However, if the original qudit had leaked, then the new qudit will be guaranteed to be in the computational subspace because it was obtained by applying Pauli operators to a qudit known to be in the computational subspace. This is the desired leakage correction protocol.

In fact, this scheme can be applied to almost any system, as long as we can guarantee that the measurement of the first two qudits will not affect the third qudit in any way, as should be the case if they are sufficiently separated.

The leakage correction scheme has a caveat from a theoretical standpoint, though. We are effectively assuming that we possess a classical leakage detection machine, through which the data "a, b" is run. That is, if the measurement produced an outcome in the form of a voltage, and then the gate $X^a Z^b$ was constructed as a Hamiltonian controlled by this voltage, we would need to guarantee that only the d^2 acceptable voltage signals could reach the machine operating on the third qudit. However, in practice, leakage correcting a classical signal is trivial, as classical information can be measured without any negative side effects.

A very similar scheme can be produced given a quantum system that is known to have exactly d states. The qudit is simply swapped into the new system, the first system is then erased and restored into the computational space, and then the qudit is swapped back. In this context, the teleportation scheme is in effect a way of swapping a qudit into a classical system.

Though the leakage correction scheme was discussed in general terms, it clearly applies to the anyons discussed in this thesis, and its use allows quantum error correction and fault tolerance to be employed. We have therefore shown that even in the presence of small sources of noise, the anyons can still be used for universal quantum computation.

3.7 Concluding remarks

The main result of this chapter is that anyons from finite groups that are solvable but not nilpotent are capable of universal quantum computation. This set includes many groups of small size, which are more likely to be found in a physical system. Combined with the results of the previous chapter, we have proven that every finite group that is not nilpotent produces anyons capable of universal quantum computation.

Furthermore, except for the groups where the methods of Sec. 3.5.6.4 must be used, the computations with anyons can be made error free in the following sense: in the theoretical limit of zero temperature and infinite separation between anyons, an arbitrarily long calculation can proceed without the need of error correction. The elementary unitaries are always perfect, whereas the measurements are either perfect, or are known to have failed (i.e., when none of the probabilistic projections succeed). This occurs with a probability that can be made exponentially small in the number of fusions. Of course, a real system will have additional exponentially small errors due to finite size and temperature effects.

The physical requirements for the constructions in this chapter include a supply of electric charge ancillas, in addition to the requirements of the previous chapter. The necessity of the electric charges may present an extra source of difficulties for a real implementation. The exception is S_3, in which case only magnetic charges are required, as mentioned at the end of Sec. 3.2. In either case, the issue of producing the elementary electric or magnetic ancillas is not addressed in this chapter, though a generalization of the construction in Sec. 2.6 may be sufficient.

We have also neglected to present an account of the resources used to perform computations. While it should be clear that computations can be done with at worst a polynomial overhead in the size of the input, some gates (in particular those that require calculations of arbitrary functions over the group) may require resources that are exponential in the size of the group. A lot of the wasted resources may come from the description in terms of general groups, though. For a fixed group, the resources can probably be significantly reduced.

Another open question is whether anyons from non-abelian nilpotent groups are capable of universal quantum computation. Additionally, not much is known about computing with anyons that do not belong to the electric and magnetic charge model discussed in this thesis. On the other hand, the universality of anyons from certain continuous groups has been discussed in [Fre00, FKLW01].

Of course, the most important open question is whether we can find a laboratory system with anyons out of which a quantum computer can be built. The requirement of a two-dimensional space severely limits the possibilities. Even if no physical implementations are ever found, though, this subject will hopefully still be interesting because of its beautiful mix of computation, particle physics and group theory.

Chapter 4

Serial composition and cheat detection for quantum coin-flipping

Analyzing quantum protocols with a large number of rounds is often difficult. One approach to obtaining a weak coin-flipping protocol with arbitrarily small bias could be to take a quantum coin-flipping protocol with a fixed number of rounds and compose it in series with itself to obtain a better coin-flipping protocol. It is well known that quantum coin protocols compose well in series, and an argument for this is given in Sec. 4.1.

Unfortunately, serially composing standard coin-flipping protocols does not decrease the overall bias [SV86]. However, quantum mechanics is good at detecting state disturbance and other deviations from a protocol. It is therefore possible to construct coin-flipping protocols with cheat sensitivity, where a dishonest player may be able to cheat by a significant amount, but only at the risk of getting caught by the honest player. Cheat sensitive protocols can produce improved coin-flipping protocols when composed in series under certain conditions.

In the present chapter, we will analyze the serial composition of cheat sensitive coin-flipping protocols. We shall treat the cheat sensitive protocols as oracles or black boxes, with a cheat sensitivity that is given by a function of the target bias. This will lead to our two main results.

In Sec. 4.2 we show that quadratic cheat detection protocols, where the probability of getting caught is proportional to the square of the bias, are a fixed point of serial composition, at least to leading order. This means that no matter how many times the protocol is composed with itself, the amount of cheat detection remains approximately the same. Because most known cheat sensitive protocols are quadratic or worse, this result is evidence that serial composition may not be useful to obtain weak coin-flipping. The main lemma used in this result is proven in Sec. 4.3.

In Sec. 4.4 we show that linear cheat detection cannot exist for strong coin-flipping. This is done by composing the linear cheat detection to obtain a strong coin-flipping protocol with arbitrarily small bias, in violation of Kitaev's lower bound. Note that the second result only uses serial composition as a tool for the proof, and the result holds for all cheat sensitive strong coin-flipping

protocols.

The result of 4.4 also applies to bit-commitment, which is a cryptographic protocol related to coin-flipping. Cheat detection as a function of ϵ can be defined in a way similar to [ATSVY00]. For Alice, ϵ is the amount by which she can change the probabilities associated with the committed bit, whereas for Bob, ϵ is the additional probability of guessing Alice's committed bit correctly. Because linear cheat detection in bit-commitment can be used to produce a strong coin-flipping protocol with linear cheat detection, it is also ruled out as a possible quantum protocol.

4.1 Serial composition of quantum coin protocols

Protocols for coin-flipping can be naturally composed in series to obtain new coin-flipping protocols. What is surprising at first is that quantum protocols for coin-flipping compose serially in such a way that a cheating party does not get any unexpected advantage by using entanglement. We shall prove this below, but let us first define carefully what we mean by unexpected advantage.

Let $\mathcal{P}$ be any quantum coin-flipping protocol. At the end of the protocol each player will output one of $\{0, 1, C\}$ where the last entry denotes the output when one player catches the other player cheating. Let us assume that Alice is honest and Bob is cheating. For each cheating strategy employed by Bob, there will be a triple of probabilities (P_0, P_1, P_C), one for each of Alice's possible outputs. Let $\Omega_A(\mathcal{P})$ be the set of all such attainable triples. Clearly $(1/2, 1/2, 0) \in \Omega_A(\mathcal{P})$ since Bob can always play honestly. If the protocol does not allow Bob to fully bias toward 1 then $(0, 1, 0) \notin \Omega_A(\mathcal{P})$. Some protocols may not have any cheat detection, in which case P_C will always be zero. For honest Bob and cheating Alice there is a similarly defined $\Omega_B(\mathcal{P})$.

Assume we take the protocol $\mathcal{P}$ and run it many times in series. We are interested in proving that a cheating player, say Bob, does not obtain any extra cheating power by using entanglement between different rounds. That is, that for every round j, independently of previous outcomes and strategies used by Bob, any strategy that Bob employs will make Alice output based on a triple of probabilities in $\Omega_A(\mathcal{P})$.

Clearly Bob can vary his strategy in each coin-flip round and even base his strategy on the outcomes of the previous coin-flips. However, when flipping N coins in series, Bob cannot obtain an outcome of all ones with a probability greater than $(P_{1,\max})^N$, where $P_{1,\max}$ is the maximum value of P_1 over any triple in $\Omega_A(\mathcal{P})$.

The reason that quantum coin-flipping can be serially composed stems from the following conditions, which are always imposed on coin-flipping protocols:

1. There is always at least one honest party.

2. The details of the protocol, which can be described in terms of fixed unitaries acting on a fixed

initial state, are known to all parties.

3. The protocol begins in an unentangled state, a condition that can be enforced by the honest party.

The first condition arises because no constraint is imposed on the case when both parties are cheating, and therefore the case never needs consideration. The third condition is always imposed to avoid trivial protocols, since establishing correlations is the goal of a coin-flipping procedure.

The proof of composition is fairly simple. At the beginning of the k^{th} round, the cheater will be unentangled from the honest player. The honest player has erased her Hilbert space and reset it to the initial state of the protocol. All that the cheating player has left over from the previous rounds is some quantum state in his Hilbert space. However, because the honest protocol is public, the cheating player knows exactly the state of his Hilbert space (which may be a mixed state, caused by the honest player erasing her Hilbert space).

Now assume that he could, with the help of this state, force the honest player (say Alice) to output with probabilities not in $\Omega_A(\mathcal{P})$. Then with the same state, he could obtain the same results in the first round or even when protocol $\mathcal{P}$ is used in a one-shot run, contradicting the definition of $\Omega_A(\mathcal{P})$. To put it another way, Bob can simulate in his private Hilbert space the first $k-1$ rounds and start playing the first round from that point, but this clearly can give him no extra advantage.

The conclusion of this section is that, given a quantum protocol for coin-flipping, we may treat the protocol as a black box when composing it in series with itself (or even with other coin-flipping protocols) without worrying about entanglement between rounds. We shall use this fact to derive two interesting results.

4.2 Quadratic cheat detection is a fixed point of coin-flipping

Because quantum coin-flipping composes in series, it is tempting to try to use a classical game layer on top of a known quantum coin-flipping protocol in order to reduce its maximum bias. That is, we wish to construct a two-player classical game that uses the quantum coin as a black box. The game could be, for instance, flipping a coin N times, and the party that wins a majority of coin tosses wins the game.

The ideal goal for this process would be to produce a weak coin-flipping protocol with arbitrarily small bias. While it is known that games built out of standard coin-flipping protocols can never reduce the maximum bias, the situation is different when cheat detection is available. Especially in the case of weak coin-flipping, where an honest player may declare himself the winner if he detects the other party cheating, there are certain black-box protocols that can be used to produce arbitrarily small bias. The question is how much cheat detection is needed in order for successive compositions

to improve a coin-flipping protocol?

We will be interested in protocols with symmetric, monomial cheat detection. Let $\mathcal{P}$ be a protocol where both parties have equal cheating opportunities (i.e., $\Omega_A(\mathcal{P}) = \Omega_B(\mathcal{P}) \equiv \Omega(\mathcal{P})$) and such that all probabilities $(P_0, P_1, P_C) \in \Omega(\mathcal{P})$ have the form:

$$P_0 = (1 - P_C)\left(\frac{1}{2} + \epsilon\right), \tag{4.1}$$

$$P_1 = (1 - P_C)\left(\frac{1}{2} - \epsilon\right), \tag{4.2}$$

$$P_C \geq a|\epsilon|^b, \tag{4.3}$$

which can be viewed as a function of a parameter ϵ that is controlled by the cheating party. The constants $a > 0$ and $b \geq 0$ denote the amount of cheat detection. Strictly speaking, P_C should also be considered a second parameter that can be controlled by the cheater, as he can always use a less optimal cheating strategy. In practice, a cheater will always minimize P_C for a given bias, and therefore we may assume equality in Eq. (4.3). We shall call the case $b = 1$ linear cheat detection, and the case $b = 2$ quadratic cheat detection.

When building games out of cheat sensitive coin-flips, the outcome of the entire game will be "cheating" if in any individual coin-flip a cheating outcome was obtained. To leading order in epsilon, the composite game will have a cheat sensitivity of the same form, with the same coefficient b, but in general a different a. When b is small, successive compositions increase a thereby producing a more cheat sensitive protocol, whereas in the large b regime composition has the opposite effect. We intend to prove in this section that $b = 2$ is a fixed point of coin-flipping. That is, for all possible games that use a coin with quadratic cheat detection as a black box, the resulting protocol has exactly the same amount of cheat detection to leading order in epsilon.

We begin by describing the set of all possible games that employs a cheat sensitive coin-flip as a black box. These can be put in correspondence with the set of binary trees, where the leaf nodes are labeled by either zero or one. For instance, the tree corresponding to the "best two out of three" game is depicted in Fig. 4.1. Each binary node corresponds to a coin flip, with up corresponding to the outcome zero and down to the outcome one. The leaves are endpoints of the game, and their labels correspond to the final game outcome at that node.

Other examples of games that produce coin-flips include "first outcome that is repeated N times" and "first outcome to occur a total of N times more than the other," both of which correspond to trees of infinite length. Allowing for trees of infinite length also accounts for all games that have a tie outcome, after which the game is restarted. For trees of infinite length we impose the additional constraint that when playing honestly, the probability of never reaching a leaf node is zero.

Let x be a variable that ranges over the binary nodes of the tree. A general cheating strategy is

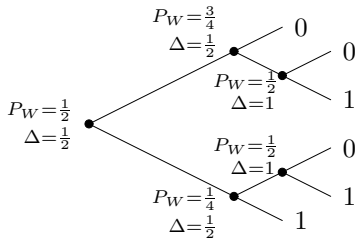


Figure 4.1: "Best two out of three" game tree. Binary nodes are labeled by the probability of winning for an honest player trying to obtain outcome 0.

a function $\epsilon(x)$, which assigns a bias to each node where a coin-flip takes place. We also define $p_\epsilon(x)$ as the probability of arriving at node x given a cheating strategy $\epsilon(x)$. In defining this quantity, we assume that if the cheater is caught at a given node, the game stops, thereby reducing the probability of arriving at the child nodes. In terms of these quantities we can write the total probability of getting caught:

$$P_{C,tot} = \sum_x a p_\epsilon(x) \left|\epsilon(x)\right|^b . \tag{4.4}$$

To leading order in $\epsilon(x)$, we can replace $p_\epsilon(x)$ with the probability of arriving at the node in the honest protocol. This can be written as $2^{-D(x)}$ where $D(x)$ is the depth of the node x, with the root node having depth zero. The formula becomes:

$$P_{C,tot} = \sum_x a 2^{-D(x)} \left|\epsilon(x)\right|^b . \tag{4.5}$$

To expand the total bias to leading order in epsilon, we only need to keep track of terms that are linear in $\epsilon(x)$ and can therefore disregard the multiplicative factor $1 - P_C$ in the probabilities for obtaining zero or one. Of course, we are assuming $b > 1$ at this point and will soon concentrate on $b = 2$. The probability of the cheater obtaining his desired outcome, and winning the game, satisfies a simple recursion relation:

$$P_W(x) = \frac{1}{2}\left(P_W(x^\uparrow) + P_W(x^\downarrow)\right) + \epsilon(x)\left(P_W(x^\uparrow) - P_W(x^\downarrow)\right), \tag{4.6}$$

where $P_W(x)$ is the probability of winning having arrived at node x, and $x^\uparrow$, $x^\downarrow$ are the two children of node x. At the root node the probability of winning is

$$P_{W,tot} = \frac{1}{2} + \sum_x 2^{-D(x)} \Delta(x) \epsilon(x), \tag{4.7}$$

where $\Delta(x) = P_W(x^\uparrow) - P_W(x^\downarrow)$, which can be computed at this point from the honest probability

of winning.

The total bias for the game is given to leading order by $|\epsilon_{tot}| = P_W - 1/2$, where we have excluded cases where the cheating benefits the honest player. For a given total bias, the cheater will choose $\epsilon(x)$ to minimize the probability of getting caught. The calculation to minimize $P_{C,tot}$ under the constraint of fixed ϵ_{tot} can easily be done using a Lagrange multiplier, to obtain the result:

$$\epsilon(x) = \mathrm{sgn}(\lambda\Delta(x)) \left| \frac{\lambda\Delta(x)}{ab} \right|^{\frac{1}{b-1}}, \tag{4.8}$$

where λ is the Lagrange multiplier. We have allowed $\epsilon(x)$ to range over the reals, but will show below that for quadratic cheat detection the optimal solution satisfies the requirement $|\epsilon(x)| \leq 1/2$.

To eliminate the Lagrange multiplier, we can substitute the expression for ϵ_{tot} into the expression for $P_{C,tot}$ to obtain:

$$P_{C,tot} = a_{new} \left|\epsilon_{tot}\right|^b, \tag{4.9}$$

where

$$a_{new} = a \left(\sum_x 2^{-D(x)} \left|\Delta(x)\right|^{\frac{b}{b-1}} \right)^{1-b}. \tag{4.10}$$

For the case $b = 2$, the following lemma shows that $a_{new} = a$ for all game trees:

Lemma 5. *For all game trees as described above, we have the equality*

$$\sum_x 2^{-D(x)} \Delta(x)^2 = 1. \tag{4.11}$$

The lemma can be proven using a simple combinatorial argument that is presented for the interested reader in the next section.

From Eq. (4.8) we see that for $b = 2$, the amount of cheating at node x is proportional to $\Delta(x)$. The constant of proportionality, as well as the Lagrange multiplier, are fixed by Eq. (4.7). After eliminating a factor of 1 using the above lemma we obtain that the optimal strategy is $\epsilon(x) = |\epsilon_{tot}|\Delta(x)$. This satisfies the intuition that the cheater will choose a larger bias on coin-flips that are more consequential. Furthermore, because $|\Delta(x)| \leq 1$, we have shown that the optimal strategy is achievable (i.e., $|\epsilon(x)| \leq 1/2$ for every x) whenever $|\epsilon_{tot}| \leq 1/2$.

For other values of b near two, we can consider the derivative of a_{new} with respect to b. For any fixed graph, this derivative is negative. The conclusion is that for every graph, serial composition of $b < 2$ cheat detection leads to improvement to lowest order, whereas in the $b > 2$ case the cheat

sensitivity worsens.

The above results complete our argument that quadratic cheat detection is a fixed point for coin-flipping. The argument is only valid in the regime where all biases, including the total bias, are small. Unfortunately, because a protocol producing weak coin-flipping with arbitrarily small bias would have to employ arbitrarily large trees, it is not sufficient to simply take the small bias limit on a fixed tree.

In essence, associated with each weak coin-flipping protocol there is a function $P_C(\epsilon)$ indicating the minimum probability with which a party will get caught cheating if they try to bias the coin by ϵ. Composing this protocol in series one obtains a new protocol with function $P_{C,tot}(\epsilon_{tot})$. That is, serial composition with a given game tree induces a map from the set of functions $P_C(\epsilon)$ to itself. We have shown that, independent of the tree, if the original function behaves as $a\epsilon^2$ for small ϵ, then so will its image.

The fact that the coefficient of the quadratic term remains fixed under the map induced by every game tree is a peculiar and interesting feature. It is indicative that serial composition of quadratic cheat detection may not be useful for producing weak coin-flipping with arbitrarily small bias. However, further research in this direction is needed in order to conclusively settle the issue.

4.3 Proof of lemma

In this section we shall prove Lemma 5, but first we introduce some notation. Let $\mathcal{T}$ be a binary tree. We will use the variable x to denote a binary node of $\mathcal{T}$ and y to denote a leaf.

Associated with $\mathcal{T}$ there is a function $P_W(y)$ on the leaf nodes that takes the values zero or one. The function can be extended to the rest of the nodes by defining $P_W(x)$ as the average of the function on its two descendants. In terms of coins, P_W is simply the probability of winning when playing honestly starting from the given node. The function corresponds to the coin outcomes on the leaf nodes if the goal is to obtain 1, and otherwise the coin outcome is $1 - P_W(y)$. To obtain a fair coin toss, we require that P_W equal $1/2$ on the root node. Otherwise, the function and the tree are arbitrary.

We also arbitrarily label the two outgoing edges from each binary node as up and down, and define $\Delta(x) = P_W(x^{\uparrow}) - P_W(x^{\downarrow})$. Finally, let $D(x)$ be the depth of node x, with the root node having depth zero.

Lemma 6 (restatement of lemma 5). *Given a tree $\mathcal{T}$ and function P_W as described above, the following equality holds:*

$$\sum_x 2^{-D(x)} \Delta(x)^2 = 1. \tag{4.12}$$

Proof. $\Delta(x)$ is a linear combination of $P_W(x)$ on its descendants, which in turn is a linear combination of $P_W(y)$ on the leaf nodes. Therefore the left-hand side of the above equation is a quadratic polynomial of $P_W(y)$.

Fix a leaf node y. The function $P_W(y)$ only appears in $\Delta(x)$ if y is a descendant of x, in which case it has a coefficient of $\pm 2^{D(x)+1-D(y)}$. The coefficient of $P_W(y)^2$ in this polynomial is therefore given by

$$\sum_{i=0}^{D(y)-1} 2^{-i} 2^{2(i+1-D(y))} = 2^{2(1-D(y))} \left(2^{D(y)} - 1\right). \tag{4.13}$$

Fix a second leaf node $y' \neq y$. Let x' be the unique node that has y as a descendant along one branch and y' along the other. The coefficient of $P_W(y)P_W(y')$ is

$$2\left[\sum_{i=0}^{D(x')-1} 2^{-i+2(i+1)-D(y)-D(y')}\right] - 2^{D(x')-D(y)-D(y')+3} = -2^{3-D(y)-D(y')}, \tag{4.14}$$

where the only negative term is contributed by $\Delta(x')$. Note the extra factor of 2 accounting for the double occurrence of $P_W(y)P_W(y')$.

Combining these results, the left-hand side of Eq. (4.12) is

$$4\sum_y 2^{-D(y)} P_W(y)^2 - 4\left[\sum_y 2^{-D(y)} P_W(y)\right]^2. \tag{4.15}$$

Note that the factor in brackets is just P_W at the root node, which must equal 1/2. The left summand can be simplified by using $P_W(y)^2 = P_W(y)$, in which case it can also be written in terms of P_W at the root node. We have shown that the left-hand side equals $4(1/2) - 4(1/2)^2 = 1$ as desired. □

4.4 No-go theorem for linear cheat detection

In this section we shall switch gears and focus on serial composition of linear cheat detection protocols. We shall prove that linear cheat detection can be serially composed to produce not only weak coin-flipping but also strong coin-flipping with arbitrarily small bias. Because of Kitaev's bound for quantum strong coin-flipping [Kit03], the result in this section proves that a strong coin-flipping protocol with linear cheat detection cannot exist.

The result of this section only applies to strong coin-flipping schemes with cheat sensitivity as described in Eq. (4.3). An alternative not covered by the proof in this section is the case where Alice can force outcome 1 (and Bob can force outcome 0) without getting caught, which corresponds to cheat sensitive weak coin-flipping. Because weak cheat sensitivity can be simulated by strong cheat sensitivity, the result of the previous section applies to weak cheat sensitivity as well. However, the

opposite is not true, and the results of the present section do not apply to linear cheat sensitivity in weak coin-flipping.

Since strong coin-flipping can be constructed out of bit-commitment, the result of this section can also be applied to cheat sensitive bit-commitment protocols. In fact, the only known coin-flipping protocol, where neither side can cheat by a finite amount without getting caught, is described in [HK04] and is a strong coin-flipping protocol that is constructed out of bit-commitment.

For the proof of the above statements, we assume the existence of a quantum protocol $\mathcal{P}$ described by Eq. (4.3) with $b = 1$ and some non-zero a. We shall describe a game, that uses $\mathcal{P}$ as a black box, which achieves strong coin-flipping with bias that becomes arbitrarily small in the limit of a game parameter $N \to \infty$. Though, for the purposes of comparing against Kitaev's bound, it is sufficient to allow honest players to output "cheat" at the end; we will construct the game so that the honest player always outputs one of the outcomes zero or one.

The game is the random walk on a 1-D line, starting from the origin, using the coin provided by $\mathcal{P}$. The game ends with the first arrival at one of the two sites $\pm N$, with the right end corresponding to the outcome zero, and the left end to one.

If one party detects cheating, they will continue the game using a private fair coin, and output according to the outcome of the game. Let $z \in \{-N, \ldots, N\}$ be a variable that runs along the line. If cheating occurs when the game is at z, then the honest players can simply output using $P_0 = (N + z)/2N$ and $P_1 = (N - z)/2N$. In essence, the only deterrent to the cheater is that he may be able to cheat more effectively in a future round. Note that the honest party cannot just flip a balanced coin after detecting cheating because in this case the cheating party would only cheat when he is about to lose.

Because the honest player keeps no state beyond the current location along the line, z, there is an optimal cheating strategy where the bias only depends on z. That is, we only need to consider functions $\epsilon(z)$ when maximizing over cheating strategies.

We assume that the cheating player is trying to bias toward zero (i.e., the right side). We can then define the function $W_\epsilon(z)$ to be the probability of winning starting from node z, using biases $\epsilon(z)$. The function is similar to P_W defined in the previous section, except that we are now using large biases, which cannot be ignored in calculating the probability of winning.

The function has the constraints $W_\epsilon(N) = 1$, $W_\epsilon(-N) = 0$ and

$$W_\epsilon(z) = P_0(\epsilon(z))W(z+1) + P_1(\epsilon(z))W(z-1) + P_C(\epsilon(z))\frac{N+z}{2N}. \tag{4.16}$$

Clearly, for optimal strategies, $W_\epsilon(z) \geq (N + z)/2N$ for all z because the cheating party can always play honestly.

The value of $W_\epsilon(0)$ is simply the probability of winning the entire game. The cheater will chose

$\epsilon(z)$ in order to maximize $W_\epsilon(0)$. We shall prove that $\max[W_\epsilon(0) - 1/2] \to 0$ as $N \to \infty$.

The analysis is made easier by using a modified black box protocol $\mathcal{P}'$ with achievable probabilities of the form:

$$P_0 = \frac{1}{2} + \epsilon, \tag{4.17}$$

$$P_1 = \frac{1}{2} - (1+a)\epsilon, \tag{4.18}$$

$$P_C = a\epsilon, \tag{4.19}$$

for $0 \le \epsilon \le \epsilon_{max}$ where $\epsilon_{max} = 1/(2+2a)$. For every ϵ, protocol $\mathcal{P}'$ gives the cheater a slightly higher probability of obtaining the desired outcome than with protocol $\mathcal{P}$. Therefore, any bounds on cheating obtained using $\mathcal{P}'$ as a black box will apply when using $\mathcal{P}$ instead.

Consider using $\mathcal{P}'$ and varying independently each of the nodes of the complete game tree. The bias $\epsilon(x)$ of each node enters linearly into $W_\epsilon(0)$, which can be maximized by letting the biases take only the boundary values of 0 or ϵ_{max}. As discussed above, the optimal biases will depend only on the corresponding value of z, and therefore we need only consider functions $\epsilon(z)$, which take values in $\{0, \epsilon_{max}\}$.

The maximization is now easy to analyze. Define $\delta(z) = W_\epsilon(z) - (N+z)/2N$, which is the extra probability of winning that the cheater is getting at position z. We shall prove below that

$$\delta(z+1) \le \frac{2+a}{2aN} \quad \Longrightarrow \quad \delta(z) \le \frac{2+a}{2aN}. \tag{4.20}$$

If $\epsilon(z) = \epsilon_{max}$ the above statement is true because Eq. (4.16) states that:

$$\delta(z) = \left(\frac{1}{2} + \epsilon_{max}\right)\left(\delta(z+1) + \frac{1}{2N}\right) = \frac{2+a}{2+2a}\left(\delta(z+1) - \frac{2+a}{2aN}\right) + \frac{2+a}{2aN}. \tag{4.21}$$

On the other hand, if $\epsilon(z) = 0$ then Eq. (4.16) states that:

$$\delta(z+1) - \delta(z) = \delta(z) - \delta(z-1), \tag{4.22}$$

that is, δ has a constant slope around z. If for all $z' < z$ we have $\epsilon(z') = 0$, then the slope must be constant through this entire region. Since $\delta(-N) = 0$, the slope can only be negative (i.e., increasing toward the left) if $\delta(z) < 0$, which proves Eq. (4.20) for this case. Otherwise, let $z' < z$ be the largest integer such that $\epsilon(z') = \epsilon_{max}$. Again the slope must be constant from z', in which case, by Eq. (4.21) the slope can only be negative if $\delta(z') < (2+a)/2aN$, which implies $\delta(z) < (2+a)/2aN$.

Having proven Eq. (4.20), and using the initial case $\delta(N) = 0$, we have shown

$$\epsilon_{tot} \equiv W_\epsilon(0) - \frac{1}{2} = \delta(0) \le \frac{2+a}{2aN}, \tag{4.23}$$

which can be made arbitrarily small by taking the limit $N \to \infty$.

The game described in this section shows that a strong coin-flipping protocol with linear cheat detection can be serially composed to obtain a strong coin-flipping protocol with arbitrarily small bias. The conclusion is that quantum strong coin-flipping protocols with linear cheat detection cannot exist.

4.5 Conclusions

Using serial composition of coin-flipping we have established an upper bound on the amount of cheat detection possible in quantum protocols for coin-flipping and bit-commitment. We have also presented evidence that serially composing quadratic cheat sensitive protocols does not lead to an improvement in the amount of cheat detection. We speculate that quadratic or worse cheat sensitivity ($b \geq 2$) cannot be composed in series to obtain weak coin-flipping with arbitrarily small bias. We also speculate that cheat detection better than quadratic ($b < 2$) does not exist for bit-commitment or strong coin-flipping, and probably not for weak coin-flipping either.

Nevertheless, linear cheat detection in weak coin-flipping remains an open, though unlikely, possibility. In fact, by serially composing weak coin-flipping with linear cheat detection, it is not hard to show that one can construct a weak coin-flipping protocol with a bias of exactly zero. The apparent contradiction with the result of Lo and Chau [LC98] is resolved by noting that they only considered protocols with a fixed number of rounds. However, there are protocols where the number of rounds is variable and possibly arbitrarily large (a good classical example is rock-paper-scissors), while still having a zero probability of going on forever. For these protocols the measurements cannot be delayed to the final round, and therefore the analysis of Lo and Chau does not apply. These protocols can always be truncated to a finite number of rounds, though, at the cost of allowing an arbitrarily small bias.

Other questions that remain open include: What happens to serial composition of quadratic cheat detection in the large bias regime? And what can be said for other functional forms of cheat detection versus bias, including cases where one party may be able to cheat by a small amount without getting caught? Unfortunately, these questions will remain open as we turn our attention in the following chapters to protocols for weak coin-flipping with no explicit cheat detection.

Chapter 5

Quantum weak coin-flipping with bias of 0.192

In this chapter we shall describe a family of protocols for quantum weak coin-flipping that achieve biases as low as 0.192. The set of protocols described in this chapter will only be a small subset of the family of protocols presented in the next chapter. Nevertheless, some of the key steps in upper bounding the bias will be taken in this chapter.

The actual protocols are described in Sec. 5.1 and are indexed by an integer $n > 1$, with $n + 2$ messages. The case of $n = 2$ with four messages will be equivalent to Spekkens and Rudolph's original protocol. The protocols with more rounds will achieve an even better bias.

The main achievement of this chapter is the construction of quantum weak coin-flipping protocols, which achieve a bias less than $\epsilon = 1/\sqrt{2} - 1/2$. This excludes the possibility that Kitaev's bound for strong coin-flipping can be directly extended to weak coin-flipping and establishes that it is not possible for the minimum bias of weak coin-flipping to equal that for strong coin-flipping in the context of quantum mechanics.

The main technique used in this chapter is Kitaev's description of coin-flipping as a semidefinite program. This description provides a dual problem whose solutions bound the amount that a party may cheat. Though this material has been previously published, it will be reviewed in Sec. 5.2.

The main contribution of the chapter is in Sec. 5.3, where we shall construct solutions to the problem dual to the protocol of Sec. 5.1. These shall provide analytic upper bounds on the bias of the protocol.

Our protocol for a given n depends on n parameters subject to one constraint, and we shall express the upper bound as a function of these parameters. Any choice of the parameters, consistent with the constraint, will give a valid protocol together with an upper bound on its bias. To find good protocols with small bias, we shall use a numerical minimization over the space of parameters. This will be done in Sec. 5.4.

We stress, however, that given a set of values for the parameters, these can be put into the

analytic expression to obtain a valid upper bound on the bias. The existence of weak coin-flipping protocols with the quoted biases does not depend in any way on the accuracy or quality of the numerical minimization.

We also include in Sec. 5.5 an analytical derivation of the bias in the limit $n \to \infty$, which will show that most reasonable choices of the protocol parameters converge to the same bias.

5.1 The protocol

In this section we will describe a family of weak coin-flipping protocols, indexed by an integer $n \geq 2$, which will involve $n+2$ messages. The protocols will also depend on a set of parameters $a_1, \ldots, a_n$ to be fixed later. These parameters define the two-qubit states

$$|\phi_i\rangle = \sqrt{a_i}|00\rangle + \sqrt{1-a_i}|11\rangle. \tag{5.1}$$

The protocol begins with Alice preparing in her private Hilbert space the states $|\phi_i\rangle$ for odd i, while Bob prepares the states with even i in his Hilbert space.

The first n messages of the protocol consist of sending halves of the states $|\phi_i\rangle$. More explicitly, the i^{th} message involves the owner of state $|\phi_i\rangle$, who sends one of the two qubits comprising the state to the other party. After the first n messages, if both players were honest, the state of the system should be:

$$|\psi\rangle_{AB} = \bigotimes_{i=1}^{n} \left(\sqrt{a_i}|0\rangle_A \otimes |0\rangle_B + \sqrt{1-a_i}|1\rangle_A \otimes |1\rangle_B\right), \tag{5.2}$$

where the labels A, B denote the owner of the qubit in question.

At this point each side will apply a two-outcome projective measurement $\{E_0, E_1\}$ to their n qubits. These operators will be described below but will have the properties $E_i^2 = E_i$ and $(E_i)_A \otimes I_B|\psi\rangle_{AB} = I_A \otimes (E_i)_B|\psi\rangle_{AB}$. These properties guarantee that when both parties are honest, their answers are perfectly correlated. We can therefore associate the outcome E_0 with an outcome of zero for the coin flip, and the outcome E_1 with coin outcome one. The requirement that the coin-flip be fair when both parties are honest:

$$\langle\psi|E_i \otimes E_i|\psi\rangle = \frac{1}{2} \tag{5.3}$$

will impose a constraint on the parameters $\{a_i\}$.

At this point both parties should know the "honest" outcome of the coin-flip. Now they enter a stage of cheat detection in which the loser will examine the qubits of the winner. If no cheating is detected (which is guaranteed when both players are honest) then the "honest" outcome becomes the

final outcome. Otherwise, if the losing party detects cheating, that party may ignore the "honest" outcome and instead output his or her desired outcome (zero for Alice and one for Bob). This is acceptable because the rules of weak coin-flipping don't require the parties to output the same bit when one party is dishonest.

We now describe the cheat detection stage, which will involve the last two messages: the winner of the coin toss according to the measurement $\{E_0, E_1\}$ sends over their entire Hilbert space for inspection. If Bob wins, he should send over his n qubits so that Alice obtains both halves of the state:

$$\sqrt{2}E_1 \otimes E_1|\psi\rangle = \sqrt{2}E_1 \otimes I|\psi\rangle. \tag{5.4}$$

This is a pure state, and Alice can perform a two-outcome projection onto this state and its complement. If she obtains the complement as outcome, she knows Bob must have cheated. More specifically, define

$$F_i = \frac{E_i \otimes E_i|\psi\rangle\langle\psi|E_i \otimes E_i}{\langle\psi|E_i \otimes E_i|\psi\rangle}. \tag{5.5}$$

Alice measures using the projections $\{F_1, I - F_1\}$, where outcome $I - F_1$ implies Bob has cheated. In the case when the honest outcome is zero, Bob does the equivalent steps with $\{F_0, I - F_0\}$.

Officially, we shall define the protocol so that Alice always uses message $n + 1$ to either send her qubits (or nothing if she lost the honest coin toss), whereas Bob will use message $n + 2$ if he needs to send qubits. The ordering is irrelevant though, and it could also be defined so that Bob sends his verification qubits first when n is odd, thereby avoiding two messages in a row from Alice. Alternatively, they could be sent in the opposite order, combining the verification state with the last message in order to run the protocol with only $n + 1$ messages.

All that remains is to describe the projections E_0 and E_1. Heuristically, the measurement consists of the following process: examine the qubits in order starting from the one belonging to $|\phi_n\rangle$ and ending with the one belonging to $|\phi_1\rangle$. The qubits are to be measured in the computational basis, until the first zero outcome is obtained, which implies that the sender of that qubit loses. If all qubits produce outcome one then Alice (being the first message sender) is the winner. Of course, the measurement is not performed in stages as described above but rather using the unique pair of projectors that produces the same distribution of probabilities. For example, for $n = 2$ we have

$$E_0 = |00\rangle\langle 00| + |10\rangle\langle 10| + |11\rangle\langle 11|, \tag{5.6}$$

$$E_1 = |01\rangle\langle 01|, \tag{5.7}$$

where the leftmost qubit corresponds to the first qubit sent or received. The rest can be defined

inductively by the formulas

$$E_0^{(k+1)} = I \otimes E_1^{(k)} + |1\cdots1\rangle\langle1\cdots1|, \tag{5.8}$$

$$E_1^{(k+1)} = I \otimes E_0^{(k)} - |1\cdots1\rangle\langle1\cdots1|, \tag{5.9}$$

where the superscript indicates the number of qubits on which they are to act. For brevity, these superscripts shall be omitted, though.

The case of $n = 2$ is equivalent to the protocol described by Spekkens and Rudolph in [SR02b], which achieves the tradeoff $P_A^* P_B^* = 1/2$. The connection is made by setting $a_1 = x$ and $(1 - a_2) = 1/(2x)$.

Summarizing, the protocol involves the following steps:

1. Alice prepares $|\phi_1\rangle \otimes |\phi_3\rangle \otimes |\phi_5\rangle \cdots$,
   Bob prepares $|\phi_2\rangle \otimes |\phi_4\rangle \otimes |\phi_6\rangle \cdots$.

2. For $i = 1$ to n:
   If i is odd: Alice sends half of the state $|\phi_i\rangle$ to Bob,
   If i is even: Bob sends half of the state $|\phi_i\rangle$ to Alice.

3. Alice performs the two-outcome measurement $\{E_0, E_1\}$ on her n qubits. Bob performs the same two-outcome measurement $\{E_0, E_1\}$ on his n qubits.

4. If Alice obtains E_0 she outputs zero and sends all her qubits to Bob.

5. If Bob obtains E_1 he outputs one and sends all his qubits to Alice.

6. If Alice obtained E_1 she measures her qubits plus any qubits received from Bob with the projections $\{F_1, I - F_1\}$. If she obtains F_1 she outputs one, otherwise (or if she receives the wrong number of qubits from Bob) she outputs zero.

7. If Bob obtained E_0 he measures his qubits plus any qubits received from Alice with the projections $\{F_0, I - F_0\}$. If he obtains F_0 he outputs zero, otherwise (or if he receives the wrong number of qubits from Alice) he outputs one.

5.1.1 Reformulation of the protocol

For the analysis in the following section, it will be helpful to delay all measurements to the last step. It will be also useful to never have to apply a unitary or, equivalently, send qubits conditioned on the outcome of a measurement. We will be able to formulate protocols with these properties if we are willing to allow one side to have increased cheating power.

The idea is that the analysis of a coin-flipping protocol is divided into two separate steps: we need to analyze the case when Alice is honest and Bob is cheating, and then we need to analyze the case when Bob is honest and Alice is cheating. Let us focus on the first case when Alice is honest.

We wish to describe a new coin-flipping protocol, where Bob's ability to cheat is exactly the same as in the original protocol, but where Alice may be able to cheat more than usual. We shall call the original protocol $\mathcal{P}$ and the new protocol $\mathcal{P}'$. The idea is that $\mathcal{P}'$ will be simpler to describe than $\mathcal{P}$ and, since at the moment we are only concerned with bounding Bob's ability to cheat, any bound derived for one protocol will apply to the other.

The protocol $\mathcal{P}'$ begins with the same initial state as $\mathcal{P}$, and the first n messages are identical. However, in $\mathcal{P}'$ after the first n messages no measurements occur. Instead, Bob sends all of his n qubits to Alice. After this last message Alice performs the two-outcome projective measurement $\{F_1, I - F_1\}$ as before and reports outcome F_1 as Bob winning and $I - F_1$ as Alice winning. Note that in $\mathcal{P}'$, even when Bob is honest the outcome $I - F_1$ can arise, that is, Alice does not differentiate between Bob losing honestly and Bob getting caught cheating.

Technically, we should allow one last classical message from Alice to Bob, where Alice announces her outcome and then Bob repeats it as his own, but this won't be necessary as we are only concerned with the probabilities associated with Alice's output.

It is not hard to see that any cheating strategy for Bob that can be used in $\mathcal{P}$ will produce the same probability of winning in $\mathcal{P}'$, because the only thing that changed from Alice's perspective is that now she always expects to receive Bob's qubits. However, as protocol $\mathcal{P}$ was written, the only time that Bob could win was when he sent his qubits, so he loses nothing by always sending them. From a mathematical perspective, we are using the fact that $F_1 E_1 = F_1$ and $(I - F_1)E_1 + E_0 = I - F_1$.

In conclusion, when analyzing the case of honest Alice, we can use protocol $\mathcal{P}'$. Of course, when analyzing the case of honest Bob and cheating Alice, $\mathcal{P}'$ is no longer useful, but we can define a new protocol $\mathcal{P}''$ in a similar way, where Alice always sends all her qubits to Bob. This protocol can be used to bound Alice's cheating power in $\mathcal{P}$.

For the rest of this chapter, we shall employ protocols $\mathcal{P}'$ and $\mathcal{P}''$ where appropriate without further comment. However, all bounds derived will apply to the original protocol $\mathcal{P}$ as well.

5.2 Coin-flipping as an SDP

The problem of finding the optimal cheating strategy for a player can be cast as a semidefinite program (SDP). The dual problem then provides bounds on the maximum bias that the cheating player may achieve. This approach was first described by Kitaev [Kit03] (and summarized in [ABDR04]).

In the following section we will review Kitaev's construction, though using a somewhat different language than the original. What few results are needed from the theory of semidefinite programming

will be derived along the way in order to keep this chapter as self contained as possible. The discussion in this section will be completely general in the sense that it applies to any coin-flipping protocol. The results of this section will then be applied to the protocol at hand in Sec. 5.3.

5.2.1 The primary problem

For simplicity, we shall focus on the case when Alice is honest and Bob is cheating. The opposite case when Bob is honest is nearly identical.

We will work with protocols that can be cast in the following form: The initial state is a fixed pure unentangled state shared by Alice and Bob. The protocol proceeds by applying unitaries on each individual side, and by sending qubits from Alice to Bob and vice-versa. In the last step, each party performs a two-outcome projective measurement and outputs the result.

In fact, the communication part of the protocol (i.e., everything but the initial state preparation and the final measurement) can be described as a sequence of the following three elementary operations: one of Alice's qubits is sent to Bob, one of Bob's qubits is sent to Alice, or each side applies a unitary to their qubits. The unitary step is often not needed and can be completely removed if we allow each party to decompose their space into qubits in different ways on each round.

Given a protocol, let m be the number of elementary steps, and let ρ_0 be the density matrix describing Alice's qubits in the first step. Let ρ_i be the density matrix describing Alice's qubits after the first i elementary operations, given some cheating strategy for Bob. These matrices must satisfy the following equations:

- If step i involves sending qubit j from Alice to Bob:

$$\rho_i = \mathrm{Tr}_j\, \rho_{i-1}. \tag{5.10}$$

- If step i involves Alice receiving a qubit from Bob and assigning it name j:

$$\mathrm{Tr}_j\, \rho_i = \rho_{i-1}. \tag{5.11}$$

- If step i involves Alice applying unitary U_i:

$$\rho_i = U_i \rho_{i-1} U_i^{-1}. \tag{5.12}$$

It will be convenient to have a shorthand notation for these equations. They shall be written as $L_i(\rho_i) = R_i(\rho_{i-1})$, where L_i and R_i are linear operators corresponding to the identity, partial trace or conjugation by a unitary as needed to match the above equations.

Clearly, no matter what Bob's strategy is, the above equations must be satisfied. Furthermore, because Alice's output probabilities are entirely determined by ρ_m, a cheating strategy for Bob can be described in terms of the above sequence of density operators $\{\rho_i\}$. In fact, it is not hard to see that by keeping the total state pure, Bob can make Alice have any sequence of density operators that are consistent with the above equations. Therefore, there is a one-to-one correspondence between cheating strategies of Bob (up to isomorphisms that produce the same result on Alice's side) and density operators $\rho_0, \ldots, \rho_m$ satisfying the above equations.

After the communication rounds have been completed, Alice makes a two-outcome projective measurement $\{E_A, E_B\}$ to determine her output. Outcome E_A will correspond to Alice winning (i.e., final outcome zero) and E_B will correspond to Bob winning (i.e., final outcome one). Note that these operators are not the same as the $\{E_0, E_1\}$ used in the last section, and in fact, when applied to our protocol E_B will correspond to F_1.

Bob's goal is to choose a sequence of positive semidefinite operators $\rho_1, \ldots, \rho_m$ satisfying the above protocol dependent equations, in order to maximize $\mathrm{Tr}(E_B\rho_m)$. Note that ρ_0 is always fixed by Alice's initial state, and the above equations fix the trace of the remaining matrices, therefore the maximization can indeed be done over all positive semidefinite matrices. We have therefore proven the following lemma:

Lemma 7. *The maximum probability of winning that can be attained by Bob through cheating in a coin-flipping protocol described by the data m, ρ_0, $\{L_i\}$, $\{R_i\}$, E_B is given by the solution of the maximization problem*

$$P_B^* = \max \mathrm{Tr}\left(E_B\rho_m\right), \tag{5.13}$$

involving the m positive semidefinite matrices $\rho_1, \ldots, \rho_m$ subject to the constraints

$$L_i(\rho_i) = R_i(\rho_{i-1}) \quad \textit{for all} \quad i = 1, \ldots, m. \tag{5.14}$$

5.2.2 The dual problem

The beauty of semidefinite programing is that each SDP has a dual SDP. When the original problem involves a maximization, the dual problem involves a minimization. Furthermore, the optimal solution of the dual problem will be greater than or equal to the optimal maximum of the original problem. In terms of coin-flipping each solution of the dual problem provides an upper bound on the amount that Bob can cheat.

The variables of a dual SDP are Lagrange multipliers, one for each constraint in the original problem. There are m equality constraints given by the m elementary operations of the protocol, therefore there will be m Lagrange multipliers $Z_1, \ldots, Z_m$. Each Z_i will be a Hermitian matrix of

the same dimension as $L_i(\rho_i)$ and will be added in as a term of the form $\mathrm{Tr}[Z_i(L_i(\rho_i) - R_i(\rho_{i-1}))]$.

We will now lift the conditions $L_i(\rho_i) = R_i(\rho_{i-1})$ on the operators $\{\rho_i\}$ allowing them to vary freely. The constraints will be dynamically imposed by the Lagrange multiplier terms. However, because the traces of $\{\rho_i\}$ are no longer fixed we shall impose the constraints $\rho_i \leq I$ so as to keep the expression $\mathrm{Tr}\, E_B \rho_m$ finite. We now have:

$$\begin{aligned}
P_B^* &= \max_{0\leq\rho_1,\ldots,\rho_m\leq I} \left\{ \mathrm{Tr}\, E_B\rho_m - \sup_{Z_1,\ldots,Z_m} \sum_{i=1}^{m} \mathrm{Tr}\left[Z_i(L_i(\rho_i) - R_i(\rho_{i-1}))\right] \right\} \\
&= \max_{0\leq\rho_1,\ldots,\rho_m\leq I} \inf_{Z_1,\ldots,Z_m} \left\{ \mathrm{Tr}\, E_B\rho_m - \sum_{i=1}^{m} \mathrm{Tr}\, Z_i L_i(\rho_i) + \sum_{i=1}^{m} \mathrm{Tr}\, Z_i R_i(\rho_{i-1}) \right\} \\
&= \max_{0\leq\rho_1,\ldots,\rho_m\leq I} \inf_{Z_1,\ldots,Z_m} \left\{ \mathrm{Tr}\, Z_1 R_1(\rho_0) + \sum_{i=1}^{m} \left(\mathrm{Tr}\, R_{i+1}(\rho_i) Z_{i+1} - \mathrm{Tr}\, L_i(\rho_i) Z_i\right) \right\} \\
&= \max_{0\leq\rho_1,\ldots,\rho_m\leq I} \inf_{Z_1,\ldots,Z_m} \left\{ \mathrm{Tr}\, Z_1 R_1(\rho_0) + \sum_{i=1}^{m} \mathrm{Tr}\left[\rho_i \left(R_{i+1}^d(Z_{i+1}) - L_i^d(Z_i)\right)\right] \right\}, \qquad (5.15)
\end{aligned}$$

where in the third line we introduced $Z_{m+1} \equiv E_B$ and $R_{m+1}(\rho) = \rho$. In the fourth line, we introduced the dual operators to L_i and R_i in the sense that $\mathrm{Tr}[L_i(\rho)Z] = \mathrm{Tr}[\rho L_i^d(Z)]$ and $\mathrm{Tr}[R_i(\rho)Z] = \mathrm{Tr}[\rho R_i^d(Z)]$ for all ρ and Z. These are easily constructed as follows: if $R_i(\rho) = \rho$ then $R_i^d(Z) = Z$, if $R_i(\rho) = U_i \rho U_i^{-1}$ then $R_i^d(Z) = U_i^{-1} Z U_i$, and if $R_i(\rho) = \mathrm{Tr}_j\, \rho$ then $R_i^d(Z) = Z \otimes I_j$, where the identity is inserted into the empty slot of qubit j. The expressions for L_i^d are defined similarly.

From the above equation it should be clear that

$$P_B^* \leq \mathrm{Tr}[Z_1 R_1(\rho_0)], \qquad (5.16)$$

for any Hermitian matrices $Z_1, \ldots, Z_m$ subject to the m constraints

$$R_{i+1}^d(Z_{i+1}) - L_i^d(Z_i) \leq 0, \qquad (5.17)$$

because under this constraint the second term is guaranteed to be non-positive for any set of $\{\rho_i\}$. We have therefore proven the following theorem:

Theorem 8. *Let $Z_1, \ldots, Z_m$ be any set of Hermitian matrices satisfying the m inequalities*

$$L_i^d(Z_i) \geq R_{i+1}^d(Z_{i+1}) \quad \textit{for} \quad i = 1, \ldots, m, \qquad (5.18)$$

where m, L_i^d, R_i^d, $Z_{m+1} \equiv E_B$ and ρ_0 are data associated with a coin-flipping protocol. The maximum probability that Bob can win such a coin-flip by cheating is bounded by

$$P_B^* \leq \mathrm{Tr}[Z_1 R_1(\rho_0)]. \qquad (5.19)$$

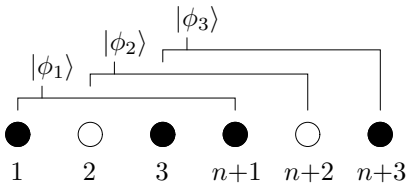


Figure 5.1: State ordering for $n = 3$. Black qubits are initially prepared by Alice.

Our goal in the next section will be to guess sets of matrices $Z_1, \ldots, Z_m$ satisfying the inequalities Eq. (5.18), and try to find a set that produces a good bound on P_B^* without worrying whether the bound is optimal.

5.3 Finding solutions to the dual problem

Continuing the analysis of the case where Alice is honest and Bob is cheating, we need to find the problem dual to $\mathcal{P}'$. The protocol $\mathcal{P}'$ can be thought of as having $m = n + 1$ elementary operations if we relax the definition somewhat to allow the receiving of n qubits in the last message as one step. Each elementary step consists of either sending or receiving a message, and unitaries are never used. The final measurement is done with $E_B = F_1$, $E_A = I - F_1$.

It will be useful to define a specific ordering for the qubits in Alice's Hilbert space. The intuition is to picture qubits as carried by particles in a lattice. When played honestly, the initial state will be prepared on $2n$ particles, some of which will be controlled by Alice, and some by Bob. Sending a qubit from Alice to Bob simply means that the particle will now be controlled by Bob rather than Alice. Alice's full Hilbert space at each step will be the ordered tensor product of the Hilbert spaces of all particles she controls in that step. Note that this does not restrict the power of a cheating player, who could have as many extra qubits as he wants that can interact with any particle under his control.

The ordering of the states will be as follows. The initial state is prepared so that $|\phi_i\rangle$ is carried by particles i and $n + i$. When n is even Alice starts off with all the odd particles in her possession whereas Bob has all the even particles. When n is odd Alice owns the odd particles between 1 and n inclusive, and the even particles between $n + 1$ and $2n$ inclusive. This is depicted in Fig. 5.1.

The first steps involve Alice sending qubit $n+1$, then receiving qubit 2, then sending qubit $n+3$, and so on. At the end of the first n messages Alice will control the first n qubits. The last step involves Alice taking possession of the other n qubits.

With these conventions, the primal problem reads:

$$\begin{aligned} \rho_i &= \mathrm{Tr}_{n+i}\,\rho_{i-1} && \text{for odd } i \le n, && (5.20)\\ \mathrm{Tr}_i\,\rho_i &= \rho_{i-1} && \text{for even } i \le n, && (5.21) \end{aligned}$$

plus one final equation

$$\mathrm{Tr}_{n+1,\ldots,2n}\,\rho_{n+1} = \rho_n. \tag{5.22}$$

With these conventions the dual problem involves finding m Hermitian matrices $Z_1, \ldots, Z_m$, where $m = n + 1$. When n is odd, all the matrices have dimension 2^n, whereas when n is even the matrix Z_{n+1} has dimension 2^n and the rest have dimension 2^{n-1}. They must satisfy the following equations

$$\begin{aligned} Z_i &\ge Z_{i+1} && \text{for odd } i \le n, && (5.23)\\ Z_i \otimes I_i &\ge Z_{i+1} \otimes I_{n+i+1} && \text{for even } i < n, && (5.24) \end{aligned}$$

where the subscript on the qubit identity matrices indicate into which slot it should be inserted. If n is even, we also need $Z_n \otimes I_n \ge Z_{n+1}$. Finally, in addition to the previous n inequalities we need to satisfy

$$Z_{n+1} \otimes I_{n+1,\ldots,2n} \ge Z_{n+2} \equiv F_1, \tag{5.25}$$

where the identity is inserted into the slot of the last n qubits. The goal is to choose the matrices in order to minimize $\langle\varphi|Z_1 \otimes I_{n+1}|\varphi\rangle$ where $|\varphi\rangle = |\phi_1\rangle \otimes |\phi_3\rangle \otimes \cdots$.

5.3.1 Choosing $Z_1, \ldots, Z_n$

Let $\beta = \langle\varphi|Z_1 \otimes I_{n+1}|\varphi\rangle$. To minimize this quantity, it is to our advantage to choose the Z_i matrices as small as possible in a sense to be discussed below. In particular, the optimal choice for Z_1 is simply to satisfy the equality $Z_1 = Z_2$. We can remove Z_1 from our equations and write

$$\beta = \langle\varphi|Z_2 \otimes I_{n+1}|\varphi\rangle = \langle\varphi_3|\,\mathrm{Tr}_{a_1}\,Z_2|\varphi_3\rangle, \tag{5.26}$$

where $|\varphi_3\rangle = |\phi_3\rangle \otimes |\phi_5\rangle \otimes \cdots$, and Tr_{a_1} denotes a weighted partial trace on the first qubit with weights a_1 and $1 - a_1$. For example, when acting on a matrix that only involves the first qubit

$$\mathrm{Tr}_{a_1}\,M = a_1\langle 0|M|0\rangle + (1 - a_1)\langle 1|M|1\rangle. \tag{5.27}$$

Note that in a slight abuse of notation, the subscript 1 in Tr_{a_1} indicates both which a_i is used and on which qubit the partial trace is performed.

The next inequality, which reads $Z_2 \otimes I_2 \geq Z_3 \otimes I_{n+3}$, is harder to satisfy, and in general equality cannot be achieved. However, we don't need to pay much attention to what happens in the subspace orthogonal to $|\varphi_3\rangle$, and we can in a sense sacrifice this subspace in order to obtain small entries in the subspace that we are interested in.

More specifically, let T_3 be the partial trace

$$T_3(M) = \mathrm{Tr}_{|\varphi_3\rangle}[(I_{1,2} \otimes |\varphi_3\rangle\langle\varphi_3|)M], \tag{5.28}$$

where the trace is taken only over qubits that are involved in $|\varphi_3\rangle$, that is, qubits 3, $n+3$, 5, $n+5$, and so on. The equation $Z_2 \otimes I_2 \geq Z_3 \otimes I_{n+3}$ requires $T_3(Z_2 \otimes I_2) \geq T_3(Z_3 \otimes I_{n+3})$, which is an equation involving only the first two qubits. We will begin by finding the optimal choice in this subspace.

Let us assume that $T_3(Z_3 \otimes I_{n+3})$ is a diagonal matrix with entries $x_{00}, x_{01}, x_{10}, x_{11}$. We want to choose $T_3(Z_2 \otimes I_2)$ to be as small as possible while still satisfying the inequality. However, because of the linearity of T_3, the matrix $T_3(Z_2 \otimes I_2)$ will have the form $M \otimes I_2$, for some one-qubit operator M. The equation $M \otimes I_2 \geq T_3(Z_3 \otimes I_{n+3})$ becomes

$$\begin{pmatrix} M_0 & 0 & M_c & 0 \\ 0 & M_0 & 0 & M_c \\ M_c^* & 0 & M_1 & 0 \\ 0 & M_c^* & 0 & M_1 \end{pmatrix} \geq \begin{pmatrix} x_{00} & 0 & 0 & 0 \\ 0 & x_{01} & 0 & 0 \\ 0 & 0 & x_{10} & 0 \\ 0 & 0 & 0 & x_{11} \end{pmatrix}, \tag{5.29}$$

where M_0, M_1 are the diagonal entries of M in the computational basis, and M_c is the complex off-diagonal entry.

Since we are trying to minimize $\langle\varphi_3| \mathrm{Tr}_{a_1} Z_2 |\varphi_3\rangle = \mathrm{Tr}_{a_1} M$, the best choice is to take $M_0 = \max(x_{00}, x_{01})$, $M_1 = \max(x_{10}, x_{11})$ and $M_c = 0$, which clearly satisfies the inequality. Notice that the maximum is taken over pairs of eigenvalues whose computational basis eigenvectors differ only in the second qubit. Symbolically, we shall write this as

$$M = \max_2 \left[T_3(Z_3 \otimes I_{n+3})\right], \tag{5.30}$$

where the operator max is defined only for diagonal matrices. The subscript 2 specifies that the maximum is to be taken over subspaces that differ in the second qubit.

The above discussion is only valid when $T_3(Z_3 \otimes I_{n+3})$ is diagonal, but we can impose this constraint on Z_3 (and the equivalent constraint on future Z_i), which is acceptable because we are

only looking for a solution of the inequalities, even if it is not the optimal solution.

Now if we could choose Z_2 to satisfy the full inequality, and still satisfy $T_3(Z_2) = M$ for the matrix chosen above we would have

$$\beta = \mathrm{Tr}_{a_1} \max_2 \left[T_3(Z_3 \otimes I_{n+3}) \right]. \tag{5.31}$$

The following lemma shows that it is possible to choose Z_2 so that we can get arbitrarily close to the above result. Because we will use β to upper bound P_B^*, it doesn't matter if it is an infimum, and therefore we can use the lemma to eliminate Z_2 in favor of the above expression.

Lemma 9. *Let T_3 be as above, and let H be a Hermitian matrix with finite eigenvalues. Given a Hermitian matrix M such that $M \otimes I_2 \geq T_3(H)$ and an $\epsilon > 0$, there exists a matrix M' such that $M' \otimes I_2 \geq H$ and $T_3(M') = M + \epsilon I$.*

Proof. Let $P = I_{1,2} \otimes |\varphi_3\rangle\langle\varphi_3|$ be the projector that was used in defining T_3. This divides the Hilbert space on which H acts into the direct sum of two parts, one invariant under P and one perpendicular to it. We write this as $\mathcal{H} = \mathcal{H}^{\|} \oplus \mathcal{H}^{\perp}$.

Let λ be the largest eigenvalue of $(I - P)H(I - P)$. Define the block diagonal matrix B as follows: the block acting on the space $\mathcal{H}^{\|}$ has the form $M \otimes I_2 + \epsilon I$, and the block acting on $\mathcal{H}^{\perp}$ has the form $(\lambda + y)I$ for some constant $y > 0$.

Let γ be the maximum over normalized states $|\Psi\rangle$,$|\Phi\rangle$ of $|\langle\Phi|PH(I - P)|\Psi\rangle|$. Then for any normalized state $|\Psi\rangle$ we have

$$\langle\Psi|(B - H)|\Psi\rangle \geq \epsilon\langle\Psi|P|\Psi\rangle + y\langle\Psi|(I - P)|\Psi\rangle - 2\gamma\sqrt{\langle\Psi|P|\Psi\rangle\langle\Psi|(I - P)|\Psi\rangle}. \tag{5.32}$$

As long as $y > \frac{\sqrt{\gamma}}{\epsilon}$, the expression is greater than zero, which implies $B > H$. It should be clear that B has the form $M' \otimes I_2$ and that the M' defined in this way satisfies $T_3(M') = M + \epsilon I$. □

The above lemma is used with $H = Z_3 \otimes I_{n+3}$ and letting $Z_2 = M'$. At this point the pattern begins to repeat itself. We can choose $Z_3 = Z_4$ and get

$$\beta = \mathrm{Tr}_{a_1} \max_2 \left[T_3(Z_4 \otimes I_{n+3}) \right] = \mathrm{Tr}_{a_1} \max_2 \left[\mathrm{Tr}_{a_3} T_5(Z_4) \right], \tag{5.33}$$

where T_5 is the partial trace using the state $|\varphi_5\rangle = |\phi_5\rangle \otimes |\phi_7\rangle \otimes \cdots$.

Using the lemma again we eliminate Z_4 in favor of Z_5:

$$\beta = \mathrm{Tr}_{a_1} \max_2 \left[\mathrm{Tr}_{a_3} \max_4 \left[T_5(Z_5 \otimes I_{n+5}) \right] \right], \tag{5.34}$$

where the expression is only valid if $T_5(Z_5 \otimes I_{n+5})$ is diagonal in the computational basis (which

will force $T_3(Z_3 \otimes I_{n+3})$ to be diagonal as well).

One may worry that repeated uses of the lemma will make Z_3 have arbitrarily large entries, which means that the lemma can no longer be used to eliminate Z_2. But the problems can be eliminated by taking the limits in the proper order, or more appropriately, by making sure that the coefficient y associated with Z_2 is much larger than the one associated with Z_4, which in turn needs to be much larger than the one associated with Z_6 and so on.

The process is repeated until in the last step, when n is odd, the innermost expression is of the form $T_n(Z_n \otimes I_{2n}) = T_n(Z_{n+1} \otimes I_{2n}) = \mathrm{Tr}_{a_n} Z_{n+1}$, yielding

$$\beta = \mathrm{Tr}_{a_1} \max_2 \left[\mathrm{Tr}_{a_3} \max_4 \left[\mathrm{Tr}_{a_5} \cdots \mathrm{Tr}_{a_n} \left[Z_{n+1} \right] \right] \right]. \tag{5.35}$$

When n is even, we have the special inequality $Z_n \otimes I_n \geq Z_{n+1}$, which is satisfied by choosing $Z_n = \max_n [Z_{n+1}]$, so that we get the same alternating expression, with the innermost operation a max:

$$\beta = \mathrm{Tr}_{a_1} \max_2 \left[\mathrm{Tr}_{a_3} \max_4 \left[\mathrm{Tr}_{a_5} \cdots \max_n \left[Z_{n+1} \right] \right] \right]. \tag{5.36}$$

Both of these formulas are valid only if Z_{n+1} is diagonal in the computational basis, which will make all the matrices of the form $T_i(Z_i)$ for odd i diagonal as well.

We are now left with the task of minimizing β as a function of Z_{n+1} with the constraint that Z_{n+1} must be real and diagonal in the computational basis and must satisfy the inequality $Z_{n+1} \otimes I \geq F_1$.

In fact, when Z_{n+1} is diagonal the inequality can be simplified further. In the qubit ordering we have chosen, the final state of the protocol right before measurement should be $|\psi\rangle = |\phi_1\rangle_{1,n+1} \otimes |\phi_2\rangle_{2,n+2} \otimes \cdots \otimes |\phi_n\rangle_{n,2n}$ where we have explicitly listed the location of each qubit. Therefore $F_1 = 2E_1 \otimes E_1 |\psi\rangle\langle\psi| E_1 \otimes E_1$ has support only on the 2^n dimensional subspace spanned by states where qubits i and $i+n$ are equal for all i. The constraint $Z_{n+1} \otimes I \geq F_1$ need only be checked in this subspace where it takes the form

$$Z_{n+1} \geq |\xi_B\rangle\langle\xi_B|, \tag{5.37}$$

where

$$|\xi\rangle = \left(\sqrt{a_1}|0\rangle + \sqrt{1-a_1}|1\rangle\right) \otimes \left(\sqrt{a_2}|0\rangle + \sqrt{1-a_2}|1\rangle\right) \otimes \cdots \otimes \left(\sqrt{a_n}|0\rangle + \sqrt{1-a_n}|1\rangle\right), \tag{5.38}$$

and $|\xi_B\rangle = \sqrt{2} E_1 |\xi\rangle$, which is correctly normalized by the factor $\sqrt{2}$ if the coin is fair when both players are honest.

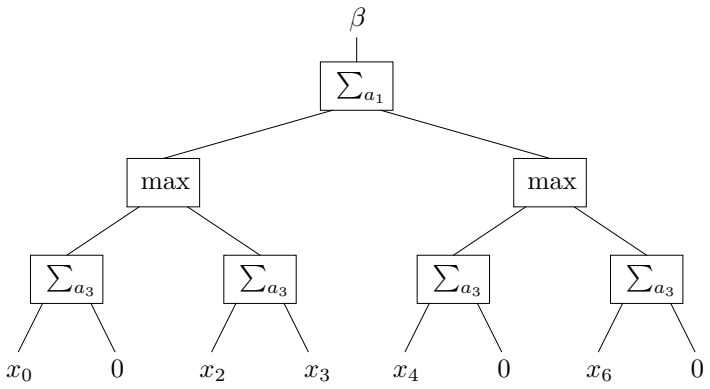


Figure 5.2: Cheating Bob's Sum-Max Tree.

5.3.2 Example $n = 3$

At this point an example would probably be helpful. We shall look at the case $n = 3$:

$$|\xi_B\rangle = \sqrt{2} \quad \Big(\quad \sqrt{a_1 a_2 a_3}|000\rangle + \sqrt{a_1(1-a_2)a_3}|010\rangle + \sqrt{a_1(1-a_2)(1-a_3)}|011\rangle \\ + \sqrt{(1-a_1)a_2 a_3}|100\rangle + \sqrt{(1-a_1)(1-a_2)a_3}|110\rangle \Big), \tag{5.39}$$

and the matrix Z_4 can be chosen as

$$Z_4 = \begin{pmatrix} x_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.40}$$

where the top row corresponds to $|000\rangle$, the second one to $|001\rangle$ and so on. The entries along the diagonal of Z_4 outside of the support of $|\xi_B\rangle\langle\xi_B|$ have already been set to zero, which should be expected for all optimal solutions. Otherwise, any set of $\{x_i\}$ that satisfies $Z_4 \geq |\xi_B\rangle\langle\xi_B|$ is a valid solution of the dual problem. The corresponding bound can be calculated from these variables using Eqs. (5.35) or equivalently by evaluating the tree depicted in Fig. 5.2.

The tree is evaluated as follows: each node has a value that is either the maximum or the weighted sum of the nodes below it. The weighted sum is just a_i times the value of the left descendant plus $1 - a_i$ times the value of the right descendant. The value of the root node corresponds to β and is

an upper bound on P_B^*. We shall call trees of this form Sum-Max trees.

Sum-Max trees appear naturally when analyzing classical protocols for coin-flipping. The basic idea is that these protocols can be described as a sequence of public random bits, with the first one announced by Alice, then the second one by Bob and so on. At the end both parties look at the sequence of bits and determine the outcome of the coin-flip. The whole protocol can be described as a binary tree, with Alice's bits choosing the path at the odd depth nodes and Bob's bits controlling the rest.

A player attempting to cheat in such a protocol will not output random bits but will instead choose the path that maximizes his chances of winning at each node. If we put ones and zeros in the leaf nodes corresponding to a win or loss, the maximum probability with which the cheater can win is given by evaluating the corresponding Sum-Max tree.

If we ignore the cheat detection stage in the protocol, then it can be described completely classically. Its Sum-Max tree would be the same as Fig. 5.2, except that all the variables would be replaced by the number one. This can easily be seen from our formalism because the only effect of removing the last round is to force Z_{n+1} to equal E_1, which is diagonal with ones in place of the variables $\{x_i\}$.

It is well known that in the classical case, one party can always fully bias the coin in their favor. However, in the quantum case with cheat detection, because the leaves of Sum-Max tree are less restricted, there is the possibility of obtaining a stronger bound on the amount of cheating.

The analysis so far has been of the case when Alice is honest and Bob is cheating. The case of Bob honest and Alice cheating is almost identical, though. The first major difference, is that this case has to be analyzed from Bob's perspective, so that the odd messages consist of receiving a qubit and the even ones involve sending a qubit. This has the effect of switching sums with maxes and vice versa. The other difference is the final state that Bob will use to verify that Alice is not cheating. For $n = 3$ it has the form

$$|\xi_A\rangle = \sqrt{2}\big(\sqrt{a_1 a_2 (1-a_3)}|001\rangle + \sqrt{(1-a_1)a_2(1-a_3)}|101\rangle + \sqrt{(1-a_1)(1-a_2)(1-a_3)}|111\rangle\big). \tag{5.41}$$

The maximum probability with which Alice can cheat, P_A^*, is bounded above by α, calculated from the Max-Sum tree in Fig. 5.3, where the leaves are the diagonal elements of Z_4 and must satisfy $Z_4 \geq |\xi_A\rangle\langle\xi_A|$.

5.3.3 Finding the optimal Z_{n+1}

Returning to the case of honest Alice, we need to finish the general case by choosing a matrix Z_{n+1} in order to obtain an expression for β in terms of the parameters $a_1, \ldots, a_n$. Recall that we have restricted our analysis to matrices Z_{n+1} that are diagonal in the computational basis. We shall now

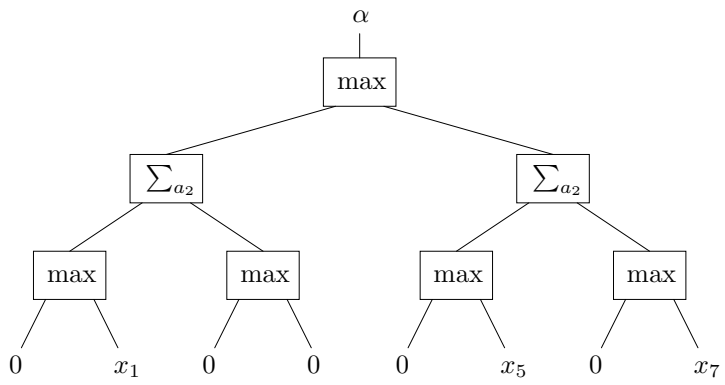


Figure 5.3: Cheating Alice's Sum-Max Tree.

search for the minimum value of β consistent with this choice.

Let $x_1, \ldots, x_{2^n}$ be the diagonal entries of Z_{n+1}. We will, as in the example above, set the variables to zero when their corresponding basis vector is orthogonal to $|\xi_B\rangle$, which will leave around half of the variables. We also wish to work in the subspace where the two values entering a max node in the Sum-Max tree are equal. For example, this is the space consistent with $a_3 x_0 = a_3 x_2 + (1-a_3)x_3$ and $x_4 = x_6$ in the example above. The only potential problem exists at the lowest level of max nodes, where a zero can be entering the node. For the following we will assume that n is odd, which eliminates this problem. The even case will be derived from the odd case below.

Working in this subspace we can replace all the maximums with weighted sums with any weight of our choice. In this situation, β can be calculated as a weighted trace of Z_{n+1}. That is, there exist diagonal matrices W such that $\beta = \mathrm{Tr}(Z_{n+1}W)$ for any Z_{n+1} in this subspace. For example, a valid choice for W is the diagonal part of $|\xi\rangle\langle\xi|$, which replaces the max nodes at each level i by the weighted sum using a_i. That is

$$\mathrm{Tr}[Z_{n+1}\,\mathrm{diag}(|\xi\rangle\langle\xi|)] = \mathrm{Tr}_{a_1}\,\mathrm{Tr}_{a_2}\,\mathrm{Tr}_{a_3}\cdots\mathrm{Tr}_{a_n}\,Z_{n+1}. \tag{5.42}$$

This will turn out to be the wrong choice for W but it gets us closer to the following lemma:

Lemma 10. *Let $|\Psi\rangle$ be a state, not necessarily normalized, and let D be the diagonal part of $|\Psi\rangle\langle\Psi|$. Let $E = E^2$ be a diagonal projector. The minimum of $\mathrm{Tr}(ZD)$ over diagonal real matrices Z, subject to the constraint $Z \geq 2E|\Psi\rangle\langle\Psi|E$, is given by $2|\langle\Psi|E|\Psi\rangle|^2$ and is attained by*

$$Z = 2\langle\Psi|E|\Psi\rangle E. \tag{5.43}$$

Proof. Because Z is diagonal we can write $\mathrm{Tr}(ZD)$ as $\langle\Psi|Z|\Psi\rangle$. Clearly if $Z \geq 2E|\Psi\rangle\langle\Psi|E$ then $\mathrm{Tr}(ZD) = \langle\Psi|Z|\Psi\rangle \geq 2|\langle\Psi|E|\Psi\rangle|^2$. It is also attainable using $Z = 2\langle\Psi|E|\Psi\rangle E$, which satisfies

the inequality constraint because by Cauchy-Schwarz $\langle\Phi|E|\Psi\rangle\langle\Psi|E|\Phi\rangle \leq \langle\Psi|E|\Psi\rangle\langle\Phi|E|\Phi\rangle$ for any $|\Phi\rangle$. □

Because $|\xi_B\rangle\langle\xi_B| = 2E_1|\xi\rangle\langle\xi|E_1$, we are almost in the situation covered by the above lemma. Unfortunately, we are only maximizing over the space consistent with the entries to every max node being equal (while keeping the zero entries in Z_{n+1} equal to zero), so in general Z_{n+1} proportional to E_1 is not a valid solution. However, by rescaling the variables, we can get to the situation where this subspace contains E_1, and therefore the above lemma is useful.

More specifically, let S be a diagonal positive matrix. Define $Z'_{n+1} = \sqrt{S}Z_{n+1}\sqrt{S}$. We now can minimize $\beta = \mathrm{Tr}(Z'_{n+1}\sqrt{S^{-1}}W\sqrt{S^{-1}})$ subject to the constraint $Z'_{n+1} \geq \sqrt{S}|\xi_B\rangle\langle\xi_B|\sqrt{S}$. We would like to choose S so that $Z'_{n+1} = E_1$ is a valid solution, that is, when $Z_{n+1} = \sqrt{S^{-1}}E_1\sqrt{S^{-1}} = S^{-1}E_1$ is put into the Sum-Max tree, the pair of values entering each max node are equal. We also need to define W as the diagonal part of $S|\xi\rangle\langle\xi|S$. For this to be valid, we must show that we can compute β as a function of Z_{n+1} by the expression $\mathrm{Tr}(Z_{n+1}W)$ for every Z_{n+1} consistent with the original requirements. If these two conditions are satisfied, though, then the lemma tells us that

$$\beta = 2\left|\langle\xi|SE_1|\xi\rangle\right|^2, \tag{5.44}$$

where we used the fact that both S and E_1 are diagonal in the computational basis.

We begin by analyzing as an example the case of $n = 3$ depicted in Fig. 5.2. Define $e_i = \langle i|E_1|i\rangle$, which takes the values zero or one. Similarly, let $s_i = \langle i|S|i\rangle$. We construct S so that

$$s_0 = s_1 = \sigma_0\sigma_{0L}, \tag{5.45}$$

$$s_2 = s_3 = \sigma_0\sigma_{0R}, \tag{5.46}$$

$$s_4 = s_5 = \sigma_1\sigma_{1L}, \tag{5.47}$$

$$s_6 = s_7 = \sigma_1\sigma_{1R}. \tag{5.48}$$

The factors σ_0, σ_{0L} and σ_{0R} should be thought of as being associated with the left max node. The first one is a normalization factor, and the other two will be used to balance the values of the left and right descendants. Similarly, the other three variables are associated with the right max node.

To satisfy the first constraint, we set $Z_{n+1} = S^{-1}E_1$, or equivalently, $x_i = s_i^{-1}e_i$. Note that the e_i factor will force the appropriate x_i variables to be zero. We focus on the left max node. The value entering through the left descendant is

$$a_3x_0 + (1-a_3)x_1 = \frac{a_3e_0 + (1-a_3)e_1}{\sigma_0\sigma_{0L}}, \tag{5.49}$$

whereas entering on the right side is

$$a_3 x_2 + (1-a_3)x_3 = \frac{a_3 e_2 + (1-a_3)e_3}{\sigma_0 \sigma_{0R}}. \tag{5.50}$$

For the two values to be equal, we can choose $\sigma_{0L} = a_3 e_0 + (1-a_3)e_1$ and $\sigma_{0R} = a_3 e_2 + (1-a_3)e_3$. Similarly, the constraint at the other max node can be met by choosing $\sigma_{1L} = a_3 e_4 + (1-a_3)e_5$ and $\sigma_{1R} = a_3 e_6 + (1-a_3)e_7$.

Now we need to check the constraint on $W = \mathrm{diag}(S|\xi\rangle\langle\xi|S)$. Now $\mathrm{Tr}(Z_{n+1}W)$ can be described as the Max-Sum tree in Fig. 5.2, with the max nodes replaced by sums. Focusing again on the left max node, it adds $a_2\sigma_0^2\sigma_{0L}^2$ of its left descendant plus $(1-a_2)\sigma_0^2\sigma_{0R}^2$ of the right descendant. We need these quantities to sum to one, and therefore $\sigma_0 = [a_2\sigma_{0L}^2 + (1-a_2)\sigma_{0R}^2]^{-1/2}$. Similarly, we choose $\sigma_1 = [a_2\sigma_{1L}^2 + (1-a_2)\sigma_{1R}^2]^{-1/2}$ to normalize the sum replacing the right max node.

Now we can finally evaluate $\beta = 2\,|\langle\xi|SE_1|\xi\rangle|^2$. This can also be represented by a tree similar to the one in Fig. 5.2, with the max nodes replaced by different sums as follows: the left max evaluates to $a_2\sigma_0\sigma_{0L}$ times the input from the left plus $(1-a_2)\sigma_0\sigma_{0R}$ times the right input. But the left and right inputs are respectively equal to $a_3 e_0 + (1-a_3)e_1 = \sigma_{0L}$ and $a_3 e_2 + (1-a_3)e_3 = \sigma_{0R}$, so the node evaluates to

$$a_2\sigma_0\sigma_{0L}^2 + (1-a_2)\sigma_0\sigma_{0R}^2 = \sqrt{a_2\sigma_{0L}^2 + (1-a_2)\sigma_{0R}^2}, \tag{5.51}$$

which can be though of as the weighted root mean square of the values of the two descendant nodes. The same thing happens at the right max node. The complete expression then becomes

$$\begin{aligned}
\beta &= 2\Big\{a_1\sqrt{a_2\left[a_3e_0+(1-a_3)e_1\right]^2 + (1-a_2)\left[a_3e_2+(1-a_3)e_3\right]^2} \\
&\qquad +(1-a_1)\sqrt{a_2\left[a_3e_4+(1-a_3)e_5\right]^2 + (1-a_2)\left[a_3e_6+(1-a_3)e_7\right]^2}\Big\}^2 \\
&= 2\Big\{a_1\sqrt{a_2a_3^2+(1-a_2)} + (1-a_1)a_3\Big\}^2.
\end{aligned} \tag{5.52}$$

The above has the shorthand notation given by

$$\beta = 2\left(\mathrm{Tr}_{a_1}\,\mathrm{RMS}_{a_2}\,\mathrm{Tr}_{a_3}\,E_1\right)^2 \tag{5.53}$$

where we define RMS_{a_i} only on diagonal matrices, as a weighted root mean square of eigenvalues whose basis vectors differ only on qubit i. This is in the same spirit as Tr_{a_i}, which does a regular weighted average.

For completeness, we also give the expression for α when $n = 3$, which can be obtained from the

formulas derived below:

$$\alpha = 2(1-a_3)\left(a_1 a_2^2 + (1-a_1)\right). \tag{5.54}$$

The general case is almost identical. Consider the original Sum-Max tree for a given odd n. Let $\mathcal{M}$ be the set of max nodes of the original tree, that is, the set of binary nodes with odd depth (where we define the depth of the root node as zero). For each $\mu \in \mathcal{M}$ we introduce three variables: σ_μ, $\sigma_{\mu L}$ and $\sigma_{\mu R}$, which are to be associated with the corresponding max node. We define the components of S in terms of these variables as follows: the value of s_j, which is to be associated with leaf j, is given as the product of $\sigma_\mu \sigma_{\mu L}$ for every node μ of which j is a left descendant, times the product of $\sigma_\mu \sigma_{\mu R}$ for every node μ of which j is a right descendant.

The conditions on W are always satisfied by choosing $\sigma_\mu = [a_\mu \sigma_{\mu L}^2 + (1-a_\mu)\sigma_{\mu R}^2]^{-1/2}$ for every $\mu \in \mathcal{M}$, where in a slight abuse of notation a_μ is the parameter associated with μ (i.e., $a_\mu = a_{d(\mu)+1}$, where $d(\mu)$ is the depth of node μ). The next condition that needs to be checked is that, when the diagonal entries of $S^{-1}E_1$ are placed on the leaves of the original tree, the left and right descendants of each max node must be equal. We shall choose the values of $\sigma_{\mu L}$ and $\sigma_{\mu R}$ in order to guarantee this, in a process that begins at the lower nodes and proceeds upwards. At the lowest level they are chosen so that $\sigma_{\mu L} = a_n e_{\mu LL} + (1-a_n)e_{\mu LR}$, where $e_{\mu LL}$ and $e_{\mu LR}$ are respectively the left and right leaf values under the left child of node μ. Similarly, we also set $\sigma_{\mu R} = a_n e_{\mu RL} + (1-a_n)e_{\mu RR}$, in terms of the leaves under the right leg of node μ. Having made such a choice, the value of node μ in the Sum-Max tree equals σ_μ^{-1} (up to multiplication by σ factors from higher nodes). For every other $\mu \in \mathcal{M}$ that is not associated with the lowest level max nodes, we set $\sigma_{\mu L} = a_{d(\mu)+2}\sigma_{\mu'}^{-1} + (1-a_{d(\mu)+2})\sigma_{\mu''}^{-1}$, where μ' and μ'' are respectively the left and right max nodes located under the left leg of max node μ. With an equivalent choice for $\sigma_{\mu R}$, the value entering either leg of max node μ will be σ_μ^{-1} (up to σ factors from higher nodes) and the second condition will be satisfied.

Finally we need to evaluate $\langle \xi | S E_1 | \xi \rangle$. Once again this is to be done as a tree, with binary nodes corresponding to sums. The factors of σ_μ, $\sigma_{\mu L}$ and $\sigma_{\mu R}$ can be moved up the tree so that the node μ becomes the weighted sum of $a_\mu \sigma_\mu \sigma_{\mu L}$ times the left descendant plus $(1-a_\mu)\sigma_\mu \sigma_{\mu R}$ times the right descendant. All that remains on the leaves are the zero or one values of E_1. The tree can be evaluated recursively from the bottom up, in which case it is easy to see that the value of a non-root binary node outside of $\mathcal{M}$ (i.e., one of the original sum nodes), is equal to either $\sigma_{\mu L}$ or $\sigma_{\mu R}$, where μ denotes its parent node, depending on whether it is a left or right descendant, respectively. On the other hand, for $\mu \in \mathcal{M}$, node μ has value $a_\mu \sigma_\mu \sigma_{\mu L}^2 + (1-a_\mu)\sigma_\mu \sigma_{\mu R}^2 = \sigma_\mu^{-1}$. The root node (which originally was a sum node) has value $a_1 \sigma_{\mu'}^{-1} + (1-a_1)\sigma_{\mu''}^{-1}$, where μ' and μ'' are respectively its left and right descendants. The square of this quantity multiplied by two is the the value of our

upper bound, which can be expanded using the definitions for the σ variables to obtain:

$$\beta = 2\left(\mathrm{Tr}_{a_1}\,\mathrm{RMS}_{a_2}\,\mathrm{Tr}_{a_3}\,\mathrm{RMS}_{a_4}\cdots\mathrm{Tr}_{a_n}\,E_1\right)^2, \tag{5.55}$$

valid only for n odd, to which the above discussion was restricted. Though the case of even n can also be found similarly, it can be obtained from the above formula by the following observation: the protocol with n steps and constants $a_1,\ldots,a_n$ is equivalent to the protocol with $n+1$ steps and constants $a'_1,\ldots,a'_{n+1}$ with $a'_{n+1}=0$ and $a'_i=a_i$ for all $1\le i\le n$. Furthermore, if E_1 is the projector associated with the n step protocol, and E'_1 is the projector associated with the $n+1$ step protocol, the two matrices are related by $\mathrm{Tr}_{a_{n+1}=0}\,E'_1=\mathrm{RMS}_{a_{n+1=0}}\,E'_1=E_1$. Therefore, for even n we have

$$\beta = 2\left(\mathrm{Tr}_{a_1}\,\mathrm{RMS}_{a_2}\,\mathrm{Tr}_{a_3}\,\mathrm{RMS}_{a_4}\cdots\mathrm{RMS}_{a_n}\,E_1\right)^2. \tag{5.56}$$

All that remains is to analyze the case where Bob is honest and Alice is cheating. Though this could be analyzed using the methods presented in this section, we can exploit further symmetries of the protocols to obtain the result. In particular, the protocol with n steps and constants $a_1,\ldots,a_n$ is equivalent to the protocol with $n+1$ steps and constants $a'_1,\ldots,a'_{n+1}$ with $a'_1=1$, $a'_{i+1}=a_i$ for all $1\le i\le n$, and Alice's and Bob's roles switched. Furthermore, if E_0 is a projector associated with the n message protocol, and E'_1 the projector associated with the $n+1$ step protocol, the two matrices are related by $\mathrm{Tr}_{a_1=1}\,E'_1=E_0$. We also need to use the fact that $\mathrm{Tr}_{a_1=1}$ commutes through all the RMS operators because it is a projector onto a subspace rather than a trace. Combining all the results, we have proven the following theorem:

Theorem 11. *In the protocol described in Sec. 5.1, Alice's and Bob's ability to win by cheating are upper bounded by*

$$\begin{aligned} P_A^* &\le \alpha = 2\left(\mathrm{RMS}_{a_1}\,\mathrm{Tr}_{a_2}\,\mathrm{RMS}_{a_3}\,\mathrm{Tr}_{a_4}\cdots\mathrm{Tr}_{a_n}\,E_0\right)^2, \\ P_B^* &\le \beta = 2\left(\mathrm{Tr}_{a_1}\,\mathrm{RMS}_{a_2}\,\mathrm{Tr}_{a_3}\,\mathrm{RMS}_{a_4}\cdots\mathrm{RMS}_{a_n}\,E_1\right)^2, \end{aligned}$$

when n is even, and by

$$\begin{aligned} P_A^* &\le \alpha = 2\left(\mathrm{RMS}_{a_1}\,\mathrm{Tr}_{a_2}\,\mathrm{RMS}_{a_3}\,\mathrm{Tr}_{a_4}\cdots\mathrm{RMS}_{a_n}\,E_0\right)^2 \\ P_B^* &\le \beta = 2\left(\mathrm{Tr}_{a_1}\,\mathrm{RMS}_{a_2}\,\mathrm{Tr}_{a_3}\,\mathrm{RMS}_{a_4}\cdots\mathrm{Tr}_{a_n}\,E_1\right)^2, \end{aligned}$$

when n is odd.

Note that all the above formulas are valid only when the parameters $a_1,\ldots,a_n$ are chosen so

that the honest probability of winning is 1/2. This is the source of the factor of 2 appearing in front of the expressions, which could be replaced by one over the honest probability of winning for more general scenarios.

In fact, the above formulas are even more general, as they apply to any choice of $\{E_0, E_1\}$ as long as they are diagonal projectors. In such a case, the symmetries that were used above are no longer valid, but direct computations should lead to the same formulas. Such a computation will be carried out in the next chapter.

5.4 Choosing $a_1, \ldots, a_n$

Recall that any choice of parameters $a_1, \ldots, a_n$ subject to the constraint $\langle\psi|E_i \otimes E_i|\psi\rangle = \frac{1}{2}$ describes a valid quantum weak coin-flipping protocol. Furthermore, we have an analytic upper bound on the bias given by

$$\epsilon \leq \max(\alpha, \beta) - \frac{1}{2}, \tag{5.57}$$

expressed as a function of these parameters. Now we need to chose values for the parameters, which ideally should be selected to produce a bias as small as possible.

Because the expressions for α and β are complicated, we shall employ numerical minimization to find optimal values for the parameters for certain small values of n. The values obtained will prove the existence of protocols with the quoted biases.

Fortunately, the quality of the minimization does not need to be verified. For example, it would be perfectly acceptable if rather than finding the true minimum, we only found a local minimum, or even if the outputted parameters did not constitute a minimum at all. All that is needed is for the parameters to satisfy the constraint and produce the quoted bias when substituted into the expressions for α and β.

Note there is an issue with the constraint because it can only be satisfied to the accuracy with which the parameters are specified. That is, when the parameters are described to finite accuracy, the coin will not be exactly fair when both players are honest. Of course, this is to be expected for any practical implementation of the protocol. However, we also claim that from a theoretical perspective, there are parameters close to the ones quoted that satisfy the constraint exactly and produce protocols with a bias equal to the quoted numbers to the given accuracy.

In addition to the constraint $\langle\psi|E_i \otimes E_i|\psi\rangle = \frac{1}{2}$, the minimizations were carried out with the constraint $\alpha = \beta$. For $n = 3$, we find $\alpha = \beta \simeq 0.69905$ at $a_1 = 0.74094$, $a_2 = 0.479696$ and $a_3 = 0.186312$. Though strictly speaking we should write that there exists a protocol with $n = 3$ and $\epsilon \leq 0.1991$, for simplicity we will write $\epsilon = 0.199$, which is understood to be correct only up to

the given accuracy.

Though so far we have only derived an upper bound on the bias, it is fair to claim that there exist protocols with a bias equal to the upper bound because protocols can always be weakened. For example, in the first round with some probability Alice decides to let Bob determine the outcome of the coin, otherwise, with some probability Bob decides to let Alice determine the coin outcome and, if none of these events take place, then the protocol is started normally.

Continuing with the analysis of small n, we find: for $n = 4$ we get $\epsilon \simeq 0.1957$, for $n = 6$ we get $\epsilon \simeq 0.1937$, for $n = 8$ we get $\epsilon \simeq 0.1931$, and for $n = 10$ we get $\epsilon \simeq 0.1927$. For completeness, we list the values used for $n = 8$: $a_1 = 0.680706$, $a_2 = 0.43281$, $a_3 = 0.323787$, $a_4 = 0.264123$, $a_5 = 0.224377$, $a_6 = 0.197997$, $a_7 = 0.177191$ and $a_8 = 0.0834815$.

To analyze larger values of n one needs to change the exponential formulas for α and β into expressions that can be computed in a time linear in n. This can be done because of the special structure of our choice of $\{E_0, E_1\}$. Because the construction is not central to the claims of this chapter, we shall only sketch the proof.

The expression $\mathrm{Tr}_{a_1} \mathrm{RMS}_{a_2} \cdots \mathrm{Tr}_{a_n} E_1$ can be computed using a tree, with binary nodes alternating between weighted sums and weighted root mean square. The leaves at the lowest level take only the two values zero and one, which should be thought of as a high value and a low value. Entering into the lowest sum nodes are only two possibilities: both entries are the high value, or the left descendant is high and the right one low. Therefore, if we assign values to the sum nodes they will only take on two values, a high value (equal to the previous high value) and a low value (equal to the weighted sum $a_n\mathrm{High} + (1 - a_n)\mathrm{Low}$). Entering into the next level RMS node there are also only two possibilities: two low values, or a high (from the right) and a low (from the left). This structure repeats all the way up to the root node, which is computed as the weighted sum of the high and low of the previous node. The original expression can therefore be calculated from a sequence of pairs of values, which depend only on the previous pair of the sequence (a precise formulation will be given in the next section). The value of the constraint can also be computed in a similar way by replacing the weighted root mean square average with a weighted linear average.

Using these linear time formulas, it is possible to compute α and β for large values of n given a specific functional form for a_k as a function of $k \leq n$. A theoretically pleasant, though non-optimal choice, is $a_k = 1/k$ for n even. Recall that $\{E_0, E_1\}$ can be described by the process whereby the qubits are examined starting from qubit n to qubit 1, and the first zero that is found determines the winner. With $a_k = 1/k$, the probability that qubit k needs to be examined is k/n, and therefore each qubit determines the outcome with a probability of $1/n$. The problem with this choice is that Bob's probability of winning given that qubit k needs to be examined keeps oscillating between $1/2$ and numbers greater than $1/2$. That is why for $a_k = 1/k$, Bob has the ability to cheat a lot, whereas Alice is more restricted. With some work, the problem could be fixed by adjusting the values of a_k

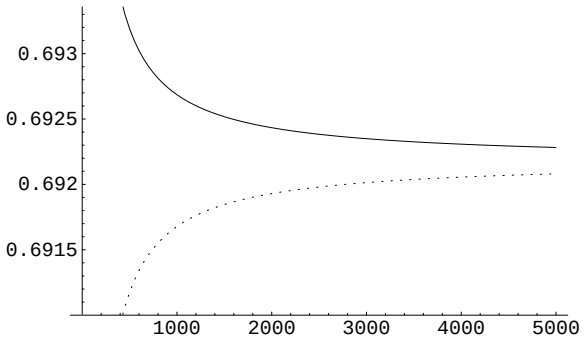


Figure 5.4: Plot of the bounds on cheating as a function of n, for parameters $a_k = 1/k$. The solid line corresponds to β and dotted line to α. These bounds are only valid for even n.

for $k \sim 1$ and $k \sim n$.

The values of α and β as a function of even n for $a_k = 1/k$ have been plotted in Fig. 5.4. The upper solid line corresponds to β and the lower dotted line to α. For odd n we would have to use $a_k = 1/(k+1)$ to satisfy the constraint, and this would switch the values of α and β.

For large n, the graphs converge towards 0.6922, or a bias of $\epsilon = 0.1922$. The same behavior occurs with many other reasonable choices for a_k as a function of k.

5.5 0.192 revisited

In this section we shall derive an analytical expression that corresponds to the bias of 0.192, that is, the bias of the protocol obtained in the $n \to \infty$ limit. Since the protocol with bias 0.192 has been superseded by the results in the next chapter, we shall only sketch the proof. Nonetheless, we hope that the techniques used in deriving this expression, which are rather different than the approach taken in the rest of the thesis, will be of use in some future applications.

As discussed above, the expression for the bias given a set of variables $a_1, \ldots, a_n$ can be computed using the following sequences: set $H_n = 1$ and $L_n = 0$ and define

$$H_k = \sqrt{a_{k+1} L_{k+1}^2 + (1 - a_{k+1}) H_{k+1}^2}, \tag{5.58}$$

$$L_k = L_{k+1}, \tag{5.59}$$

for even $k \geq 0$, and

$$H_k = H_{k+1}, \tag{5.60}$$

$$L_k = a_{k+1} H_{k+1} + (1 - a_{k+1}) L_{k+1}, \tag{5.61}$$

for odd $k \geq 0$. The value of P_A^* is then given by $2H_0^2$, as long as the variables a_i satisfy the fair coin constraint.

The sequence is defined so that H decreases and L increases with decreasing k. At every step the condition $1 \geq H_k \geq L_k \geq 0$ holds. For good choices of a_k the two sequences will approach each other and H_0 will be close to L_0.

A good sequence of parameters will also have a_k small for large k. For k small, a_k can be larger as long as $a_k(H_k - L_k)$ remains small. In such a case, we can use the expansion

$$H_k \simeq H_{k+1} - a_{k+1}\frac{H_{k+1}^2 - L_{k+1}^2}{2H_{k+1}}, \tag{5.62}$$

for even k.

Furthermore, if a_k is slowly varying, we can replace it with a continuous function $a(k)$, and the above computation can be approximated by the coupled differential equations

$$\frac{dH}{dk} = \frac{a(k)}{2}\frac{H^2 - L^2}{2H}, \tag{5.63}$$

$$\frac{dL}{dk} = -\frac{a(k)}{2}(H - L), \tag{5.64}$$

where now H and L are treated as functions of the continuous variable $k \in [0, n]$. An extra factor of $1/2$ was picked up on the right hand side of the above equations because H and L only get updated every other integer in the discrete sequence.

Of course, we are only concerned with the convergence point where $H \simeq L$. In the limit $n \to \infty$, and for appropriate $a(k)$, the two expressions will converge to the same point $H_0 = L_0$. To study the convergence point we can study H as a function of L, which satisfies the differential equation

$$\frac{dH}{dL} = -\frac{H + L}{2H}. \tag{5.65}$$

Surprisingly, the function $a(k)$ drops out of the above expression, which means it only controls the rate of convergence but not the final point of convergence (assuming it satisfies the requirements discussed above).

The differential equation is invariant under simultaneous rescaling of H and L, and therefore becomes separable under the change of variables $H \to H/L$. Its solutions have the form

$$\log\left(H^2 + \frac{1}{2}LH + \frac{1}{2}L^2\right) + \frac{2}{\sqrt{7}}\arctan\frac{\sqrt{7}L}{4H + L} = const. \tag{5.66}$$

The initial condition for the differential equation is $H(L = 0) = 1$, which corresponds to the initial starting point when $k \to \infty$. Applying the initial condition we obtain $const = 0$. We are

interested in the point where H and L converge, that is, the value L_0 such that $H(L_0) = L_0$:

$$\log 2L_0^2 = -\frac{2}{\sqrt{7}} \arctan \frac{\sqrt{7}}{5}. \tag{5.67}$$

From this value we can obtain $P_A^* = 2H_0^2 = 2L_0^2$. When a_k varies slowly enough and meets the other requirements above, it also produces a protocol that is arbitrarily close to being symmetric between Alice and Bob. This guarantees that the fair coin constraint is satisfied and that $P_B^* = P_A^*$, hence

$$\begin{aligned} P_A^* = P_B^* &= \mathrm{Exp}\left[-\frac{2}{\sqrt{7}} \arctan \frac{\sqrt{7}}{5}\right] \\ &\simeq 0.692181687, \end{aligned} \tag{5.68}$$

which corresponds to the bias $\epsilon \simeq 0.192$ found numerically.

Chapter 6

A large family of quantum weak coin-flipping protocols

The purpose of this chapter is to define and study a large family of protocols for quantum weak coin-flipping that are based on classical public-coin coin-flipping protocols. The family will contain all protocols discussed in the previous chapter. In particular, we will prove that the optimal protocol in this family has a bias of 1/6, though such a bias can only be reached in the limit of arbitrarily large number of messages. Because our lower bound analysis is constructive, we shall give explicit descriptions of protocols with biases that are arbitrarily close to 1/6.

Note that whereas the usual definition of coin-flipping requires the final bit output by Alice and Bob to be uniformly random when both are honest, in this chapter we shall consider a slightly more general scenario. Let P_A be Alice's probability of winning when both Alice and Bob play honestly, and let P_B be Bob's probability of winning when both play honestly. In this chapter we shall allow any value of $P_A \in [0, 1]$, with $P_B = 1 - P_A$, rather than the usual case of $P_A = P_B = 1/2$.

The chapter is organized as follows: Sec. 6.1 describes some of our notation concerning tree variables and will introduce the theorem relating classical coin games to quantum protocols for weak coin-flipping. Then in Sec. 6.2 we shall prove the theorem, while giving a more explicit description of the protocols themselves. Much of the work needed is simply a generalization of the results of the previous chapter.

The main new results of this chapter are the proof of lower bounds for the bias in Sec. 6.3 and the description of matching protocols in Sec. 6.4. These sections will identify the parameters needed to construct a quantum weak coin-flipping protocol with bias 1/6.

6.1 Notation

Throughout this chapter we shall make ample use of binary trees. All trees henceforth will be composed exclusively of binary nodes and leaves, and the leaves will all be located at the same

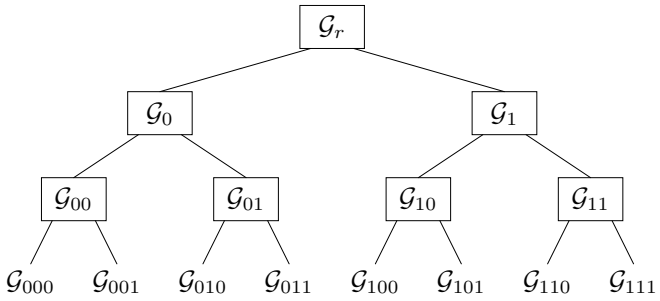


Figure 6.1: A depth 3 binary tree.

depth.

The nodes of a tree will be labeled by binary strings so that the leftmost node at depth k gets labeled by k zeroes, and the rest will equal one plus the binary value of the node to their left (keeping the number of digits constant). The root node will be denoted by the letter r, which will behave as the empty string so that $x = r$ implies $x0 = 0$ and $x1 = 1$. With these conventions the left descendant of node x is $x0$, and the right descendant is $x1$. We define $|x|$ as the length of the binary string x, which also corresponds to the depth of node x.

In this chapter we shall use calligraphic fonts, such as $\mathcal{G}$, to denote an assignment of a number or expression to each node of a binary tree. Given an assignment $\mathcal{G}$, the value of node x will be $\mathcal{G}_x$. Most of our notation is summarized by Fig. (6.1). Note that, though we shall always be working with trees of fixed finite depth, we shall usually leave the depth implicit.

We define an n-Coin-Game as an assignment $\mathcal{G}$ to a depth n binary tree such that $\mathcal{G}_x \in [0, 1]$ for all x and $\mathcal{G}_x \in \{0, 1\}$ for all leaves (i.e., for all x such that $|x| = n$). To each n-Coin-Game, $\mathcal{G}$, we can associate a classical n-message public-coin coin-flipping protocol as follows: the state of the protocol at each step will be described by a node in the tree, and this information will be kept by both Alice and Bob. The game begins at the root node and proceeds downward until reaching a leaf node. If the current node x is a binary node of even depth, then Alice chooses which path to follow and announces the choice to Bob. This is done probabilistically, by announcing the outcome of a public coin with bias $\mathcal{G}_x$, so that Alice chooses the left path with probability $\mathcal{G}_x$ and the right path with probability $1 - \mathcal{G}_x$. The same mechanism occurs at odd binary nodes, except that Bob is responsible for choosing the direction and announcing it to Alice. The game ends when arriving at a leaf node x, in which case Alice wins if $\mathcal{G}_x = 0$ and Bob wins if $\mathcal{G}_x = 1$.

Note that we do not require that the coin-flip be fair when both Alice and Bob are honest. Given

an n-Coin-Game $\mathcal{G}$, we can define $\mathcal{H}$ on a tree of the same depth by the equations:

$$\mathcal{H}_x = \begin{cases} \mathcal{G}_x & \text{if } |x| = n, \\ \mathcal{G}_x \mathcal{H}_{x0} + (1 - \mathcal{G}_x)\mathcal{H}_{x1} & \text{if } |x| < n. \end{cases} \tag{6.1}$$

The value of $\mathcal{H}_x$ indicates the conditional probability that Bob would win given that the game arrived at node x, assuming both players play honestly. The value of $\mathcal{H}_r$ is Bob's probability of winning for an honest game, which is clearly bounded between 0 and 1.

For each n-Coin-Game $\mathcal{G}$, we also define $\mathcal{A}$ and $\mathcal{B}$ on a tree of the same depth by the equations:

$$\mathcal{A}_x = \begin{cases} 1 - \mathcal{G}_x & \text{for } |x| = n, \\ \mathcal{G}_x \mathcal{A}_{x0}^2 + (1 - \mathcal{G}_x)\mathcal{A}_{x1}^2 & |x| \text{ even}, |x| < n, \\ \mathcal{G}_x \sqrt{\mathcal{A}_{x0}} + (1 - \mathcal{G}_x)\sqrt{\mathcal{A}_{x1}} & |x| \text{ odd}, |x| < n, \end{cases}$$

$$\mathcal{B}_x = \begin{cases} \mathcal{G}_x & \text{for } |x| = n, \\ \mathcal{G}_x \sqrt{\mathcal{B}_{x0}} + (1 - \mathcal{G}_x)\sqrt{\mathcal{B}_{x1}} & |x| \text{ even}, |x| < n, \\ \mathcal{G}_x \mathcal{B}_{x0}^2 + (1 - \mathcal{G}_x)\mathcal{B}_{x1}^2 & |x| \text{ odd}, |x| < n. \end{cases} \tag{6.2}$$

The importance of these quantities is given by the following theorem:

Theorem 12. *For each n-Coin-Game, $\mathcal{G}$, there exists an $(n+1)$-message quantum weak coin-flipping protocol such that*

$$P_A P_A^* = \mathcal{A}_r, \tag{6.3}$$

$$P_B P_B^* = \mathcal{B}_r^2, \tag{6.4}$$

and the honest probabilities of winning are

$$P_A = (1 - P_B) = (1 - \mathcal{H}_r), \tag{6.5}$$

where $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{H}$ are defined in terms of $\mathcal{G}$ by Eqs. (6.1,6.2).

6.2 The protocol

The purpose of this section is to describe the $(n+1)$-message quantum weak coin-flipping protocol associated to each n-Coin-Game. For each protocol we shall also derive matching upper and lower bounds on the amount that each party can cheat and thereby prove Theorem 12.

All the general ideas needed in this section have appeared in the previous chapter, though in a

somewhat different notation. The new elements needed here are

1. Chap. 5 was restricted to n-Game-Trees where all the binary nodes at the same depth had the same value (i.e, $\mathcal{G}_x = \mathcal{G}_{x'}$ if $|x| = |x'| < n$). These variables were given the name a_i so that $\mathcal{G}_x = a_{|x|+1}$. In this section we lift the restriction and consider general n-Game-Trees.

2. An upper bound on P_A^* and P_B^* was derived in Chap. 5 but was not proven optimal. In this section we shall derive a matching lower bound.

Because most of the key ideas here have been discussed in the previous chapter, we shall simply prove the necessary facts in this section without providing the intuition or motivation behind the constructions.

We begin by fixing an n-Coin-Game $\mathcal{G}$, which will be used throughout this section. We also fix $\mathcal{H}$, $\mathcal{A}$ and $\mathcal{B}$ as given by Eqs. (6.1,6.2). Because optimal protocols with $\mathcal{H}_r = 0$ and $\mathcal{H}_r = 1$ are easy to construct even classically, for what follows we shall assume that $0 < \mathcal{H}_r < 1$.

To describe the quantum protocol associated with $\mathcal{G}$ we employ the standard quantum communication model involving the Hilbert space decomposition $H_A \otimes H_M \otimes H_B$, where H_A is Alice's private space, H_B is Bob's private space, and H_M is the space used for passing messages. We further subdivide these spaces as follows:

$$H_A = H_a \otimes H_{a'} \otimes H_{ac}, \tag{6.6}$$

$$H_B = H_b \otimes H_{b'} \otimes H_{bc}, \tag{6.7}$$

$$H_M = H_m \otimes H_{mn}. \tag{6.8}$$

The spaces H_a and H_b each consists of n qubits and will be used to store a binary string x corresponding to a node in $\mathcal{G}$. The individual qubits comprising each space will be referred to as a_1 through a_n and b_1 though b_n, respectively. The one-qubit space H_m will be the primary means of communication between Alice and Bob, and will be referred to as qubit m.

The rest of the spaces will only be used in the last pair of messages. The spaces $H_{a'}$, $H_{b'}$ and H_{mn} each involve n qubits whereas H_{ac} and H_{bc} each contain one qubit.

Before describing the protocol we need to define a set of unitaries on $H_A \otimes H_M$. We begin with the controlled rotations $R_{A,k}$ defined for $k = 1, \ldots, n$ by

$$R_{A,k} = \sum_{\substack{x \\ |x|=k-1}} |x\rangle\langle x|_{a_1,\ldots,a_{k-1}} \otimes U(\mathcal{G}_x)_{a_k,m}, \tag{6.9}$$

where

$$U(z) = \begin{pmatrix} \sqrt{z} & 0 & 0 & -\sqrt{1-z} \\ 0 & \sqrt{z} & -\sqrt{1-z} & 0 \\ 0 & \sqrt{1-z} & \sqrt{z} & 0 \\ \sqrt{1-z} & 0 & 0 & \sqrt{z} \end{pmatrix}. \tag{6.10}$$

The subscripts on the operators and matrices indicate what qubits they act on, and $R_{A,k}$ acts trivially on all qubits of $H_A \otimes H_M$ not explicitly mentioned. For the case $k = 1$ the operator is not a controlled rotation but rather a regular rotation using parameter $\mathcal{G}_r$.

We shall also need the controlled rotation

$$R_{A,E} = \sum_{\substack{x \\ |x|=n}} |x\rangle\langle x|_{a_1,\ldots,a_n} \otimes \begin{pmatrix} 1-\mathcal{G}_x & -\mathcal{G}_x \\ \mathcal{G}_x & 1-\mathcal{G}_x \end{pmatrix}_{ac}, \tag{6.11}$$

which is unitary because $\mathcal{G}_x \in \{0, 1\}$ for $|x| = n$. The gate is simply a controlled-X applied to the qubit in space H_{ac}, where the control depends on a function of the qubits in H_a. Note that $R_{A,E}$ can also be defined as an operator acting purely on H_A rather than $H_A \otimes H_M$.

Finally, define $S_{A,k}$ for $k = 1, \ldots, n$ to swap qubit a_k with qubit m:

$$S_{A,k} = \mathrm{SWAP}(a_k, m). \tag{6.12}$$

We also need $T_{A,0}$, which swaps H_a with H_{mn} conditioned on qubit ac being zero, and $T_{A,1}$, which swaps the space $H_{a'}$ with the space H_{nm}, conditioned on qubit ac being one:

$$T_{A,0} = |0\rangle\langle 0|_{ac} \otimes \mathrm{SWAP}(H_a, H_{mn}) + |1\rangle\langle 1|_{ac} \otimes I, \tag{6.13}$$

$$T_{A,1} = |1\rangle\langle 1|_{ac} \otimes \mathrm{SWAP}(H_{mn}, H_{a'}) + |0\rangle\langle 0|_{ac} \otimes I. \tag{6.14}$$

The first one is used to send the qubits in H_a when Alice wins, whereas the second one is used to receive Bob's qubits and put them in $H_{a'}$ when Alice loses.

All the above operators act on Alice's Hilbert space. We can similarly define the operators $R_{B,k}$, $R_{B,E}$, $S_{B,k}$ acting in the same way on Bob's qubits. The operator $T_{B,0}$ however has to be defined to swap $H_{b'}$ with H_{mn} conditioned on qubit bc being zero, whereas $T_{B,1}$ swaps H_b with H_{mn} conditioned on qubit bc being one.

To characterize the final measurements it is useful to define the probability tree $\mathcal{P}$ by

$$\mathcal{P}_x = \begin{cases} 1 & \text{if } x = r, \\ \mathcal{G}_y \mathcal{P}_y & \text{if } x = y0, \\ (1 - \mathcal{G}_y)\mathcal{P}_y & \text{if } x = y1. \end{cases} \tag{6.15}$$

That is, $\mathcal{P}_x$ is the probability of reaching node x when the classical coin-flipping game associated with $\mathcal{G}$ is played honestly. We can now define the two normalized states

$$\begin{aligned} |\psi_{A,1}\rangle &= \frac{1}{\sqrt{\mathcal{H}_r}} \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=1}} \sqrt{\mathcal{P}_x} |x\rangle_{H_a} \otimes |x\rangle_{H_{a'}} \otimes |1\rangle_{H_{ac}}, \\ |\psi_{B,0}\rangle &= \frac{1}{\sqrt{1-\mathcal{H}_r}} \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=0}} \sqrt{\mathcal{P}_x} |x\rangle_{H_b} \otimes |x\rangle_{H_{b'}} \otimes |0\rangle_{H_{bc}}. \end{aligned} \tag{6.16}$$

The normalization is correct because $\mathcal{H}_r$ is the probability of arriving at a leaf x such that $\mathcal{G}_x = 1$, whereas $1 - \mathcal{H}_r$ is the probability of arriving at a leaf with $\mathcal{G}_x = 0$. We are now ready to describe the main protocol.

Protocol 1. *Given an n-Coin-Game, $\mathcal{G}$, and the associated operators described above, define a quantum weak coin-flipping protocol by the following steps:*

1. *Setup: Alice starts with $H_A \otimes H_M$ and Bob with H_B. They each initialize their space to the state $|0\rangle$.*

2. *First n messages. For $k = 1$ to n:*

   - *If k is odd, Alice applies $R_{A,k}$ and sends H_M to Bob who applies $S_{B,k}$.*
   - *If k is even, Bob applies $R_{B,k}$ and sends H_M to Alice who applies $S_{A,k}$.*

3. *Alice applies $R_{A,E}$ to H_A and Bob applies $R_{B,E}$ to H_B. No messages are needed for this step.*

4. *If Bob has H_M he sends it to Alice.*

5. *Alice applies $T_{A,0}$ and sends H_M to Bob who applies $T_{B,0}$.*

6. *Bob applies $T_{B,1}$ and sends H_M to Alice who applies $T_{A,1}$.*

7. *Alice measures using the two outcome POVM $\{I - |\psi_{A,1}\rangle\langle\psi_{A,1}|, |\psi_{A,1}\rangle\langle\psi_{A,1}|\}$. Bob measures the two outcome POVM $\{|\psi_{B,0}\rangle\langle\psi_{B,0}|, I - |\psi_{B,0}\rangle\langle\psi_{B,0}|\}$. They each output zero for the first outcome and one for the second.*

The basic intuition behind the protocol is that the first three steps above is a quantum implementation of the classical public-coin coin-flipping protocol associated with $\mathcal{G}$ described in Sec. 6.1. After k messages the first k bits of H_A contain a length k string indicating the depth k node at which we are currently located. The quantum amplitude associated with each such state is $\sqrt{\mathcal{P}_x}$. Step 3 is a unitary realization of the measurement that looks at the n bit string x corresponding to a leaf and stores the classical coin outcome in the qubit associated with H_{ac} for Alice and H_{bc} for Bob.

The rest of the steps involve cheat detection. Effectively, the winner declares victory immediately and then sends as much of their state as possible to the other party. The losing party then checks that the state is correct before accepting defeat.

Note that, as written, the above protocol takes either $n+2$ or $n+3$ messages. However, it is easy to see that the protocol can be run with only $n+1$ messages. For starters, only the space H_m needs to be sent back and forth in step 2, whereas only H_{mn} is used in steps 5 and 6. If we allow such a splitting, Alice starts with H_{mn} and step 4 is never needed. This reduces the protocol to $n+2$ messages always. But if n is odd then Alice ends up sending two messages in a row. The two messages can be combined into a single longer message and therefore the protocol only requires $n+1$ messages. We will also argue below that steps 5 and 6 can be interchanged, in which case when n is even Bob sends two messages in a row, and their merger leads again to a protocol with only $n+1$ messages.

We turn to the task of describing the evolution of the game when both players are honest. The action of $R_{A,k}$ entangles qubit a_k with qubit m, whereas $S_{B,k}$ swaps qubit m with b_k. Their combined effect is the transformation

$$\begin{aligned}\sqrt{\mathcal{P}_x}|x\rangle_{a_1,\ldots,a_{k-1}}\otimes|0\rangle_{a_k}\otimes|0\rangle_{b_k} \quad\longrightarrow\quad & \sqrt{\mathcal{P}_{x0}}|x\rangle_{a_1,\ldots,a_{k-1}}\otimes|0\rangle_{a_k}\otimes|0\rangle_{b_k} \\ & +\sqrt{\mathcal{P}_{x1}}|x\rangle_{a_1,\ldots,a_{k-1}}\otimes|1\rangle_{a_k}\otimes|1\rangle_{b_k}. \qquad (6.17)\end{aligned}$$

The same effect occurs on even rounds when Alice's and Bob's actions are reversed. Therefore, the state after the first k passes through step 2 is given by

$$|\psi_k\rangle=\sum_{\substack{x\\|x|=k}}\sqrt{\mathcal{P}_x}|x0\cdots0\rangle_{H_a}\otimes|0\rangle_{H_{a'}\otimes H_{ac}}\otimes|x0\cdots0\rangle_{H_b}\otimes|0\rangle_{H_{b'}\otimes H_{bc}}\otimes|0\rangle_{H_M}, \qquad (6.18)$$

where there are $n-k$ zeroes following each x.

Step 3 simply has the effect of setting up the fair coin outcome in H_{ac} and H_{bc}:

$$|\psi_E\rangle=\sum_{\substack{x\\|x|=n}}\sqrt{\mathcal{P}_x}|x\rangle_{H_a}\otimes|0\rangle_{H_{a'}}\otimes|\mathcal{G}_x\rangle_{H_{ac}}\otimes|x\rangle_{H_b}\otimes|0\rangle_{H_{b'}}\otimes|\mathcal{G}_x\rangle_{H_{bc}}\otimes|0\rangle_{H_M}. \qquad (6.19)$$

Finally, when both players are honest, step 5 has the effect of moving H_a to $H_{b'}$ conditioned on qubits ac and bc both being one. Step 6 has the effect of swapping H_b to $H_{a'}$ conditioned on ac and bc being both zero. The final state of the protocol is therefore:

$$\begin{aligned}
|\psi_F\rangle &= \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=1}} \sqrt{\mathcal{P}_x}|x\rangle_{H_a} \otimes |x\rangle_{H_{a'}} \otimes |1\rangle_{H_{ac}} \otimes |0\rangle_{H_b} \otimes |0\rangle_{H_{b'}} \otimes |1\rangle_{H_{bc}} \otimes |0\rangle_{H_M} \\
&\quad + \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=0}} \sqrt{\mathcal{P}_x}|0\rangle_{H_a} \otimes |0\rangle_{H_{a'}} \otimes |0\rangle_{H_{ac}} \otimes |x\rangle_{H_b} \otimes |x\rangle_{H_{b'}} \otimes |0\rangle_{H_{bc}} \otimes |0\rangle_{H_M} \\
&= \sqrt{\mathcal{H}_r}|\psi_{A,1}\rangle \otimes |0\rangle_{H_b \otimes H_{b'}} \otimes |1\rangle_{H_{bc}} \otimes |0\rangle_{H_M} \\
&\quad + \sqrt{1-\mathcal{H}_r}|0\rangle_{H_a \otimes H_{a'}} \otimes |0\rangle_{H_{ac}} \otimes |\psi_{B,0}\rangle \otimes |0\rangle_{H_M}.
\end{aligned} \tag{6.20}$$

Because $|\psi_{A,1}\rangle$ is orthogonal to any state with the value zero in register H_{ac} and $|\psi_{B,1}\rangle$ is orthogonal to any state with the value one in register H_{bc}, there are only two possible outcomes for the final measurements:

- Alice obtains $I - |\psi_{A,1}\rangle\langle\psi_{A,1}|$ and Bob obtains $|\psi_{B,0}\rangle\langle\psi_{B,0}|$ in which case they both output zero, that is, Alice wins. This happens with probability $1 - \mathcal{H}_r$.

- Alice obtains $|\psi_{A,1}\rangle\langle\psi_{A,1}|$ and Bob obtains $I - |\psi_{B,0}\rangle\langle\psi_{B,0}|$ in which case they both output one, that is, Bob wins. This happens with probability $\mathcal{H}_r$.

We have therefore proven the following lemma:

Lemma 13. *When playing Protocol 1 honestly, Alice's and Bob's outputs are perfectly correlated and satisfy*

$$P_A = 1 - \mathcal{H}_r, \qquad P_B = \mathcal{H}_r. \tag{6.21}$$

6.2.1 Reformulation as an SDP

We now turn to the analysis of the advantage that a cheating player can attain. Specifically, we shall focus on the case of honest Alice and cheating Bob. The case where Alice is cheating is fairly similar and will be derived at the end of the section from the case of cheating Bob.

When Bob is cheating we don't know exactly what operations (unitaries, measurements or superoperators) he may be applying to his qubits. In fact, we don't even know how many qubits he may have in his laboratory. We shall therefore focus only on the evolution of the qubits under Alice's control. This approach, first advocated by Kitaev [Kit03], will transform the maximization over Bob's cheating strategies into a semidefinite program (SDP).

Let ρ_0 be the initial state of all qubits under Alice's control, that is, it is a density operator on $H_A \otimes H_M$. Let $\rho_1, \ldots, \rho_n$ be the state of the qubits under Alice's control after each of the n passes through step 2. Note that ρ_k is a density operator for H_A when k is odd and for $H_A \otimes H_M$ when k is even. Finally let ρ_E be the state of $H_A \otimes H_M$ at the end of step 4 and let ρ_F be the state of $H_A \otimes H_M$ at the end of step 6.

Because Alice initializes her own qubits as prescribed by the protocol without interference from Bob, their initial state is given by

$$\rho_0 = |0\rangle\langle 0|_{H_A \otimes H_M}. \tag{6.22}$$

For odd k, Alice first applies the unitary $R_{A,k}$ and then sends H_M to Bob, leaving the state

$$\rho_k = \mathrm{Tr}_M \left[R_{A,k}\, \rho_{k-1} R_{A,k}^{-1} \right] \qquad (\text{for } k \text{ odd}). \tag{6.23}$$

For even k, we can't fully characterize ρ_k in terms of ρ_{k-1} but we know that given ρ_k, if we undo the swap $S_{A,k}$ and then send back H_M we must end up with ρ_{k-1}, therefore

$$\mathrm{Tr}_M \left[S_{A,k}^{-1}\, \rho_k S_{A,k} \right] = \rho_{k-1} \qquad (\text{for } k \text{ even}). \tag{6.24}$$

Step 3 only involved the use of $R_{A,E}$, a unitary on H_A. Step 4, the recovery of H_M, is only needed when n is odd. Therefore,

$$\rho_E = R_{A,E}\, \rho_n R_{A,E}^{-1} \qquad \text{for } n \text{ even}, \tag{6.25}$$

$$\mathrm{Tr}_M\, \rho_E = R_{A,E}\, \rho_n R_{A,E}^{-1} \qquad \text{for } n \text{ odd}. \tag{6.26}$$

Finally, the state of the qubits on H_A after applying $T_{A,0}$ to ρ_E must equal the state ρ_F if we undo $T_{A,1}$ (because as usual, Bob has no effect on Alice's qubits):

$$\mathrm{Tr}_M \left[T_{A,1}^{-1}\, \rho_F T_{A,1} \right] = \mathrm{Tr}_M \left[T_{A,0}\, \rho_E T_{A,0}^{-1} \right]. \tag{6.27}$$

The probability that Bob wins is given by the final measurement

$$\mathrm{Tr}\left[|\psi_{A,1}\rangle\langle\psi_{A,1}|\, \rho_F \right], \tag{6.28}$$

where it is understood that $|\psi_{A,1}\rangle\langle\psi_{A,1}|$ can be extended to an operator on $H_A \otimes H_M$ by tensoring with the identity I_M.

The preceding arguments show that no matter what cheating strategy Bob employs, the sequence of states for Alice's qubits must satisfy the above equations, and therefore P_B^* is upper bounded by

the maximum of Eq. (6.28) over all assignments to the variables $\rho_0, \ldots, \rho_n$, ρ_E, ρ_F consistent with the above equations. It is also not hard to see that Bob can achieve any set of density matrices consistent with the above equations by maintaining the purification of Alice's state. As this reduction from maximization over cheating strategies to SDP has already appeared both in the literature [Kit03, ABDR04] and in the previous chapter, we won't belabor the point and simply state the lemma we have proven:

Lemma 14. *The maximum probability with which Bob can win by cheating in Protocol 1 is given by the solution of the SDP:*

$$P_B^* = \max \mathrm{Tr}\left[|\psi_{A,1}\rangle\langle\psi_{A,1}|\,\rho_F\right], \tag{6.29}$$

over the positive semidefinite variables $\rho_0, \ldots, \rho_n$, ρ_E, ρ_F subject to the constraints of Eqs. (6.22–6.27).

The security of the above result depends solely on the laws of quantum mechanics and the assumption that Bob cannot directly influence the qubits in Alice's laboratory. We note that we are assuming, as is usual in coin-flipping protocols, that Alice can measure the size of the Hilbert space H_M (i.e., the number of qubits sent by Bob in each message) and that if at any point she receives more or less than the required number of qubits she aborts the protocol and declares herself the winner. The optimal strategy for Bob involves sending the right number of qubits in each message and therefore is described by the above formalism.

It will be important below to know that we can exchange steps 5 and 6. This would work as follows: given ρ_E we send H_M to Bob, who is supposed to apply $T_{B,1}$ to his qubits. Upon return, Alice applies $T_{A,1}$ followed by $T_{A,0}$ ending up with state ρ'_F satisfying

$$\mathrm{Tr}_M\left[T_{A,1}^{-1}T_{A,0}^{-1}\,\rho'_F T_{A,0}T_{A,1}\right] = \mathrm{Tr}_M\left[\rho_E\right]. \tag{6.30}$$

The final measurement can be done immediately before sending H_M to Bob because it only has support on H_A. The probability of Bob winning is

$$\mathrm{Tr}\left[|\psi_{A,1}\rangle\langle\psi_{A,1}|\,\rho'_F\right]. \tag{6.31}$$

However, $|\psi_{A,1}\rangle$ only has support on the space where qubit ac is one, and in this subspace $T_{A,0}$ acts trivially (and $T_{A,0}$ and $T_{A,1}$ commute). Applying projectors to both sides of the Eq. (6.27) and Eq. (6.30) we see that both SDPs are equivalent, and therefore steps 5 and 6 are interchangeable, at least from the perspective of honest Alice.

6.2.2 Lower bounds

To find a lower bound on P_B^* we shall describe a specific assignment of the variables ρ that satisfies the above equations and from it calculate $\mathrm{Tr}\left[|\psi_{A,1}\rangle\langle\psi_{A,1}|\,\rho_F\right]$. Because P_B^* is a maximum over such assignments, this will serve as a lower bound.

Let

$$\rho_k = \begin{cases} \sigma_k \otimes |0\rangle\langle 0|_{a_{k+1},\ldots,a_n} \otimes |0\rangle\langle 0|_{H_{a'} \otimes H_{ac}} & k \text{ odd} \\ \sigma_k \otimes |0\rangle\langle 0|_{a_{k+1},\ldots,a_n} \otimes |0\rangle\langle 0|_{H_{a'} \otimes H_{ac} \otimes H_M} & k \text{ even} \end{cases} \tag{6.32}$$

where σ_k is a density operator for qubits a_1 through a_k. The operators ρ_1 through ρ_n satisfy Eqs. (6.23,6.24) provided that

$$\sigma_k = \mathrm{Tr}_m \left[R_{A,k} \left(\sigma_{k-1} \otimes |0\rangle\langle 0|_{a_k,m} \right) R_{A,k}^{-1} \right] \quad (\text{for } k \text{ odd}) \tag{6.33}$$

where $\sigma_0 = 1$ is the unit, and

$$\mathrm{Tr}_{a_k}\left[\sigma_k\right] = \sigma_{k-1} \qquad (\text{for } k \text{ even}). \tag{6.34}$$

The σ operators above will be defined using a tree variable $\mathcal{W}$ given by the equation

$$\mathcal{W}_x = \begin{cases} 1 & x = r \\ \mathcal{G}_y W_y & x = y0 \text{ and } |x| \text{ odd} \\ (1-\mathcal{G}_y) W_y & x = y1 \text{ and } |x| \text{ odd} \\ \mathcal{G}_y \mathcal{B}_x^2 W_y / \mathcal{B}_y & x = y0 \text{ and } |x| \text{ even} \\ (1-\mathcal{G}_y) \mathcal{B}_x^2 W_y / \mathcal{B}_y & x = y1 \text{ and } |x| \text{ even,} \end{cases} \tag{6.35}$$

which is based on the weight matrix W of the previous chapter. Note that, though it is possible for $\mathcal{B}_y$ to be zero, this can only occur if both $\mathcal{B}_{y0}$ and $\mathcal{B}_{y1}$ are zero as well, in this case we define $\mathcal{W}_{y0} = \mathcal{W}_{y1} = 0$, which resolves the potential division by zero.

Because $\mathcal{B}$ is computed bottom-up, whereas $\mathcal{W}$ is computed top-down, every node of $\mathcal{W}$ depends on the complete n-Coin-Game assignment $\mathcal{G}$. The appearance at every node of such global information about the protocol is crucial for optimal solutions of these SDPs and will also occur with the tree variable $\mathcal{Z}$ defined below in the section on upper bounds.

Define the σ operators as diagonal matrices with entries given by

$$\langle x|\sigma_k|x\rangle = \mathcal{W}_x \qquad \text{for } |x| = k. \tag{6.36}$$

The requirements of Eq. (6.33) are satisfied if

$$\mathcal{W}_{y0} = \mathcal{G}_y \mathcal{W}_y \quad \text{and} \quad \mathcal{W}_{y1} = (1 - \mathcal{G}_y)\mathcal{W}_y \quad (\text{for } |y| \text{ even}), \tag{6.37}$$

whereas Eq. (6.34) only imposes the weaker requirement

$$\mathcal{W}_y = \mathcal{W}_{y0} + \mathcal{W}_{y1} \quad (\text{for } |y| \text{ odd}), \tag{6.38}$$

both of which are clearly satisfied by $\mathcal{W}$. We have therefore outlined a valid cheating strategy for Bob through step 2.

The next two steps will follow the protocol exactly, in which case the operator ρ_E follows from ρ_n by adjusting the space H_{ac}:

$$\rho_E = \sum_x \mathcal{W}_x |x\rangle\langle x|_{H_a} \otimes |0\rangle\langle 0|_{H_{a'}} \otimes |\mathcal{G}_x\rangle\langle \mathcal{G}_x|_{H_{ac}} \otimes |0\rangle\langle 0|_{H_M}. \tag{6.39}$$

Finally, in the last steps, conditioned on qubit ac being zero, Alice sends her state to Bob. Conditioned on qubit ac begin one, Bob returns the purification of the remaining qubits, so the final state is:

$$\begin{aligned} \rho_F &= |\phi_1\rangle\langle\phi_1|_{H_a \otimes H_{a'}} \otimes |1\rangle\langle 1|_{H_{ac}} \otimes |0\rangle\langle 0|_{H_M} \\ &\quad + C_0 |0\rangle\langle 0|_{H_a \otimes H_{a'}} \otimes |0\rangle\langle 0|_{H_{ac}} \otimes |0\rangle\langle 0|_{H_M}, \end{aligned} \tag{6.40}$$

where C_0 is an unimportant constant (equal to the sum of $\mathcal{W}_x$ for all x such that $\mathcal{G}_x = 0$), and $|\phi_1\rangle$ is the unnormalized state given by

$$|\phi_1\rangle = \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=1}} \sqrt{\mathcal{W}_x} |x\rangle_{H_a} \otimes |x\rangle_{H_{a'}}. \tag{6.41}$$

Bob's probability of winning is given by

$$\begin{aligned} p &= \left| \left(\langle \phi_1 |_{H_a \otimes H_{a'}} \otimes \langle 1 |_{H_{ac}} \right) |\psi_{A,1}\rangle \right|^2 \\ &= \left| \sum_{\substack{x \\ |x|=n}} \mathcal{G}_x \sqrt{\mathcal{W}_x \mathcal{P}_x} \right|^2 / \mathcal{H}_r, \end{aligned} \tag{6.42}$$

where the factor of $\mathcal{G}_x$ ensures that the sum is only taken over strings x satisfying $\mathcal{G}_x = 1$.

While the expression for computing p seems rather daunting, we shall show in a moment that when properly written, it is a conserved quantity that has the same value at every depth in the tree.

We begin with the following two observations: for $|y|$ even

$$\sqrt{\mathcal{B}_{y0}}\sqrt{\mathcal{W}_{y0}\mathcal{P}_{y0}}+\sqrt{\mathcal{B}_{y1}}\sqrt{\mathcal{W}_{y1}\mathcal{P}_{y1}}=\left(\mathcal{G}_y\sqrt{\mathcal{B}_{y0}}+(1-\mathcal{G}_y)\sqrt{\mathcal{B}_{y1}}\right)\sqrt{\mathcal{W}_y\mathcal{P}_y}=\mathcal{B}_y\sqrt{\mathcal{W}_y\mathcal{P}_y} \quad (6.43)$$

whereas for $|y|$ odd we have

$$\mathcal{B}_{y0}\sqrt{\mathcal{W}_{y0}\mathcal{P}_{y0}}+\mathcal{B}_{y1}\sqrt{\mathcal{W}_{y1}\mathcal{P}_{y1}}=\left(\mathcal{G}_y\mathcal{B}_{y0}^2+(1-\mathcal{G}_y)\mathcal{B}_{y1}^2\right)\sqrt{\mathcal{W}_y\mathcal{P}_y/\mathcal{B}_y}=\sqrt{\mathcal{B}_y}\sqrt{\mathcal{W}_y\mathcal{P}_y}. \quad (6.44)$$

For the special case when $\mathcal{B}_y=0$ the equation is also valid as it reads $0+0=0$. By induction, we can obtain the following result

$$\mathcal{B}_r\sqrt{\mathcal{W}_r\mathcal{P}_r}=\begin{cases}\sum_{x;|x|=k}\mathcal{B}_x\sqrt{\mathcal{W}_x\mathcal{P}_x} & \text{for any even } k,\\ \sum_{x;|x|=k}\sqrt{\mathcal{B}_x}\sqrt{\mathcal{W}_x\mathcal{P}_x} & \text{for any odd } k,\end{cases} \quad (6.45)$$

where as usual $0\leq k\leq n$. In particular, because for $|x|=n$ we have $\mathcal{G}_x=\mathcal{B}_x=\sqrt{\mathcal{B}_x}\in\{0,1\}$ we have shown that $p=\left|\mathcal{B}_r\sqrt{\mathcal{W}_r\mathcal{P}_r}\right|^2/\mathcal{H}_r$, which is the probability with which Bob can win the coin-flip by cheating using the strategy outlined above. Since $\mathcal{W}_r=\mathcal{P}_r=1$ we have proven the desired lower bound:

Lemma 15. *For Protocol 1:*

$$P_B^*\geq\frac{\mathcal{B}_r^2}{\mathcal{H}_r}. \quad (6.46)$$

6.2.3 Upper bounds

We shall prove an upper bound by exhibiting a solution to the dual SDP. We use the derivation of the dual in [ABDR04], though a direct derivation would be fairly simple as well.

Our protocol can be rewritten in the notation of [ABDR04]. Let $m=\lfloor(n+1)/2\rfloor$ and define $U_{A,1}=R_{A,1}$, $U_{A,j}=R_{A,2j-1}S_{A,2j-2}$ for $j=2,\ldots,m$, $U_{A,m+1}=T_{A,0}R_{A,E}S_{A,n}$ (or if n is odd just $U_{A,m+1}=T_{A,0}R_{A,E}$) and $U_{A,m+2}=T_{A,1}$. The final measurement is $\Pi_{A,1}=|\psi_{A,1}\rangle\langle\psi_{A,1}|$. In this notation, we are looking for the maximum of $\mathrm{Tr}[\Pi_{A,1}\,\rho_{A,m+2}]$ over assignments of the positive semidefinite variables $\rho_{A,0},\ldots,\rho_{A,m+2}$ satisfying:

$$\mathrm{Tr}_M\left[\rho_{A,j}\right]=\mathrm{Tr}_M\left[U_{A,j}\,\rho_{A,j-1}\,U_{A,j}^{-1}\right] \quad (6.47)$$

for $j=1,\ldots,m+2$ and $\mathrm{Tr}_M[\rho_{A,0}]=|0\rangle\langle 0|_{H_A}$. The initial condition for $\rho_{A,0}$ (rather than the usual $\rho_{A,0}=|0\rangle\langle 0|_{H_A\otimes H_M}$) simply gives Bob a little more cheating power (i.e., to initialize H_M) but this is acceptable as we are now focusing on deriving upper bounds on P_B^*, and this extra cheating power will not be helpful.

The dual SDP is given by Lemma 11 of [ABDR04] as the minimization of $\langle 0|Y_{A,0}|0\rangle$, subject to

$$Y_{A,j} \otimes I_{H_M} \geq U_{A,j+1}^{-1} \left(Y_{A,j+1} \otimes I_{H_M}\right) U_{A,j+1} \tag{6.48}$$

for $0 \leq j \leq m+1$, where $Y_0, \ldots, Y_{m+1}$ are Hermitian operators on H_A and $Y_{A,m+2} \equiv \Pi_{A,1}$. Because this is the dual SDP to the original coin-flipping SDP corresponding to Protocol 1, any assignment of the variables $Y_{A,i}$ that satisfies the constraints will produce a value of $\langle 0|Y_{A,0}|0\rangle$ that is an upper bound on P_B^*. However, rather than finding a solution to the above dual SDP, we shall study a modified, but equivalent, SDP:

Lemma 16. *Let $Z_0, \ldots, Z_{n+2}$ be a set of Hermitian matrices, defined on H_A, satisfying the following equations:*

$$\begin{aligned} Z_k \otimes I_{H_M} &\geq R_{A,k+1}^{-1} \left(Z_{k+1} \otimes I_{H_M}\right) R_{A,k+1} \qquad \textit{(k even)}, \\ Z_k \otimes I_{H_M} &\geq S_{A,k+1}^{-1} \left(Z_{k+1} \otimes I_{H_M}\right) S_{A,k+1} \qquad \textit{(k odd)}, \end{aligned} \tag{6.49}$$

where $0 \leq k < n$, and

$$Z_n \otimes I_{H_M} \geq R_{A,E}^{-1} \left(Z_{n+1} \otimes I_{H_M}\right) R_{A,E}, \tag{6.50}$$

$$Z_{n+1} \otimes I_{H_M} \geq T_{A,0}^{-1} \left(Z_{n+2} \otimes I_{H_M}\right) T_{A,0}, \tag{6.51}$$

$$Z_{n+2} \otimes I_{H_M} \geq T_{A,1}^{-1} \left(|\psi_{A,1}\rangle\langle\psi_{A,1}| \otimes I_{H_M}\right) T_{A,1}. \tag{6.52}$$

then $\beta \equiv \langle 0|Z_0|0\rangle$ is an upper bound on P_B^*.

The proof follows by noting that given a set of $Z_0, \ldots, Z_{n+2}$ satisfying the above equations, we can set $Y_0 = Z_0$, $Y_j = Z_{2j-1}$ for $j = 1, \ldots, m$ and $Y_{m+1} = Z_{n+2}$ to obtain a solution with the same minimum as the original dual SDP.

We introduce a new variable, defined on a tree of depth n, which shall be used in constructing solutions of the dual SDP:

$$\mathcal{Z}_x = \begin{cases} \mathcal{B}_r^2/\mathcal{H}_r & x = r \\ \sqrt{\mathcal{B}_x} Z_y/\mathcal{B}_y & |x| \text{ odd} \\ Z_y & |x| \text{ even} \end{cases} \tag{6.53}$$

where y is the parent node of x (i.e., either $x = y0$ or $x = y1$). Once again we resolve the division by zero by declaring $\mathcal{Z}_{y0} = \mathcal{Z}_{y1} = 0$ whenever $\mathcal{B}_y = 0$ and $|y|$ is even.

We begin the description of the solution to the dual SDP by choosing

$$Z_{n+2} = \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=1}} \mathcal{Z}_x |x\rangle\langle x|_{H_a} \otimes I_{H_{a'}} \otimes |1\rangle\langle 1|_{H_{ac}}. \tag{6.54}$$

To verify that Z_{n+2} satisfies Eq. (6.52), we note that we can move the unitary operators $T_{A,1}$ to the left hand side of the equation, where they act trivially (i.e., they exchange $I_{H_{a'}}$ with $I_{H_{mn}}$). We are left with the task of proving $Z_{n+2} \geq |\psi_{A,1}\rangle\langle\psi_{A,1}|$.

It is sufficient to show that

$$Z_{n+2} + \epsilon I_{H_A} \geq |\psi_{A,1}\rangle\langle\psi_{A,1}| \tag{6.55}$$

for every $\epsilon > 0$. Because Z_{n+2} is non-negative, the left-hand-side above is positive definite. We can rescale our space by $(Z_{n+2} + \epsilon I_{H_A})^{-1/2}$ to obtain the equivalent equation

$$I \geq (Z_{n+2} + \epsilon I_{H_A})^{-\frac{1}{2}} |\psi_{A,1}\rangle\langle\psi_{A,1}| (Z_{n+2} + \epsilon I_{H_A})^{-\frac{1}{2}}. \tag{6.56}$$

The right-hand-side of the above equation has only one non-zero eigenvalue, it is therefore sufficient to check that

$$1 \geq \langle\psi_{A,1}| (Z_{n+2} + \epsilon I_{H_A})^{-1} |\psi_{A,1}\rangle. \tag{6.57}$$

We need to study the quantity

$$\langle\psi_{A,1}| (Z_{n+2} + \epsilon I_{H_A})^{-1} |\psi_{A,1}\rangle = \sum_{\substack{x \\ |x|=n \\ \mathcal{G}_x=1}} \frac{\mathcal{P}_x}{\mathcal{H}_r(\mathcal{Z}_x + \epsilon)}, \tag{6.58}$$

which once again is related to a conserved quantity at every level of the tree. However, we first note the following properties that can be checked directly from the definitions:

- $\mathcal{P}_x > 0$ implies $\mathcal{P}_y > 0$ for every node y that has x as a descendant.
- $\mathcal{P}_x > 0$ and $\mathcal{B}_x > 0$ implies that $\mathcal{B}_y > 0$ for every node y that has x as a descendant.
- $\mathcal{P}_x > 0$ and $\mathcal{B}_x > 0$ implies $\mathcal{Z}_x > 0$.

We can now remove ϵ from the above expression, because if $\mathcal{Z}_x = 0$ then either $\mathcal{P}_x = 0$ or $\mathcal{B}_x = 0$ (which implies $\mathcal{G}_x = 0$):

$$\langle\psi_{A,1}| (Z_{n+2} + \epsilon I_{H_A})^{-1} |\psi_{A,1}\rangle \leq \sum_{\substack{x \\ |x|=n \\ \mathcal{Z}_x>1}} \frac{\mathcal{G}_x \mathcal{P}_x}{\mathcal{H}_r \mathcal{Z}_x} \tag{6.59}$$

where the factor $\mathcal{G}_x$ imposes the condition $\mathcal{G}_x = 1$, and the condition $\mathcal{Z}_x > 0$ has been moved into the sum.

If $|y|$ is odd and $\mathcal{Z}_y > 0$ we have

$$\frac{\mathcal{B}_{y0}^2\mathcal{P}_{y0}}{\mathcal{Z}_{y0}} + \frac{\mathcal{B}_{y1}^2\mathcal{P}_{y1}}{\mathcal{Z}_{y1}} = \left(\mathcal{B}_{y0}^2\mathcal{G}_y + \mathcal{B}_{y1}^2(1-\mathcal{G}_y)\right)\frac{\mathcal{P}_y}{\mathcal{Z}_y} = \frac{\mathcal{B}_y\mathcal{P}_y}{\mathcal{Z}_y}, \tag{6.60}$$

where the left-hand side is well defined because $\mathcal{Z}_{y0} = \mathcal{Z}_{y1} = \mathcal{Z}_y > 0$. If $|y|$ is even we have

$$\frac{\mathcal{B}_{y0}\mathcal{P}_{y0}}{\mathcal{Z}_{y0}} + \frac{\mathcal{B}_{y1}\mathcal{P}_{y1}}{\mathcal{Z}_{y1}} = \left(\sqrt{\mathcal{B}_{y0}}\mathcal{G}_y + \sqrt{\mathcal{B}_{y1}}(1-\mathcal{G}_y)\right)\frac{\mathcal{B}_y\mathcal{P}_y}{\mathcal{Z}_y} = \frac{\mathcal{B}_y^2\mathcal{P}_y}{\mathcal{Z}_y}. \tag{6.61}$$

Even if $\mathcal{Z}_y > 0$ it is possible for either $\mathcal{Z}_{y0}$ or $\mathcal{Z}_{y1}$ (or both) to be zero. If both are zero, then so is $\mathcal{B}_y\mathcal{P}_y$. If only one of them is zero (say $\mathcal{Z}_{y0}$) then the equation is still valid with the offending term removed (that is, $\mathcal{B}_{y1}\mathcal{P}_{y1}/\mathcal{Z}_{y1} = \mathcal{B}_y^2\mathcal{P}_y/\mathcal{Z}_y$). Using induction, we can prove

$$1 = \frac{\mathcal{B}_r^2\mathcal{P}_r}{\mathcal{H}_r\mathcal{Z}_r} = \begin{cases} \sum_{x;|x|=k;\mathcal{Z}_x>0} \frac{\mathcal{B}_x^2\mathcal{P}_x}{\mathcal{H}_r\mathcal{Z}_x} & \text{for any even k} \\ \sum_{x;|x|=k;\mathcal{Z}_x>0} \frac{\mathcal{B}_x\mathcal{P}_x}{\mathcal{H}_r\mathcal{Z}_x} & \text{for any odd } k \end{cases} \tag{6.62}$$

and in particular, because $\mathcal{G}_x = \mathcal{B}_x = \mathcal{B}_x^2$ for $|x| = n$ we have shown $\langle\psi_{A,1}|\left(Z_{n+2} + \epsilon I_{H_A}\right)^{-1}|\psi_{A,1}\rangle \leq 1$ for every $\epsilon > 0$, thus completing the proof that our choice for Z_{n+2} satisfies the requirement imposed by Eq. (6.52).

The next few requirements are easier to check. Since Z_{n+2} only has support on the space in which qubit ac is one, on which $T_{A,0}$ acts trivially, we can satisfy Eq. (6.51) by choosing

$$Z_{n+1} = \sum_{\substack{x \\ |x|=n}} \mathcal{Z}_x |x\rangle\langle x|_{H_a} \otimes I_{H_{a'}} \otimes |\mathcal{G}_x\rangle\langle\mathcal{G}_x|_{H_{ac}} \geq Z_{n+2}, \tag{6.63}$$

where the inequality follows because we have simply included the (non-negative) coefficients for the states with $\mathcal{G}_x = 0$.

The unitary $R_{A,E}$ operates only on the space H_A hence Eq. (6.50) can be satisfied by choosing

$$Z_n = R_{A,E}^{-1} Z_{n+1} R_{A,E} = \sum_{\substack{x \\ |x|=n}} \mathcal{Z}_x |x\rangle\langle x|_{H_a} \otimes I_{H_{a'}} \otimes |0\rangle\langle 0|_{H_{ac}}. \tag{6.64}$$

Finally, fix a new parameter $\epsilon' > 0$, and define

$$\begin{aligned} Z_k \quad = \quad & \sum_{\substack{x \\ |x|=k}} \left(\mathcal{Z}_x + \frac{(n-k)\epsilon'}{n}\right) |x\rangle\langle x|_{a_1,\ldots,a_k} \otimes |0\rangle\langle 0|_{H_{a_{k+1},\ldots,a_n} \otimes H_{a'} \otimes H_{ac}} \\ & + C_k \, I_{a_1,\ldots,a_k} \otimes (I - |0\rangle\langle 0|)_{H_{a_{k+1},\ldots,a_n} \otimes H_{a'} \otimes H_{ac}}, \end{aligned} \tag{6.65}$$

for $k = 0, \ldots, n-1$. The constants C_k will be defined recursively below, starting with C_{n-1}. For $k = 0$ the above should be interpreted as

$$Z_0 = (\mathcal{Z}_r + \epsilon') \left|0\right\rangle\!\left\langle 0\right|_{H_A} + C_0 \left(I - \left|0\right\rangle\!\left\langle 0\right|\right)_{H_A}. \tag{6.66}$$

In order to prove that our solution to the dual SDP is valid, all that remains is to check Eq. (6.49). The case of k odd is fairly simple because $\mathcal{Z}_y = \mathcal{Z}_{y0} = \mathcal{Z}_{y1}$ for $|y|$ odd, therefore qubit a_{k+1} of Z_{k+1} is unentangled with the rest of the qubits, and its state is the identity density matrix (i.e., $Z_{k+1} = I_{a_{k+1}} \otimes Z'$ where Z' is an operator on the rest of the qubits). As the swap operator $S_{A,k+1}$ acts trivially on $Z_{k+1} \otimes I_{H_M}$, it is sufficient to check $Z_k \geq Z_{k+1}$, which is satisfied if $C_k \geq C_{k+1}$. For the special case of $k = n-1$ (and n even) it suffices to choose $C_k \geq \max \mathcal{Z}_x$ where the maximum is taken over all strings x such that $|x| = n$.

What remains to be proven is Eq. (6.49) for the case of even k. Fix some even value of k, and let $\alpha = Z_k \otimes I_{H_M}$ and $\beta = R_{A,k+1}^{-1} \left(Z_{k+1} \otimes I_{H_M}\right) R_{A,k+1}$. There are just the left- and right-hand sides of the equation we are trying to prove: $\alpha \geq \beta$. Define the projector

$$\Pi = I_{a_1,\ldots,a_k} \otimes \left|0\right\rangle\!\left\langle 0\right|_{H_{a_{k+1},\ldots,a_n} \otimes H'_a \otimes H_{ac}} \otimes I_{H_M}. \tag{6.67}$$

We shall prove in a moment $\Pi(\alpha - \beta)\Pi = \frac{\epsilon'}{n}\Pi$. It is also easy to see that $\Pi\alpha(I - \Pi) = (I - \Pi)\alpha\Pi = 0$ and $(I - \Pi)\alpha(I - \Pi) = C_k(I - \Pi)$. Under these conditions, it is always possible to choose a large enough C_k so that $\alpha \geq \beta$, which defines C_k in terms of C_{k+1} (except for C_{n-1}, which can be defined directly from Z_n). For a proof, see for instance the proof of Lemma 9 in the previous chapter.

To prove $\Pi(\alpha - \beta)\Pi = \frac{\epsilon'}{n}\Pi$ we need to study the effect of the unitary $R_{A,k+1}$ on Z_{n+1}. The expression has the form of a sum of $\left|x\right\rangle\!\left\langle x\right|_{a_1,\ldots,a_k}$ tensored with

$$U(\mathcal{G}_x)^{-1} \left[\left(\mathcal{Z}_{x0} \left|0\right\rangle\!\left\langle 0\right|_{a_{k+1}} + \mathcal{Z}_{x1} \left|1\right\rangle\!\left\langle 1\right|_{a_{k+1}} \right) \otimes I_m \right] U(\mathcal{G}_x) \tag{6.68}$$

for $|x| = k$, where $U(z)$ is defined by Eq. (6.10). The component of the above that survives the projection Π has the form

$$\begin{aligned} \left(\mathcal{G}_x \mathcal{Z}_{x0} + (1 - \mathcal{G}_x)\mathcal{Z}_{x1}\right) \left|0\right\rangle\!\left\langle 0\right|_{a_{k+1}} \otimes I_m &= \left(\mathcal{G}_x \sqrt{\mathcal{B}_{x0}} + (1 - \mathcal{G}_x)\sqrt{\mathcal{B}_{x1}} \right) \frac{\mathcal{Z}_x}{\mathcal{B}_x} \left|0\right\rangle\!\left\langle 0\right|_{a_{k+1}} \otimes I_m \\ &= \mathcal{Z}_x \left|0\right\rangle\!\left\langle 0\right|_{a_{k+1}} \otimes I_m. \end{aligned} \tag{6.69}$$

It is now straightforward to check that $\Pi\alpha\Pi = \Pi\beta\Pi + \frac{\epsilon'}{n}\Pi$, completing the proof that our choice of Z_k satisfies Eq. (6.49).

Note that, while the original protocol only depends on the first column of the matrix $U(z)$, the above calculation involved the entire matrix. The reason for this is that when transforming from the

SDP involving the Y variables to the SDP involving the Z variables we gave Bob a small amount of extra cheating power to set the qubits in H_M between application of $S_{A,k}$ and $R_{A,k+1}$, in which case the full matrix $U(z)$ becomes important. However, since the upper bound derived in this section matches the lower bound from the last section, it should be clear that such extra power is not useful.

The result thus far is the description of a set of variables $Z_0, \ldots, Z_{n+2}$ satisfying the equations of the dual SDP. This gives us an upper bound $P_B^* \leq \beta = \langle 0|Z_0|0\rangle = Z_r + \epsilon'$. However, since $\epsilon' > 0$ is arbitrary, we have proven

Lemma 17. *For Protocol 1:*

$$P_B^* \leq \frac{\mathcal{B}_r^2}{\mathcal{H}_r}. \tag{6.70}$$

6.2.4 Honest Bob vs. Cheating Alice

The analysis of the case of honest Bob and cheating Alice is fairly similar to the above calculations. Fortunately, we can exploit certain symmetries in the protocol to derive expressions for P_A^* from the above expressions for P_B^*.

Given an n-Game-Tree $\mathcal{G}$, define a new $(n+1)$-Game-Tree, $\mathcal{G}'$ by the rules $\mathcal{G}'_r = 1$, $\mathcal{G}'_{0x} = \mathcal{G}'_{1x} = \mathcal{G}_x$ for $|x| < n$ and $\mathcal{G}'_{0x} = \mathcal{G}'_{1x} = 1 - \mathcal{G}_x$ for $|x| = n$. We'd like to argue that the quantum protocol associated with $\mathcal{G}'$ is equivalent to the protocol associated with $\mathcal{G}$ but with Alice's and Bob's roles exchanged.

The basic idea is that the first message of $\mathcal{G}'$, which Alice sends to Bob, is the pure state $|0\rangle$. If Bob is cheating this state reveals no extra information about Alice's state, and if Alice is cheating she has no incentive to reveal herself as a cheater by sending anything other than the state $|0\rangle$. The subsequent messages in $\mathcal{G}'$ correspond to those of $\mathcal{G}$ but with Alice and Bob reversed. The only potential problem with this argument is that the order of the cheat detection messages (steps 5 and 6) needs to be switched in order to make the protocols equivalent. However, we argued after formulating the problem as an SDP that these two steps could be exchanged without increasing or decreasing P_B^*.

Therefore, Bob's maximum probability of winning by cheating in $\mathcal{G}'$, which we call $P_B^{*\prime}$ and can be calculated using the above formulas, equals P_A^*. But $\mathcal{B}'_r = \sqrt{\mathcal{A}_r}$ and $\mathcal{H}'_r = 1 - \mathcal{H}_r$, where the primed variables are calculated from $\mathcal{G}'$. The conclusion is that

$$P_A^* = P_B^{*\prime} = \frac{\mathcal{B}_r'^2}{\mathcal{H}'_r} = \frac{\mathcal{A}_r}{1 - \mathcal{H}_r}. \tag{6.71}$$

In particular we have proven the main result of this section, which is equivalent to Theorem 12:

Theorem 18. *The quantum weak coin-flipping protocol associated to an n-Coin-Game $\mathcal{G}$ by Protocol*

1 satisfies:

$$P_A^* = \frac{\mathcal{A}_r}{1-\mathcal{H}_r}, \qquad P_B^* = \frac{\mathcal{B}_r^2}{\mathcal{H}_r}, \tag{6.72}$$

and $P_A = 1 - P_B = 1 - \mathcal{H}_r$*, where* $\mathcal{A}$*,* $\mathcal{B}$ *and* $\mathcal{H}$ *are defined in terms of* $\mathcal{G}$ *by Eqs. (6.1,6.2).*

The above result could be made more symmetric between Alice and Bob if we were to redefine $\mathcal{A}$ and $\mathcal{B}$ by

$$\mathcal{A}_x^{(new)} = \begin{cases} \sqrt{\mathcal{A}_x} & |x| \text{ even} \\ \mathcal{A}_x & |x| \text{ odd} \end{cases} \tag{6.73}$$

$$\mathcal{B}_x^{(new)} = \begin{cases} \mathcal{B}_x & |x| \text{ even} \\ \sqrt{\mathcal{B}_x} & |x| \text{ odd,} \end{cases} \tag{6.74}$$

which could be computed bottom-up by a sequence of linear and root-mean-squared averages as in the previous chapter. The new definitions would also make the conserved quantities in Eqs. (6.45, 6.62) have the same expression at even and odd depths. However, the old definitions make manifest the convexity that will be exploited in the next sections of this chapter, and therefore these definitions were selected.

6.3 Lower bounds on the bias

In this section we shall derive lower bounds for the set of P_A^* and P_B^* that can be achieved with quantum protocols based on n-Coin-Games as defined above in Theorem 18.

Definition 19. *For* $n \in \mathbb{Z}^+$*, define the set* $\Lambda_n \subset \mathbb{R}^2$ *so that* $(A, B) \in \Lambda_n$ *if and only if there exists an* n*-Coin-Game,* $\mathcal{G}$*, with* $A = \mathcal{A}_r$ *and* $B = \mathcal{B}_r$ *and* $\mathcal{A}$ *and* $\mathcal{B}$ *defined in terms of* $\mathcal{G}$ *by Eq. (6.2).*

For each $(A, B) \in \Lambda_n$ there exists an $(n+1)$-message quantum coin-flipping protocol such that $P_A P_A^* = A$ and $P_B P_B^* = B^2$. Furthermore, if $(P_A P_A^*, \sqrt{P_B P_B^*}) \notin \Lambda_n$ then there is no protocol built out of a n-Coin-Game that achieves P_A, P_A^* and P_B^*. However, it is not true that $(P_A P_A^*, \sqrt{P_B P_B^*}) \in \Lambda_n$ implies the existence of a protocol with those parameters. For example, $(0.3531, \sqrt{0.3531}) \in \Lambda_2$ because there exists a 3-message protocol with $P_A \simeq 0.515$, $P_A^* \simeq 0.686$, $P_B^* \simeq 0.728$; however, there are no 3-message protocols with $P_A = P_B = 1/2$ and $P_A^* = P_B^* \simeq 2 * 0.353 = 0.706$. The optimal symmetric 3-message protocol is the one by Spekkens and Rudolph [SR02b] with $P_A^* = P_B^* = 1/\sqrt{2} = 0.707$. Though it would be preferable to study the set of achievable triplets $(\mathcal{A}_r, \mathcal{B}_r, \mathcal{H}_r)$, the sets Λ_n are easier to analyze and in the limit $n \to \infty$ will provide us with interesting bounds.

We begin the study of the sets Λ_n by showing that they can be obtained inductively:

Lemma 20. *The set Λ_n is the convex combination of pairs of points of the form $\{(B^2, \sqrt{A}) \,|\, (A, B) \in \Lambda_{n-1}\}$.*

Proof. Given an n-Coin-Game, $\mathcal{G}$, define the variable $\gamma \equiv \mathcal{G}_r \in [0, 1]$ and the two $(n-1)$-Coin-Games $\mathcal{G}^{(0)}$ and $\mathcal{G}^{(1)}$ by

$$\mathcal{G}^{(i)}_x = \begin{cases} 1 - \mathcal{G}_{ix} & \text{for } |x| = n-1, \\ \mathcal{G}_{ix} & \text{for } |x| < n-1, \end{cases} \tag{6.75}$$

for $i = 0, 1$. There is a natural isomorphism between $\mathcal{G}$ and the triplet $\gamma, \mathcal{G}^{(0)}, \mathcal{G}^{(1)}$.

Furthermore define $\mathcal{A}^{(i)}$ and $\mathcal{B}^{(i)}$ in terms of $\mathcal{G}^{(i)}$ in the usual way. Note that $\mathcal{A}^{(i)}$ and $\mathcal{B}^{(i)}$ are not the left and right branches of $\mathcal{A}$ and $\mathcal{B}$ defined from $\mathcal{G}$ but rather $\mathcal{A}^{(i)}_x = \mathcal{B}_{ix}$ and $\mathcal{B}^{(i)}_x = \mathcal{A}_{ix}$. Therefore

$$\mathcal{A}_r = \gamma \left(\mathcal{B}^{(0)}_r\right)^2 + (1-\gamma)\left(\mathcal{B}^{(1)}_r\right)^2, \tag{6.76}$$

$$\mathcal{B}_r = \gamma\sqrt{\mathcal{A}^{(0)}_r} + (1-\gamma)\sqrt{\mathcal{A}^{(1)}_r}. \tag{6.77}$$

□

The set Λ_1 is fairly simple and corresponds to the convex combinations of the two points $(1, 0)$ and $(0, 1)$, which could be thought of as comprising Λ_0. Using Λ_1 and the above lemma we can prove two simple properties of the sets Λ_n:

1. $(0, 1) \in \Lambda_n$ and $(1, 0) \in \Lambda_n$ for all n.

2. $\Lambda_n \subset [0, 1] \times [0, 1]$ for all n.

Both properties are clearly true for Λ_1. By induction $(0, 1) \in \Lambda_{n-1}$ and $(1, 0) \in \Lambda_{n-1}$ implies that $(1^2, \sqrt{0})$ and $(0^2, \sqrt{1})$ are in Λ_n. Similarly, if $(A, B) \in \Lambda_{n-1}$ implies $A \in [0, 1]$ and $B \in [0, 1]$, then $(B^2, \sqrt{A}) \in [0, 1] \times [0, 1]$ and so are convex combinations of such points.

The first non-trivial set is Λ_2, which is the convex combination of the points on the curve $(t^2, \sqrt{1-t})$ for $t \in [0, 1]$. The curve is plotted in Fig. 6.2. The dotted line marks the lower boundary of its convex hull, which can be achieved using convex combinations of two points (the rest of the lower boundary of the convex hull is simply the curve itself).

Rather than keeping track of the sets Λ_n, it will be simpler to study exclusively their lower boundary, which will be curves connecting the points $(1, 0)$ and $(0, 1)$. All the optimal protocols will live on these curves, and all points below the curves will be unattainable. To formalize the notion

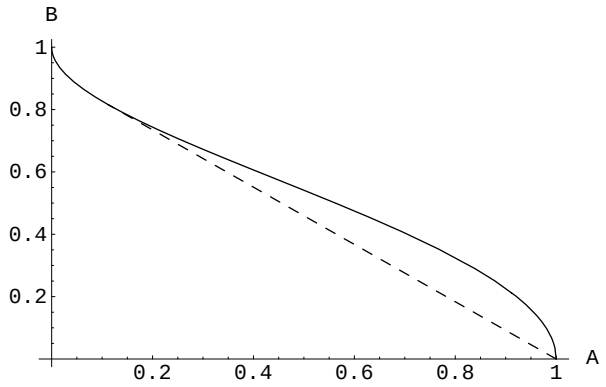


Figure 6.2: The curve $(t^2, \sqrt{1-t})$ for $t \in [0,1]$. The convex hull of the curve is the region Λ_2, with the dashed line serving as lower boundary.

of lower boundary we associate to every function $f(z) : [0,1] \to [0,1]$ the following sets:

$$f^+ = \{(z,w) \,|\, z \in [0,1], f(z) < w \leq 1\}, \tag{6.78}$$

$$f^= = \{(z,w) \,|\, z \in [0,1], f(z) = w\}, \tag{6.79}$$

$$f^- = \{(z,w) \,|\, z \in [0,1], f(z) > w \geq 0\}. \tag{6.80}$$

Returning to the case of Λ_2 and Fig. 6.2, we see that the lower boundary follows the original curve $\sqrt{1-\sqrt{z}}$ between $(1,0)$ and some point that we shall call (α_2, β_2). It then turns into a straight line connecting the point (α_2, β_2) to the point $(0,1)$. The point (α_2, β_2) can be found by calculating the slope of the line connecting each point to $(0,1)$ and choosing the point that achieves the maximum.

In fact, all of the lower boundaries will have this form. Define for $n > 1$

$$f_n(z) = \begin{cases} \sqrt{1 - \left(\frac{1-\beta_n^2}{\sqrt{\alpha_n}}\right)\sqrt{z}} & \text{for } z \in [0, \alpha_n], \\ \frac{\beta_n}{1-\alpha_n}(1-z) & \text{for } z \in [\alpha_n, 1], \end{cases} \tag{6.81}$$

where

$$\alpha_n = \frac{n-1}{3(n+1)}, \qquad \beta_n = \sqrt{\frac{n+2}{3n}}. \tag{6.82}$$

For the case $n = 1$ we define $f_1(z) = 1 - z$, which is the limit of f_n as $n \to 1$. Because $\alpha_n \in (0,1)$ and $\beta_n \in (0,1)$ for all $n > 1$, the functions satisfy $f_n(z) \in [0,1]$ for all $z \in [0,1]$. These functions are also the lower boundaries of convex regions:

Lemma 21. *For all $n \geq 1$, the function f_n is strictly decreasing, and the region $f_n^= \cup f_n^+$ is convex.*

Proof. The case of $n = 1$ is trivial. For $n > 1$ we have

$$f_n'(z) = \begin{cases} -\frac{1-\beta_n^2}{4\sqrt{\alpha_n}\sqrt{z}f_n(z)} & \text{for } z \in [0, \alpha_n], \\ -\frac{\beta_n}{1-\alpha_n} & \text{for } z \in [\alpha_n, 1], \end{cases} \tag{6.83}$$

which is well defined and negative on $(0, 1]$. For z near zero, $f(z) \simeq 1-(1-\beta_n^2)/(2\sqrt{\alpha_n})\sqrt{z}$, therefore $f(z)$ is also strictly decreasing at $z = 0$.

The derivative is also continuous on $(0, 1]$ because at $z = \alpha_n$ we have

$$\frac{\beta_n}{1-\alpha_n} = \frac{\sqrt{3}(n+1)}{2\sqrt{n(n+2)}} = \frac{1-\beta_n^2}{4\alpha_n\beta_n}. \tag{6.84}$$

Furthermore, in the region $(0, \alpha_n)$, the second derivative is

$$f_n''(z) = f_n'(z)\left[-\frac{1}{2z} - \frac{f_n'(z)}{f_n(z)}\right] = \frac{-f_n'(z)}{4z\sqrt{\alpha_n}f_n^2(z)}\left[2\sqrt{\alpha_n} - 3(1-\beta_n^2)\sqrt{z}\right] > 0 \tag{6.85}$$

where the inequality holds because $3(1-\beta_n^2) < 2$. Therefore $f_n'(z)$ is monotonically increasing on $(0, 1]$, and the region above $f_n(z)$ in this interval is convex. The point $(0, 1)$ can be included because the closure of a convex set is convex. □

We are now ready to prove the main lemma of this section.

Lemma 22. *For all $n \in \mathbb{Z}^+$, $\Lambda_n \subset f_n^= \cup f_n^+$ and $f_n^= \subset \Lambda_n$.*

Proof. The statement is clearly true for $n = 1$ since $\Lambda_1 = f_1^=$. We will prove the rest of the cases inductively. Assume the theorem holds for Λ_n, which implies that $(z, f_n(z)) \in \Lambda_n$ for all z. By Lemma 20 we have that $(f_n^2(z), \sqrt{z}) \in \Lambda_{n+1}$ for all $z \in [0, 1]$ and so are convex combinations of pairs of such points. The curve parametrized by $(f_n^2(z), \sqrt{z})$ can also be described by the points $(w, g_n(w))$ for

$$g_n(w) = \begin{cases} \sqrt{1 - \left(\frac{1-\alpha_n}{\beta_n}\right)\sqrt{w}} & \text{for } w \in [0, \beta_n^2], \\ \frac{\sqrt{\alpha_n}}{1-\beta_n^2}(1-w) & \text{for } w \in [\beta_n^2, 1]. \end{cases} \tag{6.86}$$

Note how under the map $(x, y) \to (y^2, \sqrt{x})$ the straight line turns into a curve, and the curve turns into a straight line. Furthermore, because of the exchange of x and y, the straight line ends up on the right-hand side.

The pattern of points α_n and β_n, in addition to guaranteeing that the region above $f_n(z)$ is convex, also satisfies the recursion relation

$$\frac{1-\alpha_n}{\beta_n} = \frac{2\sqrt{n(n+2)}}{\sqrt{3}(n+1)} = \frac{1-\beta_{n+1}^2}{\sqrt{\alpha_{n+1}}} \tag{6.87}$$

and therefore $g_n(z) = f_{n+1}(z)$ in the region $[0, \alpha_{n+1}]$ (since $\alpha_{n+1} \leq 1/3 \leq \beta_n^2$). Pictorially, the curve $g_n^=$ is like the curve $f_{n+1}^=$, except that the straight line intersects the curve somewhat to the right, and hence the region above $g_n^=$ is not convex. Its convex hull will give us the region above the curve $f_{n+1}^=$.

Thus far we have shown $g_n^= \subset \Lambda_{n+1}$, as are convex combinations of pairs of points on the curve $g_n^=$. Because $g_n^= = f_{n+1}^=$ in the region $[0, \alpha_{n+1}]$ we know that this segment of the curve is in Λ_{n+1}. The rest of the curve $f_{n+1}^=$ is simply the convex combination of the points $(\alpha_{n+1}, \beta_{n+1})$ and $(1, 0)$ both of which are in $g_n^=$. We have therefore proven the second part of the lemma: $f_{n+1}^= \subset \Lambda_{n+1}$.

We now intend to prove that $g_n(z) \geq f_{n+1}(z)$ for all $z \in [0, 1]$. The statement is clearly true in the region $[0, \alpha_{n+1}]$ where both are equal. In the region $[\beta_n^2, 1]$ it is also true because both functions are straight lines ending in $(1, 0)$, and the starting point of the lines are $g_n(\beta_n^2) = \sqrt{\alpha_n}$ and $f_{n+1}(\beta_n^2) = \beta_{n+1}(1 - \beta_n^2)/(1 - \alpha_{n+1})$. The inequality $f_{n+1}(\beta_n^2) \geq g_n(\beta_n^2)$ can be proven by checking that $[f_{n+1}(\beta_n^2)/g_n(\beta_n^2)]^2 - 1 = -4/[n^2(n+3)] \leq 0$ for $n \geq 1$. Finally, in the region $[\alpha_{n+1}, \beta_n^2]$ the functions $f_{n+1}(z)$ and $g_n(z)$ start off at the same point, with the same derivative, but $f''_{n+1}(z) = 0$ in this region whereas $g''_n(z)$ initially is positive and has only one zero in the region, which can be checked as in Eq. (6.85). If the curve g_n were to cross the curve f_{n+1} at any point in this region, then it would have to end below it. However, we already argued that $g_n(\beta_n^2) \geq f_{n+1}(\beta_n^2)$, and therefore the curve $g_n^=$ must lie above the curve $f_{n+1}^=$ in the middle region as well.

So far we have shown that $(g_n^= \cup g_n^+) \subset (f_{n+1}^= \cup f_{n+1}^+)$. By the induction assumption, $\Lambda_n \subset f_n^= \cup f_n^+$. Under the map $(x, y) \to (y^2, \sqrt{x})$, the region $f_n^= \cup f_n^+$ maps into the region to the right of the curve $g_n^=$, which also equals the region $g_n^= \cup g_n^+$ because $g_n(z)$ is strictly decreasing, $g_n(1) = 0$ and $g_n(0) = 1$. Finally, using Lemma 20 we know that Λ_{n+1} is contained in the convex combination of points in $g_n^= \cup g_n^+$. Because $(g_n^= \cup g_n^+) \subset (f_{n+1}^= \cup f_{n+1}^+)$ and $f_{n+1}^= \cup f_{n+1}^+$ is convex, we have $\Lambda_{n+1} \subset f_{n+1}^= \cup f_{n+1}^+$. □

Combining the previous lemma with the definition of the sets Λ_n, we have proven the following theorem:

Theorem 23. *Every $(n+1)$-message quantum weak coin-flipping protocol based on an n-Coin-Game satisfies*

$$P_B P_B^* \geq f_n^2(P_A P_A^*). \tag{6.88}$$

Additionally, we have the following corollary for the limit of $n \to \infty$:

Corollary 24. *All quantum weak coin-flipping protocols based on an n-Coin-Game (for any $n \in \mathbb{Z}^+$)*

satisfy

$$P_A P_A^* \le \frac{1}{3} \quad \Longrightarrow \quad P_B P_B^* \ge 1 - 2\sqrt{\frac{P_A P_A^*}{3}} \ge \frac{1}{3} \tag{6.89}$$

$$P_B P_B^* \le \frac{1}{3} \quad \Longrightarrow \quad P_A P_A^* \ge 1 - 2\sqrt{\frac{P_B P_B^*}{3}} \ge \frac{1}{3} \tag{6.90}$$

In particular,

$$\max\left(P_A P_A^*, P_B P_B^*\right) \ge \frac{1}{3}, \tag{6.91}$$

and

$$\max(P_A^*, P_B^*) \ge \frac{2}{3} \quad \textit{for} \;\; P_A = P_B = \frac{1}{2}. \tag{6.92}$$

Proof. The above results use the limit:

$$f_\infty(z) = \begin{cases} \sqrt{1 - \frac{2}{\sqrt{3}}\sqrt{z}} & \text{for } z \in [0, \frac{1}{3}], \\ \frac{\sqrt{3}}{2}(1-z) & \text{for } z \in [\frac{1}{3}, 1], \end{cases} \tag{6.93}$$

which has the symmetry $b = f_\infty^2(a) \Rightarrow a = f_\infty^2(b)$. □

6.4 Optimal protocols

In this section we will describe protocols that match the lower bounds derived in the previous section. In a sense, most of the work has already been done since the proof of the previous section was constructive. What remains undone is to explicitly construct the Game-Trees and to calculate from them P_A, P_A^* and P_B^* (rather than only their products).

From the discussion of the previous section we can see that the interesting Game-Trees of depth $n+1$ live on the curve $f_{n+1}^{=}$. The points on the rounded part of the curve (the left segment) involve no convex combinations of points from n-Game-Trees and therefore are not new (i.e., they are protocols that can be described by a single n-Game-Tree with Alice's and Bob's role reversed). The interesting points at level $n+1$ lie on the straight segment and are the combination of the points $(\alpha_{n+1}, \beta_{n+1})$ and $(1, 0)$. To understand this segment we need to describe the n-Game-Trees that produce points $(\beta_{n+1}^2, \sqrt{\alpha_{n+1}})$ and $(0, 1)$. The second point corresponds to a tree that is fairly simple: it has the value 1 at every leaf, and the rest of the nodes are irrelevant. The n-Game-Tree for $(\beta_{n+1}^2, \sqrt{\alpha_{n+1}})$ is what we shall describe next.

Lemma 25. *For each $n \in \mathbb{Z}^+$ there is an n-Game-Tree, $\mathcal{G}^{(n)}$, such that*

$$\mathcal{A}_r^{(n)} = \beta_{n+1}^2 = \frac{n+3}{3(n+1)}, \tag{6.94}$$

$$\mathcal{B}_r^{(n)} = \sqrt{\alpha_{n+1}} = \sqrt{\frac{n}{3(n+2)}}, \tag{6.95}$$

$$\mathcal{H}_r^{(n)} = \begin{cases} \frac{n}{2(n+1)} & n \text{ even}, \\ \frac{n+1}{2(n+2)} & n \text{ odd}, \end{cases} \tag{6.96}$$

with $\mathcal{A}^{(n)}$, $\mathcal{B}^{(n)}$ and $\mathcal{H}^{(n)}$ defined in terms of $\mathcal{G}^{(n)}$ by Eqs. (6.1, 6.2). In particular, the associated quantum weak coin-flipping protocols have:

$$P_A(n) = 1 - \mathcal{H}_r^{(n)} = \begin{cases} \frac{n+2}{2(n+1)} & n \text{ even}, \\ \frac{n+3}{2(n+2)} & n \text{ odd}, \end{cases} \tag{6.97}$$

$$P_A^*(n) = \frac{\mathcal{A}_r^{(n)}}{1-\mathcal{H}_r^{(n)}} = \begin{cases} \frac{2(n+3)}{3(n+2)} & n \text{ even}, \\ \frac{2(n+2)}{3(n+1)} & n \text{ odd}, \end{cases} \tag{6.98}$$

$$P_B^*(n) = \frac{\left(\mathcal{B}_r^{(n)}\right)^2}{\mathcal{H}_r^{(n)}} = \begin{cases} \frac{2(n+1)}{3(n+2)} & n \text{ even}, \\ \frac{2n}{3(n+1)} & n \text{ odd}. \end{cases} \tag{6.99}$$

Proof. Define the parameters

$$\gamma_n = \frac{n}{n+2}, \tag{6.100}$$

which are the weights needed for the convex combinations. And let

$$\mathcal{G}_r^{(1)} = \gamma_1, \quad \mathcal{G}_0^{(1)} = 1, \quad \mathcal{G}_1^{(1)} = 0, \tag{6.101}$$

which leads to $\mathcal{A}_r^{(1)} = 2/3$ and $\mathcal{B}_r^{(1)} = \mathcal{H}_r^{(1)} = 1/3$. The rest of the Game-Trees are defined inductively:

$$\mathcal{G}_r^{(n)} = \gamma_n, \tag{6.102}$$

$$\mathcal{G}_{0x}^{(n)} = \begin{cases} 1 - \mathcal{G}_x^{(n-1)} & \text{for } |x| = n-1, \\ \mathcal{G}_x^{(n-1)} & \text{for } |x| < n-1, \end{cases} \tag{6.103}$$

$$\mathcal{G}_{1x}^{(n)} = \begin{cases} 0 & \text{for } |x| = n-1, \\ \mathcal{G}_x^{(n-1)} & \text{for } |x| < n-1. \end{cases} \tag{6.104}$$

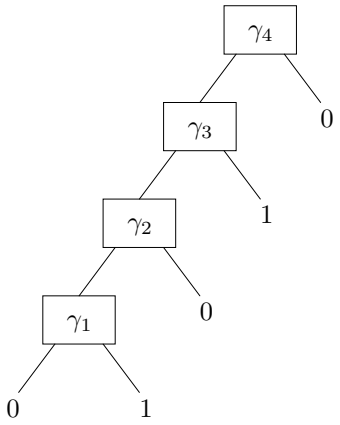


Figure 6.3: A truncated tree equivalent to $\mathcal{G}^{(4)}$.

The values of $\mathcal{G}_{1x}^{(n)}$ for $|x| < n-1$ are actually irrelevant but were chosen so that $\mathcal{G}_x^{(n)} = \gamma_{n-|x|}$ whenever $|x| < n-1$.

The reason for inverting the value of the leaves relates to our insistence that Alice always send the first message, which implies that the sender of the last message alternates as n is increased and correspondingly the assignments of winning and losing for the coin outcome need to be flipped.

In fact, the pattern of the leaves is fairly simple. It is chosen so that it depends on the parity of the location (from left to right) of the first 1 symbol in the string x. In the quantum protocol this translates into the first sender of a 1 qubit being the winner of the coin-flip (assuming they pass the cheat detection phase).

In fact, the trees $\mathcal{G}^{(n)}$ would best be described by truncated trees of the form of Fig 6.3. However, we shall continue using trees with all leaves at the same depth in order to be consistent with the previous section.

Returning to the proof of the lemma, it is easy to see that $\mathcal{A}_{1x}^{(n)} = 1$ and $\mathcal{B}_{1x}^{(n)} = \mathcal{H}_{1x}^{(n)} = 0$ for all strings x. The left side of the tree satisfies $\mathcal{A}_{0x}^{(n)} = \mathcal{B}_x^{(n-1)}$, $\mathcal{B}_{0x}^{(n)} = \mathcal{A}_x^{(n-1)}$ and $\mathcal{H}_{0x}^{(n)} = 1 - \mathcal{H}_x^{(n-1)}$ for all strings x. Therefore the root nodes are

$$\mathcal{A}_r^{(n)} = \gamma_n \left(\mathcal{B}_r^{(n-1)}\right)^2 + (1-\gamma_n)\ 1, \tag{6.105}$$

$$\mathcal{B}_r^{(n)} = \gamma_n \sqrt{\mathcal{A}_r^{(n-1)}} + (1-\gamma_n)\ 0, \tag{6.106}$$

$$\mathcal{H}_r^{(n)} = \gamma_n \left(1 - \mathcal{H}_r^{(n-1)}\right) + (1-\gamma_n)\ 0. \tag{6.107}$$

It is then straightforward to plug in the expressions as functions of n for all the above parameters and check that Eqs. (6.94–6.96) are always satisfied. □

Interestingly, the sequence of protocols is such that P_A and P_B do not change when n increases from an odd integer to an even one, whereas P_A^* and P_B^* do not change when n increases from an

even integer to an odd one. We offer no intuition for this property. Note, however, that for a given n, the associated protocol corresponds to a single point on the surface of optimal protocols in the 3-dimensional space of triplets (P_A, P_A^*, P_B^*) that can be achieved with $n+1$ quantum messages.

For large n, the sequence of protocols converges to $P_A = P_B = 1/2$ and $P_A^* = P_B^* = 2/3$, yielding a protocol with bias of $1/6$. It would also be desirable to show the existence of a sequence of protocols that converges to the same point but such that $P_A = P_B = 1/2$ for every protocol in the sequence. This can be easily accomplished by choosing, for each n, the point along the curve $f_n^=$ that has $\mathcal{H}_r = 1/2$. In the Game-Tree language we need to modify the top coin $\mathcal{G}_r$, and we therefore introduce a new sequence of Game-Trees $\mathcal{G}'^{(n)}$ defined as

$$\mathcal{G}_x'^{(n)} = \begin{cases} 1/\left(2 - 2\mathcal{H}_r^{(n-1)}\right) & x = r, \\ \mathcal{G}_x^{(n)} & \text{otherwise.} \end{cases} \tag{6.108}$$

For simplicity, we will concentrate on the case when n is even so that:

$$\mathcal{A}_r'^{(n)} = \frac{n+1}{n+2}\left(\mathcal{B}_r^{(n-1)}\right)^2 + \frac{1}{n+2}, \tag{6.109}$$

$$\mathcal{B}_r'^{(n)} = \frac{n+1}{n+2}\sqrt{\mathcal{A}_r^{(n-1)}}, \tag{6.110}$$

$$\mathcal{H}_r'^{(n)} = \frac{n+1}{n+2}(1 - \mathcal{H}_r^{(n-1)}) = \frac{1}{2} \tag{6.111}$$

and the associated probabilities of winning by cheating are

$$P_A^*(n)' = 2\mathcal{A}_r'^{(n)} = \frac{2}{3}, \tag{6.112}$$

$$P_B^*(n)' = 2\left(\mathcal{B}_r'^{(n)}\right)^2 = \frac{2}{3}\,\frac{(n+1)^2}{n(n+2)}. \tag{6.113}$$

That is, we have identified a nice sequence of quantum protocols with $n+1$ messages (for n even) where $P_A = P_B = 1/2$ and $P_A^* = 2/3$ are all fixed and P_B^* decreases from $3/4$ to $2/3$. Of course, the case $n = 2$ belongs to the family studied by Spekkens and Rudolph [SR02b] and satisfies $P_A^* P_B^* = 1/2$.

As discussed in the introduction to the previous section, the above protocols are optimal in the following sense: to decrease one of P_A^* or P_B^* while keeping the number of messages fixed, we would have to increase the other parameter. However, the protocols are not optimal in the sense that they minimize the bias $\epsilon = \max(P_A^*, P_B^*) - 1/2$ for a fixed number of messages. Only in the limit of infinite messages is the bias of the above protocols optimal.

Thus far, we have identified the point $(1/3, \sqrt{1/3}) \in f_\infty^=$ as a protocol with $P_A = 1/2$ and $P_A^* = P_B^* = 2/3$. The other points on the curve $f_\infty^=$ can be found using the same trick of modifying the top coin $\mathcal{G}_r$. That is, let $\mathcal{G}'^{(n)}$ be as above but with $\mathcal{G}_r'^{(n)} = t$, where $t \in [0,1]$ is a parameter we

can choose freely. In the limit of $n \to \infty$ we find:

$$\mathcal{A}'^{(\infty)}_r(t) = t\,\frac{1}{3} + (1-t), \tag{6.114}$$

$$\mathcal{B}'^{(\infty)}_r(t) = t\,\sqrt{\frac{1}{3}}, \tag{6.115}$$

$$\mathcal{H}'^{(\infty)}_r(t) = t\,\frac{1}{2}. \tag{6.116}$$

The associated quantum weak coin-flipping parameters are

$$P_A(t) = 1 - \frac{t}{2}, \tag{6.117}$$

$$P^*_A(t) = \frac{2}{3}\,\frac{3-2t}{2-t}, \tag{6.118}$$

$$P^*_B(t) = \frac{2}{3}\,t. \tag{6.119}$$

These protocols correspond to the right half of the curve $f^{=}_{\infty}$ (i.e., the points $(z, f_\infty(z))$ for $z \in [1/3, 1]$). The other half of the curve can be obtained by symmetry between Alice and Bob. In the Game-Tree formalism this symmetry arises by creating a new $(n+1)$-Game-Tree, $\mathcal{G}'$, out of given n-Game-Tree, $\mathcal{G}$, by the rules $\mathcal{G}'_r = 1$, $\mathcal{G}'_{0x} = \mathcal{G}'_{1x} = \mathcal{G}_x$ for $|x| < n$ and $\mathcal{G}'_{0x} = \mathcal{G}'_{1x} = 1 - \mathcal{G}_x$ for $|x| = n$. In the language of protocols, we are forcing Alice's first message to have no content, which is equivalent to allowing Bob to begin the game.

The results can be best summarized by eliminating the variable t from Eqs. (6.117–6.119), which proves this section's main theorem:

Theorem 26. *There exist quantum weak coin-flipping protocols that asymptotically approach the curve*

$$P^*_A + P^*_B - \frac{3}{4} P^*_A P^*_B = 1 \tag{6.120}$$

in the limit of large number of messages. The corresponding probabilities of winning when the game is played honestly are

$$P_A = \frac{3}{4}\,P^*_A \quad \textit{when } P^*_A \le P^*_B, \tag{6.121}$$

$$P_B = \frac{3}{4}\,P^*_B \quad \textit{when } P^*_A \ge P^*_B. \tag{6.122}$$

6.5 Conclusions

We have identified a large family of quantum protocols for weak coin-flipping, that are based on classical public-coin games. The family contains protocols approaching the curve $P^*_A + P^*_B - \frac{3}{4} P^*_A P^*_B =$

1, which can be reached asymptotically in the limit of large number of messages. The most important of these protocols is symmetric between Alice and Bob and achieves $P_A = P_B = 1/2$ and $P_A^* = P_B^* = 2/3$, that is, it has a bias of 1/6.

Furthermore, we have proven lower bounds for the bias achievable by protocols in this family. In particular, $\max(P_A^*, P_B^*) \geq 2/3$ or equivalently $\epsilon \geq 1/6$. These lower bounds show that the protocols found above are optimal within their family.

Our lower bounds also establish a strict hierarchy among coin-flipping protocols in our family with different number of messages. Admittedly, the hierarchy is of little practical interest since a small number of messages suffices in all cases to construct protocols that are reasonably close to optimal.

The question of optimal bias for a general quantum weak coin-flipping protocol remains open. We speculate that it might be possible to show that every protocol is equivalent to one contained in the family analyzed in this chapter. Future work will be needed to verify this conjecture.

Though it is not clear that the protocols presented in this thesis will ever find a practical application, they do demonstrate the power of quantum information, as summarized by the beautiful and simple equation

$$\frac{1}{6} \leq \frac{1}{2}. \tag{6.123}$$

Bibliography

[ABDR04] A. Ambainis, H. Buhrman, Y. Dodis and H. Roehrig, Multiparty Quantum Coin Flipping, in *19th IEEE Annual Conference on Computational Complexity*, pages 250–259, IEEE Computer Society, 2004, quant-ph/0304112.

[Amb01] A. Ambainis, A New Protocol and Lower Bounds for Quantum Coin Flipping, in *33rd Symposium on Theory of Computing (STOC '01)*, pages 134–142, ACM Press, 2001, quant-ph/0204022.

[Amb02] A. Ambainis, *Lower bound for a class of weak quantum coin flipping protocols*, (2002), quant-ph/0204063.

[ATSVY00] D. Aharonov, A. Ta-Shma, U. Vazirani and A. Yao, Quantum Bit Escrow, in *32nd Symposium on Theory of Computing (STOC '00)*, pages 705–724, ACM Press, 2000, quant-ph/0004017.

[Bar89] D. A. Barrington, *Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1*, J. Comput. Syst. Sci. **38**, 150–164 (1989).

[BB84] C. H. Bennet and G. Brassard, Quantum Cryptography: Public Key Distribution and Coin Tossing, in *IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, IEEE Computer Society, 1984.

[Blu81] M. Blum, Coin flipping by telephone, in *Advances in Cryptology: A Report on CRYPTO '81*, edited by A. Gersho, pages 11–15, Santa Barbara, 1981, ECE Report No 82-04.

[BOGW88] M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, in *20th Symposium on Theory of Computing (STOC '88)*, pages 1–10, ACM Press, 1988.

[BOL85] M. Ben-Or and N. Linial, Collective coin flipping, robust voting schemes and minima of Banzhaf values, in *26th Symposium on Foundations of Computer Science (FOCS '85)*, pages 408–416, IEEE Computer Society, 1985.

[BST90] D. A. M. Barrington, H. Straubing and D. Therien, *Non-uniform Automata over Groups*, Information and Computation **89**, 109–132 (1990).

[Bur55] W. Burnside, *Theory of Groups of Finite Order*, Dover, New York, second edition, 1955, Note M.

[BvDdWP92] F. A. Bais, P. van Driel and M. de Wild Propitius, *Quantum symmetries in discrete gauge theories*, Phys. Lett. B **280**, 63–70 (1992), hep-th/9203046.

[CCD88] D. Chaum, C. Crepeau and I. Damgard, Multiparty unconditionally secure protocols, in *20th Symposium on Theory of Computing (STOC '88)*, pages 11–19, ACM Press, 1988.

[CFH97] D. G. Cory, A. F. Fahmy and T. F. Havel, *Ensemble quantum computing by NMR spectroscopy*, Proc. Natl. Acad. Sci. USA **94**, 1634–1639 (1997).

[CGS02] C. Crepeau, D. Gottesman and A. Smith, Secure multi-party quantum computation, in *34th Symposium on Theory of Computing (STOC '02)*, pages 643–652, ACM Press, 2002, quant-ph/0206138.

[CY95] I. L. Chuang and Y. Yamamoto, *Simple quantum computer*, Phys. Rev. A **52**, 3489–3496 (1995), quant-ph/9505011.

[CZ95] J. I. Cirac and P. Zoller, *Quantum Computations with Cold Trapped Ions*, Phys. Rev. A **74**, 4091–4094 (1995).

[DDL02] L.-M. Duan, E. Demler and M. D. Lukin, *Controlling Spin Exchange Interactions of Ultracold Atoms in Optical Lattices*, (2002), cond-mat/0210564.

[DIV03] B. Doucot, L. B. Ioffe and J. Vidal, *Discrete non-Abelian gauge theories in two-dimensional lattices and their realizations in Josephson-junction arrays*, (2003), cond-mat/0302104.

[dWPB95] M. de Wild Propitius and F. A. Bais, *Discrete gauge theories*, (1995), hep-th/9511201, Lectures presented at Particles and Fields 94.

[FKLW01] M. Freedman, A. Kitaev, M. Larsen and Z. Wang, *Topological Quantum Computation*, (2001), quant-ph/0101025.

[Fre00] M. Freedman, *Quantum Computation and the localization of Modular Functors*, (2000), quant-ph/0003128.

[FT63] W. Feit and J. G. Thompson, *Solvability of Groups of Odd Order*, Pac. J. Math **13**, 775–1029 (1963).

[GC97] N. A. Gershenfeld and I. L. Chuang, *Bulk Spin-Resonance Quantum Computation*, Science **275**, 350–356 (1997).

[GLS94] D. Gorenstein, R. Lyons and R. Solomon, *The Classification of the Finite Simple Groups*, AMS, Providence, RI, 1994.

[GMW87] O. Goldreich, S. Micali and A. Wigderson, How to play ANY mental game, in *19th Symposium on Theory of Computing (STOC '87)*, pages 218–229, ACM Press, 1987.

[Got98] D. Gottesman, Fault-Tolerant Quantum Computation with Higher-Dimensional Systems, in *QCQC '98*, edited by C. P. Williams, pages 302–313, Berlin, 1998, Springer-Verlag, quant-ph/9802007.

[Gro96] L. K. Grover, A fast quantum mechanical algorithm for database search, in *28th Symposium on Theory of Computing (STOC '96)*, pages 212–219, ACM Press, 1996, quant-ph/9605043.

[GVW99] L. Goldenberg, L. Vaidman and S. Wiesner, *Quantum Gambling*, Phys. Rev. Lett. **82**, 3356–3359 (1999), quant-ph/9808001.

[Hal02] S. Hallgren, Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem, in *34th Symposium on Theory of Computing (STOC '02)*, pages 653–658, ACM Press, 2002.

[HK04] L. Hardy and A. Kent, *Cheat Sensitive Quantum Bit Commitment*, Phys. Rev. Lett. **92**, 157901 (2004), quant-ph/9911043.

[KBDW01] J. Kempe, D. Bacon, D. P. DiVincenzo and K. B. Whaley, *Encoded Universality from a Single Physical Interaction*, Quantum Information and Computation **1**, 33–55 (2001), quant-ph/0112013.

[Ken99] A. Kent, *Coin Tossing is Strictly Weaker than Bit Commitment*, Phys. Rev. Lett. **83**, 5382–5384 (1999), quant-ph/9810067.

[Kil88] J. Kilian, Founding cryptography on oblivious transfer, in *20th Symposium on Theory of Computing (STOC '89)*, pages 20–31, ACM Press, 1988.

[Kit95] A. Kitaev, *Quantum measurements and the Abelian Stabilizer Problem*, (1995), quant-ph/9511026.

[Kit97a] A. Kitaev, *Fault-tolerant quantum computation by anyons*, (1997), quant-ph/9707021.

[Kit97b] A. Kitaev, *Quantum Computations: algorithms and error correction*, Russ. Math. Surv. **52**, 1191 (1997).

[Kit02] A. Kitaev, 2002, Private communication.

[Kit03] A. Kitaev, 2003, Results presented at QIP 2003 (slides and video available from MSRI).

[KN04] I. Kerenidis and A. Nayak, *Weak coin flipping with small bias*, Inf. Process. Lett. **89**, 131–135 (2004).

[LC98] H.-K. Lo and H. F. Chau, *Why quantum bit commitment and ideal quantum coin tossing are impossible*, Physica D **120**, 177–187 (1998), quant-ph/9711065.

[Lo97] H.-K. Lo, *Insecurity of quantum secure computations*, Phys. Rev. A **56**, 1154–1162 (1997), quant-ph/9611031.

[May96] D. Mayers, *Unconditionally secure quantum bit commitment is impossible*, (1996), quant-ph/9605044.

[Moc03] C. Mochon, *Anyons from non-solvable finite groups are sufficient for universal quantum computation*, Phys. Rev. A **67**, 022315 (2003), quant-ph/0206128.

[Moc04a] C. Mochon, *Anyon computers with smaller groups*, Phys. Rev. A **69**, 032306 (2004), quant-ph/0306063.

[Moc04b] C. Mochon, Quantum weak coin-flipping with bias of 0.192, in *45th Symposium on Foundations of Computer Science (FOCS '04)*, pages 2–11, IEEE Computer Society, 2004, quant-ph/0403193.

[Moc04c] C. Mochon, *Serial composition of quantum coin-flipping, and bounds on cheat detection for bit-commitment*, Phys. Rev. A **70**, 032312 (2004), quant-ph/0311165.

[Moc05] C. Mochon, *A large family of quantum weak coin-flipping protocols*, (2005), quant-ph/0502068.

[MR65] W. D. Maurer and J. L. Rhodes, *A Property of Finite Simple Non-Abelian Groups*, Proc. Am. Math. Soc. **16**(3), 552–554 (1965).

[NC00] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, First edition, 2000.

[NW96] C. Nayak and F. Wilczek, *2n-quasihole states realize 2^{n-1}-dimensional spinor braiding statistics in paired quantum Hall states*, Nucl. Phys. B **479**, 529–553 (1996), cond-mat/9605145.

[OP99] R. W. Ogburn and J. Preskill, Topological Quantum Computation, in *QCQC '98*, edited by C. P. Williams, pages 341–356, Berlin, 1999, Springer-Verlag.

[Pre97] J. Preskill, *Fault-tolerant quantum computation*, (1997), quant-ph/9712048.

[RBO89] T. Rabin and M. Ben-Or, Verifiable secret sharing and multiparty protocols with honest majority, in *21th Symposium on Theory of Computing (STOC '89)*, pages 73–85, ACM Press, 1989.

[RR99] N. Read and E. Rezayi, *Beyond paired quantum Hall states: Parafermions and incompressible states in the first excited Landau level*, Phys. Rev. B **59**, 8084–8092 (1999), cond-mat/9609079.

[RS04] T. Rudolph and R. W. Spekkens, *Quantum state targeting*, Phys. Rev. A **70**, 052306 (2004), quant-ph/0310060.

[Sch64] I. J. Schoenberg, *A note on the cyclotomic polynomial*, Mathematika **11**, 131–136 (1964).

[Sha] W. Shakespeare, *The Tragedy of Hamlet, Prince of Denmark*, Washington Square Press, New York, August 2004 edition, Act 3, Scene 1.

[Shi02] Y. Shi, *Both Toffoli and Controlled-NOT need little help to do universal quantum computation*, (2002), quant-ph/0205115.

[Sho94] P. W. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring, in *35th Symposium on Foundations of Computer Science (FOCS '94)*, pages 124–134, IEEE Computer Society, 1994.

[Sho95] P. W. Shor, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A. **52**, 2493–2496 (1995).

[Sho96] P. W. Shor, Fault-tolerant quantum computation, in *37th Symposium on Foundations of Computer Science (FOCS '96)*, pages 56–65, IEEE Computer Society, 1996, quant-ph/9605011.

[SR02a] R. W. Spekkens and T. Rudolph, *Degrees of concealment and bindingness in quantum bit commitment protocols*, Phys. Rev. A **65**, 012310 (2002), quant-ph/0106019.

[SR02b] R. W. Spekkens and T. Rudolph, *Quantum Protocol for Cheat-Sensitive Weak Coin Flipping*, Phys. Rev. Lett. **89**, 227901 (2002), quant-ph/0202118.

[Ste96] A. M. Steane, *Error Correcting Codes in Quantum Theory*, Phys. Rev. Lett. **77**, 793–797 (1996).

[SV86] M. Santha and U. V. Vazirani, *Generating Quasi-random Sequences from Semi-random Sources*, J. Comput. Syst. Sci. **33**, 75–87 (1986).

[SV99] L. J. Schulman and U. Vazirani, Scalable NMR Quantum Computation, in *31st Symposium on Theory of Computing (STOC '99)*, pages 322–329, ACM Press, 1999, quant-ph/9804060.

[THL+95] Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi and H. J. Kimble, *Measurement of Conditional Phase Shifts for Quantum Logic*, Phys. Rev. Lett. **75**, 4710–4713 (1995), quant-ph/9511008.

[vD04] W. van Dam, *Quantum Computing and Zeroes of Zeta Functions*, (2004), quant-ph/0405081.

[Wie83] S. Wiesner, *Conjugate coding*, SIGACT News **15**, 77 (1983).

[Yao82] A. C. Yao, Protocols for Secure Computation, in *23rd Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164, IEEE Computer Society, 1982.

[Yao95] A. C. Yao, Security of quantum protocols against coherent measurements, in *27th Symposium on Theory of Computing (STOC '95)*, pages 67–75, ACM Press, 1995.