RESTRICTED VERSIONS OF THE TWO-LOCAL HAMILTONIAN PROBLEM

A Dissertation in
Physics
by
Sandeep Narayanaswami

© 2013 Sandeep Narayanaswami

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2013
The dissertation of Sandeep Narayanaswami was reviewed and approved* by the following:

Sean Hallgren  
Associate Professor of Computer Science and Engineering  
Dissertation Adviser, Co-Chair of Committee

Nitin Samarth  
Professor of Physics  
Head of the Department of Physics  
Co-Chair of Committee

David S Weiss  
Professor of Physics

Jason Morton  
Assistant Professor of Mathematics

*Signatures are on file in the Graduate School.
Abstract

The Hamiltonian of a physical system is its energy operator and determines its dynamics. Understanding the properties of the ground state is crucial to understanding the system. The Local Hamiltonian problem, being an extension of the classical Satisfiability problem, is thus a very well-motivated and natural problem, from both physics and computer science perspectives. In this dissertation, we seek to understand special cases of the Local Hamiltonian problem in terms of algorithms and computational complexity.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>ix</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Background</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Classical Complexity</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Quantum Computation</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Generalizations of SAT</td>
<td>11</td>
</tr>
<tr>
<td>2.3.1 The Ising model</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2 QMA-complete Local Hamiltonians</td>
<td>13</td>
</tr>
<tr>
<td>2.3.3 Projection Hamiltonians, or Quantum $k$-SAT</td>
<td>14</td>
</tr>
<tr>
<td>2.3.4 Commuting Local Hamiltonians</td>
<td>14</td>
</tr>
<tr>
<td>2.3.5 Other special cases</td>
<td>15</td>
</tr>
<tr>
<td>2.3.6 Approximation Algorithms and Heuristics</td>
<td>15</td>
</tr>
<tr>
<td>3. 8-state particles on a line</td>
<td>17</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Background</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Encoding a computation in a sequence of orthogonal states of a line of 8-dimensional qudits</td>
<td>23</td>
</tr>
</tbody>
</table>
3.3.1 Legal configurations ........................................ 27
3.4 The Hamiltonian .................................................. 30
  3.4.1 The penalty Hamiltonian .................................... 33
  3.4.2 The propagation Hamiltonian ............................... 35
3.5 Completeness ...................................................... 45
3.6 Soundness .......................................................... 48
  3.6.1 Type 1 subspace .............................................. 49
  3.6.2 Type 3 subspace .............................................. 51
4 The Q-2-SAT problem on 1 qutrit and n-1 qubits ........................ 53
  4.1 Introduction ..................................................... 53
  4.2 Background ..................................................... 55
  4.3 Eliminating higher-rank constraints ............................ 56
    4.3.1 Constraints of rank $d_a d_b - 1$ ......................... 57
    4.3.2 Constraints of rank $d_a d_b - d_a$ ....................... 57
  4.4 Constraints of rank 1 and 2 .................................. 59
    4.4.1 Generating new constraints ............................... 59
    4.4.2 Complete set with all constraints of rank one ............ 60
    4.4.3 Complete set with at least one constraint of rank 2 .... 61
5 Discussion and Open Problems .................................. 66
  5.1 1D 2-local Hamiltonians ...................................... 66
  5.2 Quantum SAT .................................................. 67
  5.3 Other Directions .............................................. 68
Bibliography ......................................................... 70
Appendix A. 8-state particles on a line ............................ 75
  A.1 Eigenvalues .................................................. 75
Appendix B. Quantum SAT .......................................... 78
  B.1 Notes on the Bravyi paper ................................. 78
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The transition rules, which together with a carefully chosen initial state define the 2-local gates of the circuit $\tilde{V}<em>x$. Note that some of these rules are 2-local, some 3-local and some even 4-local, which helps them identify their intended locations uniquely. However, the transformations themselves are only 2-local. See also Table 3.2 for an example of a progression of configurations and the unique applicability of these rules. We will later write a Hamiltonian $H</em>{\text{prop}}$ with only 2-local terms checking these transitions.</td>
</tr>
<tr>
<td>3.2</td>
<td>The configurations occurring in one cycle of the computation with $n = 3$ qubits. The rules whose application brings the state to the next one are listed on the right.</td>
</tr>
<tr>
<td>3.3</td>
<td>Building up the legal configuration structure from pairs of symbols (unlisted symbol pairs do not appear in legal configurations). We list symbol pairs allowed at positions $(2i-1, 2i)$ and their designated followups at positions $(2i+1, 2i+2)$. Note the mirror symmetry of the table across the antidiagonal. The allowed configurations of the whole chain must then have form $(x \cdots x)[\text{qubits}](u \cdots u)$, with a substring $[\text{qubits}]$ given by (3.6)-(3.13), with at most one active pair. Further restrictions come into play from considering the block boundary locations (see Table 3.5) the number of “qubits” and their proper alignment with respect to the block boundaries.</td>
</tr>
<tr>
<td>3.4</td>
<td>The general form of the sequence of legal configurations in one round of computation. The middle block boundary is not shown in steps 7 and 8. The full computation ends at Step 5 without the trailing circles. A particular example for $n = 3$ is shown in Table 3.2</td>
</tr>
</tbody>
</table>
3.5 The 56 allowed pairs $XY$ of symbols at positions $(i, i+1)$ in the $d = 8$ construction according to $H_{pen}$. There are 5 types of locations (A, B, C, D, E) for the pair, according to location parity and block-boundary position. For each of the 36 symbol combinations, we list its allowed location types. The forbidden pairs implied by this table are penalized by $H_{pen}$ (3.17).

3.6 Substrings identifying active spots in legal configurations. We list all of the substrings appearing in the projector terms of the type $|XY\rangle \langle XY|_{j,j+1}$ and $|ZW\rangle \langle ZW|_{k,k+1}$ from all of the $H_{prop,i}^{\text{rule } \rho}$ terms (3.24), (3.26), (3.34), (3.27), (3.28) and (3.29). Finding a substring $XY$ of a legal configuration $C_t$ uniquely indicates that the configuration is connected to some configuration $C_{t+1}$ ahead of it. Similarly, finding a substring $ZW$ uniquely locates a backward transition to some $C_{t-1}$. 

viii
Acknowledgments

I would like to thank the CS theory group for welcoming into their midst this interloper from Physics. I would also like to thank the staff at the Physics and CSE departments for greatly easing the way after I made the decision to work outside the Physics department. Above all, thanks are due to Sean, who’s been a great advisor.
Chapter 1

Introduction

Computers today are cheaper and more powerful than ever before, and are pushing the limits of Moore’s law. As the components in our computers get smaller and smaller, as the size of transistors approaches the size of a single atom, quantum effects will become more and more significant. To be able to cope with these effects, we need an understanding of how Nature processes information at the smallest scales.

In the 1980s, Feynman [Fey82], among others [Ben82], proposed a new way of thinking about computation. The simulation of quantum systems was, and continues to be, a hard problem of interest to physicists and, the idea went, might be easier on a quantum computer than on a classical one. Classical computers store and transform data as bits. However, the world is quantum mechanical: quantum objects are represented by vectors in a linear vector space, and are transformed according to unitary operations. A quantum computer would then be a machine that stored its data as quantum bits or qubits (two-level quantum systems) and transformed it by the action of unitary operators or quantum gates [NC00].

Classical computations can be thought of as a series of Boolean gates acting on binary information. While the gates in ordinary classical computers are not usually reversible, it was shown [Ben73] that they could be made reversible with some overhead. Since unitary operations or gates on qubits generalize the classical reversible gates on bits, classical circuits are essentially a special case of quantum circuits. Therefore, in a quantum computing model, we are making the computer capable of doing more things. The obvious question to ask then is: does our seemingly more general picture of computing actually buy us anything in terms of efficiently solving problems that couldn’t be solved
One way to measure the performance of an algorithm is in terms of its running time. We require that the running time of an algorithm not increase too rapidly as the size of the instance increases. In particular, we consider an algorithm to be efficient if its running time is at most some polynomial in the size of the input. The simplest notion of efficiency is given by the class of problems solvable by a deterministic polynomial-time algorithm running on a classical computer. But we may also consider probabilistic algorithms. Going still further, we could allow the computer to be quantum. Do these different notions of efficiency correspond to different classes of problems? Clearly, any problem that has an efficient classical algorithm also has an efficient quantum algorithm. However, a demonstration of quantum computers’ power came when Peter Shor [Sho94] gave an efficient quantum algorithm for a problem that has so far eluded all attempts at an efficient classical solution: the factoring of composite numbers into their constituent primes. This was a particularly significant development because a fast quantum algorithm, if realized on a quantum computer, would break the security of the widely-used RSA cryptosystem.

There are many problems of theoretical and practical interest that are not known to be efficiently solvable but whose solutions can be easily/efficiently checked to be correct. If this efficient verification is done on a classical computer, this set of problems constitutes the complexity class NP. The notion of efficient solution versus efficient verification is often fleshed out using the example of a sudoku puzzle: while it is not always easy to correctly complete a partially filled sudoku grid, it is easy to verify that a given filled grid is a valid solution. The idea of efficient verification can encompass an even larger set of problems if we allow the verifying computer to be quantum. This larger class of problems is called QMA, and is often thought of as a quantum analogue of NP.

One problem in NP with no known efficient algorithm is the Boolean Satisfiability problem (SAT): given a set of clauses, for example, \((x_1 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor x_5) \land \ldots \land (\neg x_1 \lor x_5 \lor \neg x_6)\) on some Boolean variables \(\{x_i\}\), does there exist an assignment to the variables under which every clause in the formula evaluates to 1? SAT also has the property that every other problem in NP can be reduced to it: given an NP problem, there is an efficiently computable mapping from any instance of this language to an instance of SAT. Thus, if we can solve SAT efficiently, we can solve any problem in NP efficiently by first mapping (‘reducing’) it to an instance of SAT, and then using the efficient algorithm for SAT to solve the SAT instance. Problems like SAT are the
‘hardest’ problems in NP, and are called NP-complete. NP-complete problems are not thought to be efficiently solvable.

Problems like SAT, where a set of variables must satisfy certain constraints, have a neat analogue in physical systems. A physical system tends to want to be in its ground state, its state of least energy. The ground state is an eigenvector of the system’s Hamiltonian, which is usually a sum of many terms. In a many-particle system, these terms are ‘local’: each term acts only on a small number of particles. For a state of the system to have a low energy, it must be a low-energy eigenvector of all or most of the local Hamiltonian operators. That is, just as a satisfying assignment to an instance of SAT must satisfy each of the clauses, the ground state of a physical system must ‘satisfy’ each of the local constraints imposed by the terms in the Hamiltonian. Since SAT is a well-studied problem in computer science, and since the ground state and the ground state energy is key to understanding a physical system, we have a well-motivated generalization of SAT: the Local Hamiltonian problem. The input for this problem is a set of Hamiltonian operators, \( \{H_i\} \), each acting on some constant number \( k \) out of a total of \( n \) particles, and two real constants \( a \) and \( b \) (\( a < b \)). The problem is to decide if there exists a state \( \ket{\psi} \) on the \( n \) particles that has energy at most \( a \) under the system Hamiltonian \( H = \sum_i H_i \), \( \langle \psi | \sum_i H_i | \psi \rangle \leq a \) for some \( \ket{\psi} \), or if every state has energy greater than \( b \), \( \langle \psi | \sum_i H_i | \psi \rangle > b \) for every \( \ket{\psi} \). In other words, we are estimating the ground state energy: deciding if the system has a ‘low’ ground state energy (\(< a\)) or a ‘high’ one (\(> b\)). We say a term \( H_i \) in the Hamiltonian is satisfied if the state of the system is a lowest-eigenvalue eigenvector for that \( H_i \). A Hamiltonian is not always completely satisfiable: there may exist no state that is a lowest-energy eigenstate of each term in the Hamiltonian. Such a system is said to be ‘frustrated’.

Instances of SAT can be represented as instances of the Local Hamiltonian problem. For example, each clause of a 3-SAT instance corresponds to a matrix acting on three qubits, and its eigenvectors are basis vectors and have eigenvalues either 0 or 1. Then the problem is equivalent to deciding if the smallest eigenvalue of the system Hamiltonian is 0 or at least 1. The Local Hamiltonian (LH) problem is thus a generalization of SAT, and therefore, at least as hard as SAT. It is certainly hard to solve on a classical computer. Can we expect to do better with a quantum computer? It turns out the parallels between SAT and LH extended to their hardness: the Local Hamiltonian problem was shown, in a landmark result by Kitaev et al. \[KSV02\], to be hard even
on a quantum computer. In particular, just as SAT is complete for the class NP, LH was shown to be complete for QMA.

The QMA-completeness result \cite{KSV02} uses 5-local Hamiltonians. This result was improved with respect to locality: it was first shown that the problem remains QMA-complete when the Hamiltonians are only three-local \cite{KR03}, and this was further improved to two \cite{KKR06}. However, in all these results, the interactions are not spatially local: there is no restriction on where the qubits being acted upon by a term \(H_i\) are located in relation to each other. This makes the construction somewhat unrealistic, since in physical settings, interactions are usually nearest-neighbor or otherwise spatially localized. Would the problem remain hard if some physically realistic restrictions were imposed on the interaction Hamiltonians? For example, we might choose to restrict the particles to lie on some kind of lattice and to have only nearest-neighbor interactions. Since we are restricting the possible instances of the problem, we might expect that they turn out to be easier to solve. The restriction to a two-dimensional grid of qubits with nearest neighbor interactions was shown in \cite{OT08} to remain QMA-complete. There remained a further possible restriction: particles on a line with nearest-neighbor interactions. This is the simplest possible geometry, and some one-dimensional quantum systems have been easier to understand than their higher-dimensional extensions. Moreover, a one-dimensional restriction of a generalization of 2-SAT for \(d\)-state variables is in P. However, it was shown by Aharonov et al \cite{AGIK09} that LH with nearest-neighbor interactions between 13-state particles on a line remains QMA-complete. This left open the question of how the size of the particles affects hardness. In particular, could the one-dimensional problem remain hard when the particles are all 2-state? Or would there be some point between 13 and 2 where the problem has an efficient solution? In this dissertation, we show that the one-dimensional Local Hamiltonian problem remains QMA-complete when the dimension of the qudits is 8. The problem remains open for particles with fewer than eight states.

We then look at another restricted version of the Local Hamiltonian problem: Quantum SAT. This is a version of Local Hamiltonian where the constraint Hermitian operators are all projectors, and the problem is to determine if the system is unfrustrated, i.e., if it has a state that is a zero eigenvector of each term in the Hamiltonian. More formally, the input to the problem is a set of projectors \(\{\Pi_i\}\), and a positive real number \(\epsilon\), and it is to be determined if there exists a simultaneous zero eigenvector, i.e., some state \(|\psi\rangle\) such that \(\Pi_i |\psi\rangle = 0\) for all \(i\), or if every state \(|\psi\rangle\)
has energy at least $\epsilon$, i.e., $\langle \psi | \sum_i \Pi_i | \psi \rangle > \epsilon$. The classical version of this problem is 2-SAT – two-literal constraints acting on Boolean variables – and is in P. The clauses in a 2-SAT instance can be viewed as implications: a clause $(x_i \lor x_j)$ corresponds to the implications $\neg x_i \Rightarrow x_j$ and $\neg x_j \Rightarrow x_i$.

This in turn leads to a nice implication graph through which we can search for contradictions (like a path from the vertex representing $x_i$ to the vertex representing $\neg x_i$). If no contradictions are found, the instance is satisfiable.

If we replace one of the two-level variables in our 2-SAT instances with a 3-level variable (with suitably redefined constraints to accommodate the higher-level particle), it is easy to see that the problem remains easily solvable. Fixing the value of the 3-level variable gives us an instance of 2-SAT, for which we have an efficient algorithm. Running this algorithm thrice, once for each of the possible values of the 3-level variable, we can efficiently decide if our instance has a satisfying assignment. On the other hand, having too many 3-level variables can make the problem hard. In particular, if a constant fraction of the variables were three-level, we could reduce the 3-Coloring problem to this $(3,2)$-SAT problem, with the implication that the $(3,2)$-SAT problem would be intractable.

In the quantum case, there is no obvious notion of implication. Instead we have projectors giving energy penalties to subspaces of systems that live in a continuous state space. Any intuition we had for classical 2-SAT does not help us much here, and it is not obvious why we should be able to efficiently search through a vector space of exponentially large dimension to find a satisfying state. However, it was shown by Bravyi [Bra06] that Quantum SAT on qubits is solvable efficiently on a classical computer, thereby placing the problem in P.

We mentioned that, in classical 2-SAT, replacing one of the 2-level variables with a 3-level variable does not affect the complexity of the problem. How, if at all, does this extend to Quantum SAT? In the quantum case, it is not immediately obvious that replacing one of the qubits in the Quantum-SAT problem with a three-level particle (a qutrit) results in a problem with an efficient algorithm. In particular, unlike in the classical case, we can not try out all possible values of the qutrit. An efficient algorithm for such a problem seems less straightforward. In this dissertation, we show that some special cases of this problem can be efficiently solved. A completely general solution remains an open question.
Chapter 2

Background

In this chapter, we lay the groundwork for our main results by introducing the main concepts, reviewing important past results in the area, and defining the problems we address later on.

2.1 Classical Complexity

A computation is generally understood as the application of some sequence of simple transformations to an input. The sequence of transformations is called an algorithm. It is intuitively obvious that some algorithms are better than others. One quantifiable measure of quality is the complexity of the algorithm: how long it takes to execute, as a function of the size of the input. For a comprehensive introduction to complexity theory, [AB09] may be consulted. The Complexity Zoo [APKG] is a useful online reference for quick definitions and relationships between complexity classes.

The Turing machine was defined in an effort to abstract away the details of a computation. A Turing machine is a model for a computer: it has a finite set of internal states, a tape that stores data (a string of symbols from some alphabet), and a read/write head that can read information from the tape, and write to it. It also has an instruction set. Depending on the state of the machine and the symbol being read, the Turing machine can change the symbol beneath the head, and move the head along the tape. Given an input to the Turing machine (represented by some string of symbols on the tape), the operation of a Turing machine corresponds to the execution of an algorithm on the input. The output of the Turing machine is the contents of the tape when the machine halts. The number of steps it takes to halt is then a measure of the time complexity of
the algorithm.

How powerful a model is the Turing machine? Does it truly capture computational power? The Church-Turing thesis says that any ‘reasonable’ model of computation, regardless of the actual physical realization can be simulated on a Turing machine. The extended Church-Turing thesis goes one step further and says that this simulation can be done efficiently. The advent of quantum computing has thrown this stronger version in doubt: it seems as though quantum computers will not be efficiently simulatable on Turing machines.

We usually represent data in binary. A language is some set of binary strings, \( L \in \{0, 1\}^* \). Some of the simplest computational problems to state are decision problems: given a string \( x \), does \( x \) belong to a language \( L \) or not? For example, the language could be the set of all prime numbers. Then the decision problem is deciding the primality of an input integer.

Given an input \( x \), the running time of a Turing machine on that input will be a function of the length of the input \(|x|\). We say that an algorithm is efficient if the corresponding Turing machine halts in a time that is at most some polynomial in the size of the input. The set of all languages that are efficiently decidable (have an efficient algorithm) comprise the complexity class \( P \).

**Definition 1** (\( P \)). A language \( L \) is in \( P \) if there exists a polynomial-time Turing Machine \( M \) such that for every \( x \in \{0, 1\}^* \), \( x \in L \iff M(x) = 1 \).

There are many problems for which we do not have an efficient algorithm, but can still recognize (or have a computer recognize) a correct solution when we see one. The complexity class \( NP \) helps formalize this idea: a Turing machine acts as a verifier that can be given a ‘proof’ or ‘witness’ string as input.

**Definition 2** (\( NP \)). A language \( L \) is in \( NP \) if there exists a polynomial \( p : \mathbb{N} \rightarrow \mathbb{N} \) and a polynomial-time Turing Machine \( M \) such that for every \( x \in \{0, 1\}^* \),
\[
x \in L \iff \exists u \in \{0, 1\}^{p(|x|)} \text{s.t.} M(x, u) = 1
\]

Any language \( L \) in \( P \) also belongs in \( NP \): we may simply use the Turing Machine \( M \) that decides it in polynomial time, disregarding any witness string. Therefore, we may write \( P \subseteq NP \). Whether this containment is strict is one of the biggest open problems in mathematics and computer science.

Sometimes, it is possible to ‘reduce’ one language \( L \) to another language \( L' \). A reduction is simply the application of a function \( f \) (for our purposes, this function will always be computable
in polynomial time) to the strings of \( L \) to produce strings of some other language \( L' \), such that a string \( x \) belongs to \( L \) if and only if \( f(x) \) belongs to \( L' \). The upshot of the existence of such a reduction is that, given an algorithm that decides \( L' \) in polynomial time, we could also decide \( L \) in polynomial time.

An NP-hard problem is one to which every problem in NP can be reduced. It is therefore at least as hard as every problem in NP – an efficient algorithm for an NP-hard problem can be used to efficiently solve any problem in NP. NP-hard problems that are also in NP are called NP-complete. These are the ‘hardest’ problems in NP. In 1971, Cook and Levin independently showed that every problem in NP can indeed be reduced to a single problem, the Satisfiability problem.

A Boolean formula in Conjunctive Normal Form (CNF) over \( n \) variables \( x_1, x_2, \ldots, x_n \) is an AND of ORs of variables: \((y_{11} \lor y_{12} \lor \ldots) \land (y_{21} \lor y_{22} \lor \ldots) \land \ldots \land (y_{n1} \lor y_{n2} \lor \ldots)\), where each \( y_{ij} \) is either a variable \( x_k \) or its negation \( \overline{x_k} \). A formula is said to be satisfiable if there exists an assignment to the variables \( \{x_i\} \) under which the formula evaluates to 1. An instance of the Satisfiability (SAT) problem is simply one such Boolean formula. More precisely, the set of all satisfiable CNF formulae constitutes the language SAT. If we restrict the formula to have at most 3 variables per clause, we have the 3SAT problem.

The Cook-Levin theorem states that SAT and 3SAT are NP-complete. After this was shown, many natural problems that had eluded attempts at an efficient solution were shown to be NP-complete. Some famous examples are the Traveling Salesman Problem, Independent Set, 3Coloring, and Integer Programming. SAT will be of particular relevance to us throughout this dissertation. Note that 2SAT has an efficient algorithm, and is thus in P.

We mentioned earlier that the complexity class P defines an efficient computation. However, this is often thought of as an overly restrictive definition of tractability: with the additional resource of randomness, we may be able to efficiently solve a larger class of problems. We may also allow our randomized algorithms to give us the wrong answer with a small probability – this is not a problem because, by repeatedly running the algorithm, we can reduce the error probability to as small a value as desired. A probabilistic Turing machine (one that has access to random bits that it can use in its computation) is a model for a randomized algorithm. This more general notion of tractability is captured by the complexity class BPP:
Definition 3 (BPP). A language \( L \) is in BPP if it has a polynomial-time randomized algorithm that accepts with probability such that:

- if the input to the algorithm is in \( L \), the algorithm accepts with probability at least \( \frac{2}{3} \).
- if the input is not in \( L \), the algorithm accepts with probability at most \( \frac{1}{3} \).

We note that \( P \subseteq BPP \)

2.2 Quantum Computation

We have so far looked at a classical model of computation. What happens if we allow our computers to behave quantumly? Does this gain us computational power? How do the concepts of efficiency and intractability generalize to the quantum world?

Classical computers (which we have modeled by Turing machines) store and manipulate data in the form of bits. A bit can be thought of as a (classical) two-level system. What happens if we allow our systems to be quantum? In quantum mechanics, a \( k \)-level quantum system is a vector that can take values in a \( k \)-dimensional complex vector space. Therefore, a state of a two-level quantum system can be represented as a pair of complex numbers \((\alpha, \beta)\) – this is a vector in \( \mathbb{C}^2 \). Representing the basis vectors of this two-dimensional Hilbert space by \(|0\rangle \) and \(|1\rangle \), we can write the state as \(|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \). We also require that state vectors be of unit norm, in this case \(|\alpha|^2 + |\beta|^2 = 1 \).

In quantum computing, we term these two-level systems ‘qubits’. The combined Hilbert space of two qubits \( a \) and \( b \) is a tensor product of the two individual Hilbert spaces \( \mathbb{C}_a^2 \) and \( \mathbb{C}_b^2 \): the state of the two-qubit system is in general a vector \( c_{00} |0\rangle_a |0\rangle_b + c_{01} |0\rangle_a |1\rangle_b + c_{10} |1\rangle_a |0\rangle_b + c_{11} |1\rangle_a |1\rangle_b \) (for normalization, we have \(|c_{00}|^2 + |c_{01}|^2 + |c_{10}|^2 + |c_{11}|^2 = 1 \)). This is easily extended to an \( n \)-qubit system – the state vector lives in a \( 2^n \)-dimensional vector space. Our quantum computer will represent its data as qubits – the state of the computer will be a vector in the combined Hilbert space of the qubits.

The axioms of quantum mechanics dictate that a state vector can only evolve via a unitary transformation, i.e., a state \(|\psi\rangle\) can only evolve to some \(|\psi'\rangle = U |\psi\rangle\), where \( U^\dagger U = I \). Therefore, any operation that a quantum computer can perform on its qubits must be a unitary transformation. However, every unitary transformation can be feasibly implemented. For our quantum computing
model to be reasonable, we restrict ourselves to ‘local’ transformations: unitary matrices that act on only a constant number of qubits. This is not yet a reasonable model, however. We must further restrict our local unitary operations to some finite set. We call these local unitaries gates, in analogy with the components of classical circuits. Though we can compose these gates to achieve a global unitary transformation, we cannot exactly execute every possible global unitary. However, there exist ‘universal’ gate sets which can approximate any global unitary.

A quantum circuit is essentially a collection of gates (from some finite set) applied to a register of qubits on some sequence. After all the gates are applied, measuring a designated output qubit yields either 0 or 1, which we interpret as the answer returned by the computation. One measure of computational complexity of a problem is the size (number of gates) of a circuit that solves the problem. In particular, we will be interested in polynomial-size circuits.

In order to obtain a reasonable notion of efficiency, it is not enough that we restrict ourselves to polynomial-size circuits: the circuits can not be chosen independently of each other. We need to have a way of actually constructing these circuits efficiently, i.e., there should be a classical algorithm that, given an input size $n$, outputs the description of a circuit $C_n$ in $\text{poly}(n)$ time.

While the quantum Turing machine has been defined to help formalize notions of quantum complexity, we find quantum circuits a much easier model to think about. If we restrict ourselves to uniformly generated families of circuits, we find that an efficient computation on a quantum Turing machine is equivalent to a polynomial-sized quantum circuit. We shall henceforth make reference only to quantum circuits while defining quantum algorithms or complexity classes.

The complexity class that captures the notion of efficient computation in the quantum world is BQP. This is the class of problems that are efficiently decideable with a small error probability on a quantum computer.

**Definition 4 (BQP).** A language $L$ is in BQP if there exists a uniform family of quantum circuits $\{C_n\}$ such that, for every input string $x$ of length $n$:

- if $x \in L$, then $C_n$ acting on input $|x\rangle$ accepts with probability at least $2/3$.
- if $x \notin L$, then $C_n$ acting on input $|x\rangle$ accepts with probability at most $1/3$.

BQP generalizes, and therefore contains, BPP.
The quantum complexity class QMA (short for Quantum Merlin Arthur) is the quantum analogue of NP. Here, the verifier is a quantum circuit, and the witness or proof is a quantum state on polynomially-many qubits. We now define the class QMA:

**Definition 5 (QMA).** A language \( L \in \text{QMA} \) if there exists a quantum polynomial time verifier \( V \) and a polynomial \( p \) such that:

- \( \forall x \in L \exists |\xi\rangle \in B^{\otimes p(|x|)}, Pr(V(|x|\langle\xi|)) = 1 \geq \frac{2}{3} \)
- \( \forall x \notin L \forall y, |y| = \text{poly}(|x|), Pr(V(|x|\langle\alpha|)) = 1 \leq \frac{1}{3} \)

Similar to NP, we have notions of QMA-hardness and -completeness. The notion of reduction is the same: a polynomial-time classical algorithm is used to map instances of one language to instances of another. The 5-local Hamiltonian problem was the first problem to be shown to be QMA-complete. We will define this problem shortly. For a comprehensive list of QMA-complete problems, see [Boo12].

### 2.3 Generalizations of SAT

The Local Hamiltonian problem can be viewed as a generalization of SAT. Let us see how. To keep things concrete, let us consider 3-SAT. Each clause in an instance of 3-SAT can be written as a projection matrix: a clause can be thought of as constraining the space of 3-bit strings by penalizing one of the strings. Consider a clause \( C_i = (\neg x_1 \lor x_2 \lor \neg x_3) \). The only unsatisfying assignment for this clause is 101. \( C_i \) can be written as a projection \( |101\rangle \langle 101| \), acting on an eight-dimensional vector space (corresponding to the combined Hilbert space of three qubits). Note that the matrix corresponding to this projection operator is Hermitian. The eigenvectors of this matrix are precisely the basis vectors of this three-qubit space. If we denote the sum of these projectors by \( H \), the 3-SAT problem can be rephrased as deciding whether the least eigenvalue of \( H \) is 0 (corresponding to the CNF being satisfiable) or at least 1 (corresponding to unsatisfiability). If we think of \( H \) as representing the Hamiltonian of some \( n \)-qubit quantum system, the ground state of \( H \) will be a vector corresponding to some basis string.

Now what happens if we allow the projectors to be non-diagonal? Clearly, the problem no longer makes sense in the classical setting: the eigenvectors are not restricted to be the basis vectors. However, in a quantum setting, we can still ask the following question: Does there exist a
state of \( n \) qubits that is simultaneously a zero eigenvector of each of the projectors? We thus have the quantum \( k \)-SAT problem:

**Definition 6 (Quantum \( k \)-SAT).** Given a set of \( k \)-qubit projectors \( \{ H_i \} \) acting on \( n \) qubits, and a real number \( \epsilon = \Omega(1/poly(n)) \), decide if there exists some state \( |\psi\rangle \) such that \( H_i |\psi\rangle = 0 \) for all \( i \), or if \( \langle \psi | \sum_i H_i |\psi\rangle > \epsilon \) for all states \( |\psi\rangle \).

We can generalize SAT still further by relaxing the constraint that the \( \{ H_i \} \) be projectors. If we merely require each \( H_i \) to be Hermitian, the question of estimating the ground state energy of \( H = \sum_i H_i \) is called the Local Hamiltonian problem.

**Definition 7 (The \( d \)-state \( k \)-Local Hamiltonian Problem).** We are given a Hamiltonian \( H = H_1 + H_2 + \ldots + H_s \) on \( n \) \( d \)-state qudits, with the matrix elements of each \( H_i \) specified by \( poly(n) \) bits. \( H \) is \( k \)-local: each \( H_i \) acts non-trivially on only \( k \) of the \( n \) qudits. We are also given two constants \( a, b \in \mathbb{R} \) such that \( b - a \geq 1/poly(n) \), with the promise that the smallest eigenvalue of \( H \), \( \lambda(H) \), is either at most \( a \) or greater than \( b \). We must decide if \( \lambda(H) \leq a \) or \( \lambda(H) > b \).

The \( k \)-SAT problem is thus a very specific case of the \( k \)-Local Hamiltonian problem. The 5-Locally Hamiltonian problem was the first to be shown to be QMA-complete, in [KSV02]. It is also a very natural complete problem, given that it is a generalization of SAT. Moreover, physicists have worked on similar problems, developing a number of heuristic tools for approximating ground states and ground state energies. However, the Hamiltonian constructed in [KSV02] does not have any constrain the spatial arrangement of the qubits, making it unrealistic. In physical (e.g. spin) systems, the Hamiltonians are often spatially local: the interacting systems (qubits or qudits) may be arranged on a grid, or the interactions are (at least approximately) short-ranged (e.g. nearest-neighbor). Simulation of local Hamiltonians on one- or two-dimensional grids is an important problem in physics, and it is natural to try to understand the complexity in the different cases obtained by changing the locality and the dimensionality of the qudits. Since it is also much easier to realize and manipulate lower-dimensional qudits in the lab, these cases are particularly important.

We shall now review a number of results about the complexity of these various specializations.
2.3.1 The Ising model

We shall begin our review by looking at a few results about the computational complexity of the Ising model [Isi25]. The Ising model is a simple and fundamental model in statistical physics, and can be thought of as a classical constraint satisfaction problem consisting of 2-local (nearest neighbor) and 1-local terms. The (classical) variables in this model are spins $\{S_i\}_{i=1}^{n}$ taking values $\pm 1$, located at lattice sites. The energy of the system is given by the Hamiltonian function

$$H = -\sum_{\langle i,j \rangle} J_{ij} S_i S_j - \sum_{i=1}^{n} B_i S_i,$$  \hspace{1cm} (2.1)

where $\langle i,j \rangle$ denotes a nearest neighbor pair.

Ising, in his doctoral thesis [Isi25], showed how the 1D Ising model may be analytically solved. However, the 1D model was insufficiently complex to capture the phenomenon it was intended to model – ferromagnetism. The 2D model in the absence of magnetism (all the $B_i$ are zero) was also tractable. However, in the presence of an external magnetic field, the 2D model was shown to be NP-complete, as was the 3D model [Bar82, Ist00].

2.3.2 QMA-complete Local Hamiltonians

Since the 5-local Hamiltonian problem was shown to be QMA-complete, there have been improvements on this result. The locality was brought down to 3 [KR03] and then to 2 [KKR06]. The 2-local problem remains QMA-complete when the Hamiltonians are restricted to be nearest-neighbor interactions on a 2D grid [OT08]. The 1D case was not expected to be so hard: its classical counterpart, the 1D constraint satisfaction problem, has efficient algorithms. Moreover, there are good heuristic methods that are effective on many instances of the problem. Therefore, it was a somewhat surprising result when [AGIK09] showed a hardness proof for the Local Hamiltonian problem on a chain (with nearest-neighbor interactions) of 13-state particles. The size of the particle was later brought down to 11 by Nagaj [Nag08].

In this dissertation, we combine ideas from [AGIK09] and [Nag10] to show that the Local Hamiltonian problem with nearest-neighbor interactions between 8-dimensional particles on a line is QMA-complete.
2.3.3 Projection Hamiltonians, or Quantum \(k\)-SAT

We may expect that Quantum \(k\)-SAT may be computationally easier than the more general \(k\)-Local Hamiltonian problem. Bravyi shows in [Bra06] that Quantum-\(k\)-SAT is QMA\(_1\)-complete (QMA\(_1\) is a ‘one-sided’ version of QMA: the positive instances are accepted by the verifier with probability one) for \(k \geq 4\) or more. It is unknown if Quantum-3-SAT is QMA\(_1\)-complete.

Just as 3SAT is NP-complete, while 2-SAT is amenable to a polynomial-time algorithm, we may ask if something similar happens in the quantum case. Does Quantum-\(k\)-SAT have a polynomial-time algorithm for a small enough \(k\)? In [Bra06], Bravyi answers this question in the affirmative by giving a polynomial-type algorithm for Quantum-2-SAT, placing it in P.

The Quantum-\(k\)-SAT problems we have considered so far are all on qubits. Does increasing the dimensionality of the particles change anything? In particular, how hard is Quantum-2-SAT on qutrits or other higher-dimensional particles? It is easy to show that this is NP-hard. However, we could restrict ourselves to cases of Quantum-2-SAT where there are a small number of qutrits (constant or log-many) among the qubits. In this dissertation, we investigate a version of Quantum-2-SAT where a constant number of higher-dimensional qudits interact with qubits.

2.3.4 Commuting Local Hamiltonians

Another restricted version of the Local Hamiltonian problem is the commuting Local Hamiltonian problem: the local terms in the Hamiltonian are mutually commuting. Such a system can be thought of as lying somewhere between classical and ‘fully’ quantum, and we may expect it to lie closer in complexity to SAT. It is certainly NP-hard, since it generalizes the Ising model. However, the commuting 2-local Hamiltonian problem was also shown to lie in NP [BV05], for any qudit dimensionality. This result was extended by [AE11], who showed that the 3-local Hamiltonian problem on qubits (and a restricted version of the 3-local problem on qutrits) lay in NP.

If the commuting terms are each a tensor product of Pauli matrices, this problem lies in P (for any constant locality) [YB12].
2.3.5 Other special cases

There are other, physically meaningful, restrictions that can be placed on the terms in a Hamiltonian. ‘Stoquastic’ Hamiltonians occur commonly in nature, and the complexity of computing their ground energy has been studied [BDOT08, BT09]. Another restriction that applies to many physical systems is a spatial symmetry: the Hamiltonian terms are translationally invariant. Such systems have been studied in [GI09, Kay07, VC08].

2.3.6 Approximation Algorithms and Heuristics

We have seen that only a few special cases of the Local Hamiltonian problem can be efficiently solved. We have also seen that the Local Hamiltonian problem is in general QMA-complete, meaning that no efficient algorithm is likely to exist. However, we do not always need an exact solution to every instance of a problem. Often, it suffices to be close enough to the true answer. For example, we do not have an efficient algorithm that solves every instance of the Travelling Salesman Problem, but we do have approximation algorithms that give us an answer that is ‘close’ (up to some specified error) to the optimum. Similarly, we have approximation algorithms for the general $k$-local Hamiltonian problem [GK12]. Classical approximation algorithms for the classical and quantum Ising spin glass problems on planar graphs (which are NP-hard and QMA-hard respectively) are given in [BBT09]. An efficient algorithm for approximating the ground state of a certain class of 1D Hamiltonian systems is given in [AAI10]. Other classes of Hamiltonians that can be efficiently simulated are characterized in [Vid04, Vid08].

Physicists have been interested in computing the solutions for many cases of the Local Hamiltonian problem for a long time before computer scientists began formally studying its complexity, and have developed a number of heuristic approaches to approximating the properties of such systems. It is important to distinguish between heuristics and algorithms. An algorithm produces the right answer on every instance (an approximation algorithm always produces a good approximation). A heuristic may perform well on a large fraction of inputs, but is not guaranteed to always do well, and can only be empirically validated. The heuristics for the Local Hamiltonian problem draw from a varied set of tools, such as the density matrix renormalization group (DMRG) [Sch05], which has proven particularly useful in approximating the ground states of 1D systems. See [CV09] for a
review of these techniques.
Chapter 3

8-state particles on a line

In this chapter, we consider the 2-Local Hamiltonian problem on a line of 8-dimensional qudits with nearest neighbor interactions, and show a reduction from an arbitrary QMA language to this problem, thereby proving that the 1D 8-state LH problem is QMA-complete.

3.1 Introduction

The Local Hamiltonian problem – estimating the ground state energy of a local Hamiltonian – is a natural problem in physics, and belongs to the complexity class QMA. QMA is the quantum analogue of NP. Languages in QMA have a quantum verifier: a polynomial-time quantum algorithm that takes (poly-sized) quantum states as witnesses.

The input to the $k$-Local Hamiltonian problem is a set of Hermitian matrices $\{H_i\}$, each $H_i$ acting on a set of $k$ qubits (out of a total of $n$), and the problem is to estimate the lowest eigenvalue of the sum $H = \sum_{i=1}^{M} H_i$. Note that even though each $H_i$ acts non-trivially on a constant number $k$ of qubits and is constant-dimensional, $H$ itself acts on the whole space of $n$ qubits and is therefore exponential in size. The Local Hamiltonian problem can be thought of as a generalization of SAT [AN02]. In particular, MAX2SAT is a special case of the 2-Local Hamiltonian problem. Therefore, 2-Local Hamiltonian is NP-hard.

The 5-Local Hamiltonian problem was the first to be shown to be QMA-complete, in [KSV02]. It is also a very natural complete problem, given that it is a generalization of SAT. Moreover, physicists have worked on similar problems, developing a number of heuristic tools for approximating ground
states and ground state energies. However, the Hamiltonian constructed in [KSV02] does not have any constraint the spatial arrangement of the qubits, making it unrealistic. In physical (e.g. spin) systems, the Hamiltonians are often spatially local: the interacting systems (qubits or qudits) may be arranged on a grid, or the interactions are (at least approximately) short-ranged (e.g. nearest-neighbor). Simulation of local Hamiltonians on one- or two-dimensional grids is an important problem in physics, and it is natural to try to understand the complexity in the different cases obtained by changing the locality and the dimensionality of the qudits. Since it is also much easier to realize and manipulate lower-dimensional qudits in the lab, these cases are particularly important.

There have been improvements on this result and its unfrustrated variant Quantum $k$-SAT [Bra06, ER08, Nag10]. The locality was brought down to 3 [KR03] and then to 2 [KKR06]. The 2-local problem remains QMA-complete when the Hamiltonians are restricted to be nearest-neighbor interactions on a 2D grid [OT08]. The 1D case was not expected to be so hard: its classical counterpart, the 1D constraint satisfaction problem, has efficient algorithms. Moreover, there are good heuristic methods that are effective on many instances of the problem. Therefore, it was a somewhat surprising result when [AGIK09] showed a hardness proof for the Local Hamiltonian problem on a chain (with nearest-neighbor interactions) of 13-state qudits. In this paper, we bring the number of states down from 13 to 8.

The hardness of the 1D problem (with nearest-neighbor interactions only) for the cases with 2-7 state qudits remains an interesting open question. It is not clear if the QMA-completeness result will continue to hold as we further decrease the dimensionality of the qudits down towards 2. It may happen that below a particular dimensionality, we could find that the problem has an efficient quantum or classical algorithm, e.g. if the ground state entanglement could be shown to be low. Recently, an interesting qutrit chain with a unique unfrustrated ground state with lots of entanglement was analyzed in [BCM+12]. Finally, these QMA-completeness results also bear a close relationship to adiabatic quantum computing: the computation models in these results, and the Hamiltonians that check these computations, can be used to perform universal adiabatic

\footnote{The paper states hardness for $d = 12$. However, there are two illegal configurations that are not penalized: $\otimes \otimes \otimes \otimes$ and $\otimes \otimes \otimes \otimes$ turn into each other under the action of the Hamiltonian. The superposition of these two configurations forms a zero energy state of the Hamiltonian, which means that the Hamiltonian no longer has the promised $\frac{1}{\text{poly}}$ gap. This can be fixed by adding a 13th state, as discussed in footnote 4 in [AGIK09].}
quantum computing. It will be interesting to see if restricted local Hamiltonian systems (e.g., low-dimensional qudits on a line) that most likely do not encode a QMA-complete problem can still be used to perform universal adiabatic QC.

To show QMA completeness for our version of Local Hamiltonian, we reduce an arbitrary QMA language $L$ to a Local Hamiltonian in 1D with $d = 8$ particles, outputting a Hamiltonian that either has a ground state energy below some value $a$, or whether this energy is at least $1/poly$ larger than $a$. We base our proof on three ideas.

First, we use Kitaev’s Hamiltonian $[KSV02]$: a Hamiltonian that has as its ground state a *history state* of the verification circuit $V_x$ for the language $L$. A history state is a state of the form $\sum_t |\phi_t\rangle|t\rangle$, where $|\phi_t\rangle$ is the state after applying the first $t$ gates of $V_x$ to $|\phi_0\rangle$. Kitaev’s Hamiltonian induces forward and backward transitions between consecutive time-steps, i.e., $|\phi_t\rangle|t\rangle \leftrightarrow |\phi_{t+1}\rangle|t+1\rangle$. In addition, the Hamiltonian serves to ensure that no illegal (i.e., not corresponding to an encoding of the time-step) states occur in the clock register, that the input to the circuit is correct, and that the computation ultimately accepts.

Second, the encoding of a computation in the ground state of a nearest-neighbor 1D Hamiltonian is based on the construction of $[AGIK09]$. The $n$ computational qubits are encoded in subspaces of $n$ of the qudits (of which there are polynomially many) on the line. The line is divided into blocks, and in each block a set of nearest-neighbor gates is performed on the encoded qubits before the qubits are transferred to the next block where the next set of gates can be performed. The gate applications and the qubit transfers occur via two-local (nearest-neighbor) operations. The construction in $[AGIK09]$ uses a 2-dimensional “gate” subspace of the qudits to mark the position along the line where a gate is being performed. The qudits storing the qubits on which gates have already been performed are indicated by a two-dimensional space $[,]$, and the ones on which a gate is yet to be performed are labeled $[\ ]$. There are also one-dimensional states $\bigcirc$ and $\bigcirc$: the former marks the transition between the gate-performing steps and the qubit-transferring steps of the computation, and the latter shifts the qubits to the right. A two-dimensional state $\bigtriangledown$ serves to move the active spot back to the right after $\bigcirc$ has moved the qubits over one site.

Third, our main contribution is reducing the dimensionality of the qudits to 8. This “leaner” qudit construction comes at a price – allowing the forward/backward transitions in our Hamiltonian

\[2\text{The subscript } x \text{ in the verifier circuit } V_x \text{ stands for the instance } x \text{ of the problem.}\]
to be non-unique, possibly resulting in “illegal” configurations of the qudit chain. However, we can work around this problem and suppress those by adding penalty terms. Raising the energy of states away from the allowed subspace allows us to use the projection lemma from [KKR06], showing that even despite the illegal transitions, the ground state must have a substantial overlap with the legal subspace. Restricted to that subspace, the illegal transitions in the Hamiltonian do not contribute to the expectation value of the energy for a correct history state. Therefore, the history state of a computation that accepts with high probability can be close to the ground state of the entire Hamiltonian, and results in a low ground-state energy. Of course, we also need to show a lower bound on the ground state energy for Hamiltonians corresponding to quantum circuits without easily-accepted witnesses.

In more detail, our dimension reduction comes from getting rid of the distinction between the two two-dimensional qudit states – the $\mathbb{L}$ (done: qubits in a block that have already participated in gate applications) and $\mathbb{R}$ (ready: qubits that are yet to have a gate applied to them) qubit types – using instead just one type of qubit $\circ$ combined with a 1-dimensional state $\otimes$, using parity of the qubit position to distinguish between “done”/“ready”. We use the mapping $\mathbb{L} \rightarrow |\circ\circ|\otimes$ and $\mathbb{R} \rightarrow |\circ\circ\rangle$, doubling the number of particles on the line. Furthermore, we get rid of the $\mathbb{D}$ qubit type – we instead use the boundary between “done” and “ready” sequences of qubits as the active spot. We also will not need the $\mathbb{D}$ state (used in [AGIK09]) anymore.

### 3.2 Background

Let us recall the definition of the Local Hamiltonian problem:

**Definition 8 (The $d$-state $k$-Local Hamiltonian Problem).** We are given a Hamiltonian $H = H_1 + H_2 + \ldots + H_s$ on $n$ $d$-state qudits, with the matrix elements of each $H_i$ specified by poly($n$) bits. $H$ is $k$-local: each $H_i$ acts non-trivially on only $k$ of the $n$ qudits. We are also given two constants $a, b \in \mathbb{R}$ such that $b - a \geq 1$/poly($n$), with the promise that the smallest eigenvalue of $H$, $\lambda(H)$, is either at most $a$ or greater than $b$. We must decide if $\lambda(H) \leq a$ or $\lambda(H) > b$.

As mentioned previously and shown by Kitaev [KSV02], this problem lies in (and is complete for) the class QMA.
Definition 9 (QMA). A language $L$ is in the class QMA iff for each instance $x$ there exists a uniform polynomial-size quantum circuit $V_x$ such that

- if $x \in L$, $\exists |\xi\rangle$, a polynomial-size quantum state (a witness) such that $\Pr(\text{accept}(V_x, |\xi\rangle)) \geq \frac{2}{3}$,
- if $x \notin L$, $\forall |\xi\rangle \Pr(\text{accept}(V_x, |\xi\rangle)) \leq \frac{1}{3}$.

Previous proofs for QMA-completeness rely on a special state encoding a computation (a history state) for showing QMA-hardness of Local Hamiltonian. A circuit is transformed into an appropriate Hamiltonian such that a history state is a zero-eigenvector when there is a witness to make the circuit accept.

Definition 10 (History state). Let $V = U_K \cdots U_2 U_1$ be a circuit of $K$ gates on $n$ qubits. Consider a Hilbert space with $K + 1$ orthogonal subspaces $\{S_t\}_{t=0}^K$, each with basis $\{|j_t\rangle\}_{j=0}^{2^n-1}$ of dimension $2^n$. We define the history state corresponding to the action of $V$ on an initial $n$-qubit state $|\varphi\rangle$ as a superposition over states coming from orthogonal spaces:

$$|\eta^\varphi\rangle = \frac{1}{\sqrt{K+1}} \sum_{t=0}^{K} |\gamma^\varphi_t\rangle$$

where $|\gamma^\varphi_t\rangle = \sum_{j=0}^{2^n-1} |j_t\rangle \langle j| U_t \cdots U_2 U_1 |\varphi\rangle$ is a vector in the subspace $S_t$.

Note that the Hilbert space as a whole can be bigger than the union of $S_t$'s, and we can write it as an orthogonal direct sum of subspaces $\left( \bigoplus_{t=0}^{K} S_t \right) \oplus C_{\text{rest}}$, with the rest of the Hilbert space denoted $C_{\text{rest}}$.

A propagation Hamiltonian can be defined to ensure that a low-energy candidate state has the form (3.1), when the state evolution satisfies a certain orthogonality condition. Note that for any initial $n$-qubit state $|\varphi\rangle$ and any $t \in \{0, \ldots, K\}$, we have $|\gamma^\varphi_t\rangle \in S_t$. The propagation Hamiltonian associated with the circuit $V$ is $H_{\text{prop}} := \sum_{t=0}^{K-1} H_t$ where

$$H_t := \sum_{j=0}^{2^n-1} \left( |j_t\rangle \langle j_t| + |j_{t+1}\rangle \langle j_{t+1}| - U_{t+1}' |j_{t+1}\rangle \langle j_t| - |j_t\rangle \langle j_{t+1}| U_{t+1}'^\dagger \right),$$

with $U_{t+1}' = \sum_{k,j=0}^{2^n-1} |k_{t+1}\rangle \langle k| U_{t+1} |j\rangle \langle j_{t+1}|$ acting as the unitary $U_{t+1}$ on the subspace $S_{t+1}$. 
Observe that the action of $H_t$ on $|\gamma_t^\varphi\rangle$ and $|\gamma_{t+1}^\varphi\rangle$ is (not summing over $t$)

\begin{align*}
H_t |\gamma_t^\varphi\rangle &= |\gamma_t^\varphi\rangle - |\gamma_{t+1}^\varphi\rangle, \\
H_t |\gamma_{t+1}^\varphi\rangle &= |\gamma_{t+1}^\varphi\rangle - |\gamma_t^\varphi\rangle,
\end{align*}

since $|\gamma_{t+1}^\varphi\rangle = U_{t+1}' \sum_{j=0}^{2^n-1} |j_{t+1}\rangle \langle j_t|\gamma_t^\varphi\rangle$. It is then straightforward to verify that $|\gamma_t^\varphi\rangle + |\gamma_{t+1}^\varphi\rangle$ is a zero-energy eigenvector of $H_t$. A history state is then also a zero eigenvector of each $H_t$, and so a zero eigenvector of $H_{\text{prop}}$. The propagation Hamiltonian thus serves to “check” the progress of the computation, by giving an energy penalty to all non-history states. For a specific construction of a QMA-complete $k$-Local Hamiltonian problem, it will have to be shown that $H_{\text{prop}}$ can be built from operators that obey the chosen locality restrictions.

Let $V_x$ be a verifying circuit for an instance $x \notin L \in QMA$, taking as input $n - m$ ancilla qubits in the state $|0\rangle$ and an $m$-qubit state $|\xi\rangle$, and it has squared amplitude $2/3$ on some designated output qubit if $x \in L$, and less than $1/3$ otherwise. Kitaev’s proof used a history state of the following form. The unitaries $U_i$ are the ones from the original verifier circuit $V_x = U_K \ldots U_1$ and are 2-local. An extra unary clock register is used to build the structure of orthogonal subspaces $S_t$, requiring a 2-local clock checking Hamiltonian $H_{\text{clock}}$ in the Hamiltonian for distinguishing the subspaces $S_t$ from $C_{\text{rest}}$ spanned by states with illegal clock configurations. The history state for verifying a valid witness $|\xi\rangle$ using $V_x$ is

$$
|\eta\rangle = \frac{1}{\sqrt{K + 1}} \sum_{t=0}^K \left( U_t \ldots U_1 \left( |0\rangle^{n-m} \otimes |\xi\rangle \right) \right) \otimes |t\rangle_{\text{clock}}.
$$

This history-state structure for low-energy state candidates is enforced by $H_{\text{prop}}$ imposing energy penalties for deviating from the indicated form. With the unary clock construction, the required locality of the terms in $H_{\text{prop}}$ is 5. In addition, Kitaev adds two more Hamiltonian terms: $H_{\text{in}}$ penalizing states with improperly initialized ancillae (not of the form $|\gamma_0\rangle = |0\rangle^{n-m} \otimes |\xi\rangle$), and $H_{\text{out}}$ verifying whether the computation accepts. This turns out to be enough to ensure that $x \notin L$ instances of the 5-Local Hamiltonian have no low-energy eigenvector.
3.3 Encoding a computation in a sequence of orthogonal states of a line of 8-dimensional qudits.

Our goal is to encode a quantum verifier circuit $V_x$ into a 2-Local Hamiltonian instance with nearest-neighbor interactions on a line of qudits, satisfying certain properties. In this section we do the first step, transforming $V_x$ into a modified circuit $\tilde{V}_x$ that does the same computation as $V_x$, but instead of on $n$ qubits, it acts on a line of $\text{poly}(n)$ qudits of dimension $d = 8$. All gates in $\tilde{V}_x$ are nearest-neighbor on this line, and the states occurring during the computation are pairwise orthogonal. This is the condition given in Section 3.2. Finding the circuit $\tilde{V}_x$ with these properties allows us to define a Hamiltonian such that, in the case that there exists a witness on which the circuit $\tilde{V}_x$ accepts with high probability, the history state of the computation on the witness is a low-energy state of the Hamiltonian. Otherwise, we will be able to lower bound the ground state energy of this Hamiltonian.

Assume $V_x$ works on a space of $n$ qubits. Choose a way to arrange the qubits on a line. The original circuit can be transformed to a circuit $V'_x$ consisting of $R$ rounds of gates, where each round is composed of $n - 1$ nearest-neighbor gates: the first gate in a round acts on qubits 1 and 2, the second on qubits 2 and 3, and so on. Any quantum circuit can be recast in this fashion, by inserting swap gates and identity gates, with a polynomial blowup increase in the number of gates.

We now convert the circuit $V'_x$ to a circuit $\tilde{V}_x$ acting on a line of 8-state qudits arranged in $R$ blocks of $2n$ particles each. The qudits are 8-dimensional, and we will utilize some of the 8 states as data-carriers (holding qubits the computation acts on). The rest will guarantee the orthogonality conditions and the proper progress of the computation. At any time during the computation, we want exactly $n$ of the qudits to be in the “data-holding” states, and we simply call them qubits. Initially, all of the $n$ qubits are located in the first block of particles. After each round of gates from $V'_x$ is carried out, the qubits are transferred to the next block of $2n$ particles where the next set of gates from $V'_x$ can be performed.

**Claim 11.** Given a QMA verifier circuit $V_x$ on $n$ qubits, an equivalent QMA verifier circuit $\tilde{V}_x$ can be efficiently computed such that $\tilde{V}_x$ operates on $2nR$ 8-state qudits on a line, only uses nearest neighbor gates, and such that the states occurring during the computation are pairwise orthogonal.
In the rest of this Section we describe the sequence of orthogonal states that appear in the computation on the qudit line. Later, in Section 3.4 we present the positive semidefinite 2-local Hamiltonian whose ground state is the uniform superposition over states from this desired sequence.

Let us choose the Hilbert space of each particle as an orthogonal direct sum: $\mathbb{C}_8 = \bigoplus \bigoplus \bigoplus \bigoplus \bigoplus \bigoplus \bigoplus \bigoplus \bigoplus$. The subspaces denoted $\bigcirc$, $\bigotimes$, $\bigotimes$, and $\bigotimes$ are 1-dimensional. Then we have 2-dimensional subspaces $\bigbox$ and $\bigtriangledown$, each designed to hold a state of a qubit (specified by two complex numbers $a_0$ and $a_1$, with $a_0, a_1 \in \mathbb{C}$ and $|a_0|^2 + |a_1|^2 = 1$). We label the basis vectors of these 2-dimensional subspaces $\bigbox^{(s)}$ and $\bigtriangledown^{(s)}$ with $s = 0, 1$. A qudit in the state $\sum_{s=0}^{1} a_s \bigbox^{(s)}$ or $\sum_{s=0}^{1} a_s \bigtriangledown^{(s)}$ is then said to have the qubit content $a_0 |0\rangle + a_1 |1\rangle$.

When we label a qudit by one of the symbols $\{\bigcirc, \bigotimes, \bigotimes, \bigotimes, \bigotimes, \bigotimes\}$, we mean that its state belongs to a particular subspace of $\mathbb{C}_8$. Such labeling of the whole chain defines a configuration. The Hilbert space of the qudit chain thus decomposes into orthogonal subspaces indexed by configurations. We can choose a basis for the Hilbert space of the entire system as a tensor product of $2nR$ (one for each site) of the basis vectors $\{\bigcirc\rangle, \bigotimes\rangle, \bigotimes\rangle, \bigotimes\rangle, \bigbox^{(0)}\rangle, \bigbox^{(1)}\rangle, \bigtriangledown^{(0)}\rangle, \bigtriangledown^{(1)}\rangle\}$. The state of the system is a vector in the span of the basis vectors.

Let us now construct a sequence of configurations, corresponding to the progression of a computation with the circuit $\tilde{V}_x$. We view the qudit chain as $R$ blocks of length $2n$ and mark their boundaries $\|$. To highlight the parity of the sites, we also draw $\|$ after every even, non-boundary site.

In the initial configuration, the first block holds qubits at odd-numbered sites, interspersed with $\bigotimes$s. The rest of the chain consists of $\bigcirc$s:

$$\| \bigcirc \bigotimes \bigotimes \cdots \bigotimes \bigcirc \bigotimes \bigotimes \bigcirc \bigotimes \bigcirc \cdots$$

(3.4) the first block of length $2n$

The qubit content of the $\bigtriangledown$ and $\bigbox$ sites carries the initial $n$-qubit input to the circuit $V'_x$ (the ancillae and the witness). Each step of the computation is a 2-local unitary operation applied to two adjacent particles, resulting in a change of configuration (building up an orthogonal sequence), and possibly a change in the state of the qubit content (doing the computation). Let us now write the rules for building up the circuit $\tilde{V}_x$.

We choose a list of transition rules (for configurations) and list them in Table 3.1 Each rule
1. ▶ □ ←→ $U_m(□|■)$ performs a two-qubit gate $U_m$ (location-dependent) on the qubit content of the two particles, while shifting the active site to the right.

2. (a) ▶ ◯ ←→ ◯ ▶ moves an “active” qubit ▶ to the right (not near a block boundary),
   (b) ▶ ◯ ←→ ◯ ▶ is applicable when a block boundary is to the left of it,
   (c) ▶ ◯ ←→ ◯ ▶ is applicable when a block boundary is in front of it.

3. (a) ◯ □ ◯ □ ←→ ◯ ◯ □ □ moves the leftmost qubit (not after a boundary),
   (b) □ □ □ □ ←→ □ □ □ □ moves a qubit □ to the right, only noting correct parity, regardless of the boundary location. We denote this using the symbol □.
   (c) □ □ □ □ ←→ □ □ □ □ moves the rightmost qubit (not before a boundary).
   (d) ◯ □ □ □ ←→ ◯ □ □ □ a special rule ensuring that if there is a single qubit in the chain, it can still move. Rule 3(d) does not actually apply to any legal configuration.

4. (a) ▶ □ □ ←→ □ □ ◯ creates a left-moving pusher ◯ at the front near a boundary □.
   (b) □ □ □ ←→ □ □ □ introduces ◯ when away from a block boundary.

5. (a) □ □ ◯ ←→ ◯ □ □ pushes ◯ left and a qubit to the right (not caring for the boundary).
   (b) □ □ ◯ ←→ □ □ ◯ does the same with ◯ and ◯, at locations with this parity.

6. (a) ◯ ◯ □ □ ←→ ◯ ◯ □ □ kills the pusher ◯ at the left end of the qubits at a boundary □, changing the last qubit to □, allowing the next round of gate applications to begin.
   (b) ◯ ◯ □ ←→ ◯ ◯ □ simply kills the pusher, when away from the boundary. ◯.

Table 3.1: The transition rules, which together with a carefully chosen initial state (3.4) define the 2-local gates of the circuit $\tilde{V}_x$. Note that some of these rules are 2-local, some 3-local and some even 4-local, which helps them identify their intended locations uniquely. However, the transformations themselves are only 2-local. See also Table 3.2 for an example of a progression of configurations and the unique applicability of these rules. We will later write a Hamiltonian $H_{\text{prop}}$ with only 2-local terms checking these transitions.
connects configurations that differ in two particular neighboring spots, and are connected by a 2-local unitary transformation. The sequence of these transformations (as applied sequentially to the initial configuration) defines the 2-local gates of the circuit $\tilde{V}_x$. This assignment of unitaries is unique by construction, as we choose the transition rules so that for any configuration arising from the initial one, there is always exactly one rule that possibly applies to it (see also Table 3.2 for a part of the sequence of configurations for $n = 3$). We ensure this uniqueness by rules involving up to 4 particles in the rules. However, in Section 3.4 we will write a Hamiltonian made from 2-local terms that checks these transitions.

Let us explain the logic behind the rules. Rule 1 applies the unitary from the modified, nearest-neighbor circuit $V'_x$. The rest of the rules ensure the orthogonalization and locality properties. Initially, the qubits are placed at the odd sites, separated by ⃝s. If we want them to interact, we have to move them together, which is what rules 2 and 3 do. The nearest-neighbor gates from $V'_x$ are then performed at the ▶ junctions using rule 1. The ⃝ label marks sites that the computation hasn’t reached yet, while the ⊗ sites will not be used again. The ⎕ (a pusher state) serves to move the qubits to the right.

The computation can be divided into $R$ “rounds”, each corresponding to the application of a “round” of gates from $V'_x$, and then moving the qubit block $2n$ positions to the right. Let us look at the two phases of a round of computation in detail, referring to Table 3.2 (a $n = 3$ qubit example).

The goal of the first phase of the computation is gate application. It involves rules 2 and 1, moving the ▶ qubit from the left end of the chain while applying the gates from a given “round”. When ▶ reaches the front end of the chain, rule 4 creates a the “pusher” state ⎕. After $2n - 1$ applications of rule 5, the pusher gets to the left end of the qubit sequence, where it disappears through rule 6. This first phase thus moves ▶ $n$ times, makes $n - 1$ gate applications, adds 1 pusher creation, $2n - 1$ pushes and 1 killing of ⎕, altogether making $4n$ steps.

The second phase (which is repeated $n - 1$ times) moves the qubits to the right until they are all within the next block. It takes $n$ applications of rule 8 to move all the qubits one step to the right. Then we create the pusher ⎕, move it to the left ($2n - 1$ steps) and kill it. Altogether, this takes $3n + 1$ steps. If we now are not at the boundary, the second phase repeats. If we are at a

---

3To “check” a transition means adding an energy penalty to terms that do not have the same amplitude for both of the states involved in the transition.
block boundary $\parallel$, the second phase concludes, and the “round” of computation concludes as well, as all the qubits have now moved $2n$ positions to the right. A new “round” of computation (with the particle $\blacksquare$ starting to move) starts according to rule $\$$. Summing it up, a whole “round” of computation consists of $4n + (n - 1)(3n + 1) = 3n^2 + 2n - 1$ steps. During each “round”, $n - 1$ gates from $V'_x$ are applied and the qubits are moved over to the next block of qudits. This happens for each of the first $R - 1$ blocks. In the last block, after the gates are applied, the computation comes to a halt in the state: $\parallel \otimes^{2n(R-1)} \parallel \otimes \square \| \cdot \cdot \cdot \otimes \square \| \otimes \blacksquare \|$. Also, without loss of generality, we take all the gates in the very first round to be identities. This allows us to verify that the ancilla qubits (laid out on the left of the qubit sequence) all start out in the correct state $\mid 0 \rangle$.

The entire computation with $(R - 1)$ regular rounds and a last round with $2n$ steps (until $\blacksquare$ reaches the right end) together take $K = (R - 1)(3n^2 + 2n - 1) + 2n$ steps, corresponding to $K + 1$ configurations of the qudits. Also note that a configuration is never repeated in the course of the computation – all of the $K + 1$ configurations are distinct, and therefore orthogonal.

### 3.3.1 Legal configurations

At the moment, we are interested only in the (legal) configurations that we want to appear during a computation. Of course, the whole Hilbert space is much larger, containing many other states. We will call those illegal, and want them to be “detectable”. For now, we will not deal with these other states until Section 3.4.1.

Let the set of legal configurations $C_0, \ldots, C_K$ be the $K+1$ configurations that can be obtained by applying the rules in Table 3.1 starting with the initial configuration (3.4). The legal configurations correspond to the $K + 1$ (including the initial state) intermediate computational states generated by the circuit $\tilde{V}_x$. We call all other configurations illegal.

We will now look at the properties shared by the legal configurations. It will be convenient to look at pairs of particles at locations $(2i - 1, 2i)$ and $(2i + 1, 2i + 2)$. Table 3.3 lists the allowed pairs of symbols and which ones can be adjacent to each other. The pairs play the roles of “dead” (labeled x, particles not to be used anymore), “done” (labeled D, qubits to the left of the active site), “active” (labeled A, the active site), “ready” (labeled R, qubits to the right of the active site) and “unborn” (labeled u, “unborn” particles, not used yet) from the construction in [AGIK09]. There, the legal states were of the form $(x \cdots x) (D \cdots D) A (R \cdots R) (u \cdots u)$, with a single active site.
Table 3.2: The configurations occurring in one cycle of the computation with \( n = 3 \) qubits. The rules whose application brings the state to the next one are listed on the right.
Here \((z \cdot \cdot z)\) stands for a variable-length string made from the letter “z”.

Connecting subsequent pairs according to the rules listed in Table 3.3 imposes a particular form for the legal states (brackets indicate variable-length, possibly empty substrings)

\[
(x \cdot \cdot x)[\text{qubits}](u \cdot \cdot u)
\]

where \([\text{qubits}]\) is a \textit{nonzero} string of the form

\[
A_x(R \cdot \cdot R) R_u
\]

\[
(D_x D \cdot \cdot D) A_1 (R \cdot \cdot R) R_u
\]

\[
D_x (D \cdot \cdot D) A_p (R \cdot \cdot R) R_u
\]

\[
D_x (D \cdot \cdot D) A_2 (R \cdot \cdot RR_u)
\]

\[
D_x (D \cdot \cdot D) A_u
\]

\[
(R \cdot \cdot R) R_u
\]

\[
D_x (D \cdot \cdot D)\]

(3.5)

The first five options involve an “active” pair, while the last three have no “active” pair in them. Furthermore, note that the whole \([\text{qubits}]\) string cannot be empty, because the rightmost particle of the whole chain cannot be \(\otimes\), the leftmost one cannot be \(\bigcirc\) and the combination \(\otimes \bigcirc\) is illegal.

Next, for legal configurations, the number of particles holding qubits needs to be exactly \(n\).

Let us have a closer look at the legal \([\text{qubits}]\) strings with an active site \(\bigotimes\) or \(\bigcirc\), which translates to a single active pair \((A_x, A_1, A_p, A_2 \text{ or } A_u)\). One example is \(\begin{array}{c|c|c}\bigotimes & \bigotimes \end{array}\) (which is of the type \(D_x A_1 R_u\)). In the case there is no \(D_x\) pair, the active pair has to be \(A_x\) (3.6) or \(A_1\) (3.7) – an example is the sequence \(\begin{array}{c|c|c|c}\bigotimes & \bigotimes \end{array}\) (with pairs \(A_x RR_u\)). In the case there is no \(R_u\) pair, the active pair has to be of the \(A_2\) or \(A_u\), as in e.g. \(\begin{array}{c|c|c|c} & \bigotimes \end{array}\) (this is \(D_x DDA_u\)).

The other three types of legal substrings \([\text{qubits}]\) do not have an active pair. First, we could have a done qubit pair on the right end as in \(\begin{array}{c|c|c|c} & & \end{array}\) (simply \(D_x DD\) without any R’s).
allowed to be followed by

| property                  | symbol pair | x  | Dx | Dx | Ax | Ax | Ap | Ap | A2 | A2 | Au | Au | Ru | Ru | u  |
|---------------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| dead: x                   | ✗ ✗         | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| dead+done: Dx             | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| done: D                   | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| active leftmost: Ax       | ✗ ▶         | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| active gate 1: A1         | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| active pusher: Ap         | ✗ ▶         | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| active gate 2: A2         | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| active rightmost: Au      | ✗ ▶         | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| ready: R                  | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| ready+unborn: Ru          | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |
| unborn: u                 | ✗           | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  |

Table 3.3: Building up the legal configuration structure from pairs of symbols (unlisted symbol pairs do not appear in legal configurations). We list symbol pairs allowed at positions \((2i - 1, 2i)\) and their designated followups at positions \((2i + 1, 2i + 2)\). Note the mirror symmetry of the table across the antidiagonal. The allowed configurations of the whole chain must then have form \((x \cdots x)[\text{qubits}](u \cdots u)\), with a substring \([\text{qubits}]\) given by (3.6)-(3.13), with at most one active pair. Further restrictions come into play from considering the block boundary locations (see Table 3.5) the number of “qubits” and their proper alignment with respect to the block boundaries.

Second, observe that two neighboring \(\Box\) particles can appear at positions \((2i, 2i + 1)\), when coming from two consecutive pairs as in \(|\Box \Box\Box|\Box \Box\Box\Box\Box\Box\Box\Box|\) (read as \(D_x DR_u\)). Finally, it is possible to have a ready qubit pair on the left end as in \(|\Box \Box\Box|\Box \Box\Box\Box\Box\Box\Box\Box\Box\Box\Box|\) (simply \(RRR_u\) without any \(D\)'s).

The location of the \([\text{qubits}]\) substring matters. For a legal configuration with a \(\pmb{◳}\) symbol, the string \([\text{qubits}]\) must fit exactly between two block boundaries as \(|[\text{qubits}]\rangle\) (see Table 3.2). On the other hand, the string \([\text{qubits}]\) without the symbol \(\pmb{◳}\) always has runs across a block boundary somewhere. These two properties later help us check that we do not have too few or too many qubits or whether the qubits are properly aligned between the boundaries, ruling out illegal but locally undetectable states.

### 3.4 The Hamiltonian

We aim to construct a Hamiltonian corresponding to a circuit \(\tilde{V}_x\) such that the ground state energy of the Hamiltonian is \(E \leq a\) for ‘yes’ instances \((x \in L)\) and \(E \geq b\) for the ‘no’ instances, where \(a\) and \(b\) have a \(1/poly(n)\) separation. We will show the history state of the computation on the
1. $\otimes\otimes\otimes((\square\circ)^n)^{n-2}\square\circ((\bigcirc\bigcirc)^n)\bigcirc$ (2b)

2. $\otimes\otimes\otimes\otimes((\square\circ)^n)^{n-2}\square\circ((\bigcirc\bigcirc)^n)\bigcirc$ (1)

3. For $i$ from 0 to $n - 3$:
   (a) $\otimes\otimes\otimes\otimes((\square\circ)^i)((\square\circ)^n-i-2)((\bigcirc\bigcirc)^i)((\bigcirc\bigcirc)^n)\bigcirc$ (2a)
   (b) $\otimes\otimes\otimes\otimes((\square\circ)^i)((\square\circ)^n-i-2)((\bigcirc\bigcirc)^i)((\bigcirc\bigcirc)^n)\bigcirc$ (1)

4. $\otimes\otimes\otimes((\square\circ)^n)^{n-2}\square\circ((\bigcirc\bigcirc)^n)\bigcirc$ (2c)

5. $\otimes\otimes\otimes((\square\circ)^n)^{n-2}\otimes\otimes((\bigcirc\bigcirc)^n)\bigcirc$ (4a)

6. $\otimes\otimes\otimes((\square\circ)^n)^{n-1}\otimes\circ((\bigcirc\bigcirc)^n)\bigcirc$ (5a)

7. For $j$ from 0 to $n - 2$:
   (a) For $k$ from 0 to $n - 2$:
      i. $\otimes\otimes\otimes\otimes((\square\circ)^j)((\square\circ)^n-k-2)((\bigcirc\bigcirc)^j)((\bigcirc\bigcirc)^n)\bigcirc((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (5b)
      ii. $\otimes\otimes\otimes\otimes((\square\circ)^j)((\square\circ)^n-k-2)((\bigcirc\bigcirc)^j)((\bigcirc\bigcirc)^n)\bigcirc((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (5a)
   (b) $\otimes\otimes\otimes\otimes((\square\circ)^j)(\square\circ)((\bigcirc\bigcirc)^n-1)\square\circ((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (6b)
   (c) $\otimes\otimes\otimes\otimes((\square\circ)^j)(\square\circ)((\bigcirc\bigcirc)^n-1)\square\circ((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (3a)
   (d) For $l$ from 0 to $n - 3$:
      $\otimes\otimes\otimes\otimes((\square\circ)^j+1)((\square\circ)^j)((\bigcirc\bigcirc)^n-l-2)\square\circ((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (3b)
   (e) $\otimes\otimes\otimes\otimes((\square\circ)^j+1)((\square\circ)^j)((\bigcirc\bigcirc)^n-l-2)\square\circ((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (3c)
   (f) $\otimes\otimes\otimes\otimes((\square\circ)^j+1)((\square\circ)^j)((\bigcirc\bigcirc)^n-2)\square\circ((\bigcirc\bigcirc)^n-j-1)\bigcirc$ (4b)
   (g) $\otimes\otimes\otimes\otimes((\square\circ)^j+1)((\square\circ)^j)((\bigcirc\bigcirc)^n-2)\square\circ((\bigcirc\bigcirc)^n-j-2)\bigcirc$ (5a)

8. For $i$ from 0 to $n - 2$:
   (a) $\otimes\otimes\otimes\otimes((\square\circ)^n-1)((\square\circ)^i)((\bigcirc\bigcirc)^n-i-2)\square\circ((\bigcirc\bigcirc)^i)\square\circ\bigcirc$ (5b)
   (b) $\otimes\otimes\otimes\otimes((\square\circ)^n-1)((\square\circ)^i)((\bigcirc\bigcirc)^n-i-2)\square\circ((\bigcirc\bigcirc)^i)\square\circ\bigcirc$ (5a)

9. $\otimes\otimes\otimes\otimes((\square\circ)^n-1)\otimes\otimes\otimes((\bigcirc\bigcirc)^n-1)\square\circ\bigcirc$ (6a)

10. $\otimes\otimes\otimes\otimes((\square\circ)^n)\square\circ((\bigcirc\bigcirc)^n-2)\square\circ\bigcirc$

Table 3.4: The general form of the sequence of legal configurations in one round of computation. The middle block boundary is not shown in steps 7 and 8. The full computation ends at Step 5 without the trailing circles. A particular example for $n = 3$ is shown in Table 3.2.
witnesses for ‘yes’ instances has a low energy. Our Hamiltonian is a sum of four terms:

\[ H := J_{\text{in}}H_{\text{in}} + J_{\text{prop}}H_{\text{prop}} + J_{\text{pen}}H_{\text{pen}} + H_{\text{out}}. \]

The coefficients \( J_{\text{in}}, J_{\text{prop}}, \) and \( J_{\text{pen}} \) will later be chosen to be some polynomials in \( n \). For the term \( H_{\text{prop}} \), any valid history state (a uniform superposition of legal configurations whose qubit content comes corresponds to the computation with the gates from \( \tilde{V}_x \)) will be a zero-energy state. The term \( H_{\text{in}} \) raises the energy of states which do not have ancilla qubits initialized to 0, which is required in the circuit \( \tilde{V}_x \). The role of \( H_{\text{out}} \) is to raise the energy of the states which encode computations that are not accepted. Finally, the terms in \( H_{\text{pen}} \) penalize (i.e. raise the energy of) locally detectable illegal configurations which do not have the proper form as described by equations (3.5)-(3.13) in Section 3.3.1.

We start with the ancilla-checking term \( H_{\text{in}} \), defined as

\[ H_{\text{in}} := |\begin{array}{c}\begin{array}{c}\uparrow \end{array}\end{array}|^{(1)}\rangle \langle |\begin{array}{c}\begin{array}{c}\uparrow \end{array}\end{array}|^{(1)}|_1 + \sum_{i=2}^{n-m} |\begin{array}{c}\begin{array}{c}\square \end{array}\end{array}|^{(1)}\rangle \langle |\begin{array}{c}\begin{array}{c}\square \end{array}\end{array}|^{(1)}|_{2i-1}. \]

By raising the energy of states with qubit content \(|1\rangle\), it ensures that in a low-energy state candidate the ancilla qubits (the first \( n-m \)) are all initially (in the initial configuration (3.4) they are located at odd positions in the first block) in the \(|0\rangle\) state. Without loss of generality, we assume that the first round of \( V'_x \) consists of identity gates. This is necessary when we want the ancilla qubits to remain unpenalized by \( H_{\text{in}} \) until they are moved from the first block into the second block.

The term \( H_{\text{out}} := |\begin{array}{c}\begin{array}{c}\uparrow \end{array}\end{array}|^{(0)}\rangle \langle |\begin{array}{c}\begin{array}{c}\uparrow \end{array}\end{array}|^{(0)}|_{2nR} \) checks that when the computation finishes (the \(|\begin{array}{c}\begin{array}{c}\uparrow \end{array}\end{array}|\rangle\) state appears at the very right end of the qudit chain), the qubit content of the output qubit is \(|1\rangle\). For computations that do not accept, the output qubit state is \(|0\rangle\) and \( H_{\text{out}} \) penalizes this.

In defining the remaining terms of the Hamiltonian, we will need to be able to distinguish between different kinds of configurations of the chain. We classify three types:

1. **legal** configurations are defined in Section 3.3.1 to be the configurations \( C_0, \ldots, C_K \) occurring during the computation with a circuit \( \tilde{V}_x \) when starting with the initial configuration (3.4) with \( n \) qudits. All other configurations are **illegal**.

2. **locally detectable illegal** configurations are those that contain a pair of neighboring qudits
labeled by a pair of symbols that does not occur in the legal configurations. These can be identified and penalized locally by means of a projector onto such a pair.

3. *locally undetectable illegal* configurations are those that are not detectable by local projections, but are still not legal, as they do not appear in the legal progression of a computation. As shown in Lemma [12] these states have too many or too few qubits, or an improperly aligned [qubits] block.

### 3.4.1 The penalty Hamiltonian

The role of $H_{\text{pen}}$ is to ensure that there are no locally detectable illegal configurations in the computation. That is, we wish to leave the legal states unpunished while raising the energy of the locally detectable illegal ones by projecting on neighboring pairs of symbols that do not occur in a proper course of computation described in Section 3.3. We call such pairs *forbidden*. Since we have 6 different symbols in the construction, there are 36 possible neighboring pairs. Furthermore, we can distinguish 5 types of location pairs depending on the parity of the positions and their position with respect to the block boundaries $\parallel$ as listed on the right in Table 3.5. We list the 56 allowed pairs of symbols in Table 3.5 which gives us $36 \times 5 - 56 = 124$ types of projector terms

$$|XY\rangle\langle XY|_{i,i+1},$$

where $XY \in \{\otimes, \bigcirc, \odot, \bigodot, \bigcirc\bigotimes\}$ is a forbidden pair at a location $(i, i+1)$. For example, the forbidden pair $\otimes\bigcirc$ (disallowed in all 5 types of locations) is energetically penalized by

$$\sum_{f=1}^{5} H_{\text{pen},f} = \sum_{i=1}^{2nR-1} \mathbb{I}_{1,\ldots,i-1} \otimes |\otimes\bigcirc\rangle\langle\otimes\bigcirc|_{i,i+1} \otimes \mathbb{I}_{i+2,\ldots,2nR}. \tag{3.15}$$

while the pair $\bigboxtimes\bigboxtimes$ is forbidden on even-parity sites (type A,C,E), and is penalized by

$$\sum_{f=93}^{95} H_{\text{pen},f} = \sum_{i=1}^{nR} \mathbb{I}_{1,\ldots,2i-2} \otimes |\bigboxtimes\bigboxtimes\rangle\langle\bigboxtimes\bigboxtimes|_{2i-1,2i} \otimes \mathbb{I}_{2i+1,\ldots,2nR}. \tag{3.16}$$

To take the qubit content of the $\bigboxtimes$ particles into account, as in [AGIK09] we use the notation $|A\rangle\langle B| := \sum_s |a_s\rangle\langle b_s|$, meaning that subspace $B$ is mapped to subspace $A$ in some prescribed way
Table 3.5: The 56 allowed pairs $XY$ of symbols at positions $(i, i+1)$ in the $d = 8$ construction according to $H_{\text{pen}}$. There are 5 types of locations (A, B, C, D, E) for the pair, according to location parity and block-boundary position. For each of the 36 symbol combinations, we list its allowed location types. The forbidden pairs implied by this table are penalized by $H_{\text{pen}}$ (3.17).

specified by the pairing of the basis vectors. Thus, $\langle 00\rangle\langle 00\rangle$ preserves the qubit contents as $\langle 00\rangle\langle 00\rangle := \sum_{s,t=0}^{1} \langle s\rangle\langle t\rangle = \sum_{s,t=0}^{1} \langle s\rangle\langle t\rangle$.

Furthermore, to rule out configurations without any qubit-holding particles, we need to penalize the symbols $\{\odot, \bigcirc, \bigotimes, \bigoplus\}$ at the leftmost qudit (only $\otimes$ or $\bigtriangledown$ can appear there) and project onto $\{\bigotimes, \bigcirc, \bigotimes, \bigoplus\}$ on the rightmost qudit (only $\bigtriangledown$ or $\bigcirc$ are allowed at the right end). Together, the Hamiltonian imposing an energy penalty on configurations containing any of the forbidden pairs is

$$H_{\text{pen}} = \sum_{f=1}^{124} H_{\text{pen},f} + H_{\text{left}} + H_{\text{right}}.$$  \hspace{1cm} (3.17)

Observe that $H_{\text{pen}}$ only catches illegal configurations with $\text{locally detectable}$ errors, and there exist illegal configurations that are not locally detectable, i.e., that have zero energy under $H_{\text{pen}}$, such as this one with too many qubits

$$\bigotimes \bigotimes \bigcirc \bigcirc \bigotimes \bigotimes \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc.$$

To identify these states as illegal, we will have to show they propagate into states with forbidden pairs. First, though, we want to ensure this propagation, which is the topic of the next section.
3.4.2 The propagation Hamiltonian

We want to check whether the computation on the line of qudits proceeds correctly, in a linear sequence of configurations $C_0 \leftrightarrow \cdots \leftrightarrow C_t \leftrightarrow C_{t+1} \leftrightarrow \cdots \leftrightarrow C_K$ (see Section 3.3 and the example in Table 3.2), ensuring the intended unitary operations are applied in the correct order.

The propagation-checking Hamiltonian $H_{\text{prop}}$ should have a low energy only for a state which is a superposition of all the legal configurations, with the gates applied to their qubit content as planned.

For now, let us look only at the states from the span of the legal configurations, where we want $H_{\text{prop}}$ to give an energy penalty to all states except the history states corresponding to the circuit $\tilde{V}_x$. We would like to construct it as $H_{\text{prop}} = \sum_{t=0}^{K-1} H_t$, where $H_t$ checks the transition from the state $|\psi_t\rangle$ to $|\psi_{t+1}\rangle$. For a candidate low-energy state that has a nonzero overlap with $|\psi_t\rangle$, it should insist that it has to have the same amplitude as the state $|\psi_{t+1}\rangle$. Any two legal configurations are orthogonal, so if locality did not matter, it would suffice to use projections onto the states $|\psi_t\rangle$ as in (3.2) in Section 3.2. However, we want our Hamiltonian to be 2-local.

The computation of the circuit $\tilde{V}_x$ on an initial state runs according to the rules in Table 3.1, which are (up to) 4-local. A rule $\text{LNOR} \leftrightarrow \text{LPQR}$ applied at some location corresponds to a transition between states $|\psi_t\rangle = |\cdots \text{LNOR} \cdots\rangle$ and $|\psi_{t+1}\rangle = |\cdots \text{LPQR} \cdots\rangle$. In the language of Hamiltonians, this transition is facilitated by

$$(\mathbb{I} \otimes (|\text{LPQR}\rangle \langle \text{LNOR}| + |\text{LNOR}\rangle \langle \text{LPQR}|) \otimes \mathbb{I}) |\psi_t\rangle = |\psi_{t+1}\rangle$$

To penalize states whose overlap with the states $|\psi_t\rangle$ and $|\psi_{t+1}\rangle$ is not the same, we would use

$$\mathbb{I} \otimes (|\text{LNOR}\rangle \langle \text{LNOR}| + |\text{LPQR}\rangle \langle \text{LPQR}|) \otimes \mathbb{I}$$

$$-\mathbb{I} \otimes (|\text{LPQR}\rangle \langle \text{LNOR}| + |\text{LNOR}\rangle \langle \text{LPQR}|) \otimes \mathbb{I},$$

which within the subspace spanned by $|\psi_t\rangle$ and $|\psi_{t+1}\rangle$ projects onto a state proportional to $|\psi_t\rangle - |\psi_{t+1}\rangle$. The equal superposition of the two states is thus an eigenvector with eigenvalue 0. However, we want to use 2-local, not 4-local operators. If we simply involved only the two particles that

\footnote{Up to a constant, as it equals $2|\alpha\rangle \langle \alpha|$ with $|\alpha\rangle = \frac{1}{\sqrt{2}} (|\psi_t\rangle + |\psi_{t+1}\rangle)$.}
actually change \((NO \leftrightarrow PQ)\), it would be possible that the resulting terms like \(|PQ\rangle\langle NO|\) would apply to several places in a given configuration, leading to a branching of the legal configuration sequence, instead of producing a simple connected line \(C_0 \leftrightarrow \cdots \leftrightarrow C_t \leftrightarrow C_{t+1} \leftrightarrow \cdots \leftrightarrow C_K\). This could doom the construction by giving some energy to history states. However, we will now show how to construct \(H_t\) from several 2-qudit terms that can “pick out” and “check” the intended transitions between the configurations \(C_t\) and \(C_{t+1}\). The trick involves 2-local terms on surrounding qudits as well.

Let us look at a forward application of a rule that changes the qudits \((i, i+1)\), taking them from a sub-configuration \(NO_{(i,i+1)}\) to a sub-configuration \(PQ_{(i,i+1)}\). We constructed the legal configurations (see Table 3.4) so that this rule is applicable only to a configuration that is uniquely identifiable by a sub-configuration \(XY_{(j,j+1)}\) on some nearby qudits \((j,j+1)\). Similarly, the backwards applicability of this rule is uniquely identifiable by a sub-configuration \(ZW_{(k,k+1)}\) on some nearby qudits \((k,k+1)\). We now write a Hamiltonian checking the application of this rule as

\[
H_{\text{prop},i}^{(\text{rule})} = |XY\rangle\langle XY|_{j,j+1} + |ZW\rangle\langle ZW|_{k,k+1} - |PQ\rangle\langle NO|_{i,i+1} - |NO\rangle\langle PQ|_{i,i+1}. \tag{3.19}
\]

In the simplest case, \(XY = NO\), \(ZW = PQ\) and \(i = j = k\), so that \(XY, ZW, NO, PQ\) all involve the same pair of particles \((i, i+1)\). For a more complicated case, let us look at rule 4(b) \(\Box\bigcirc\bigcirc \leftrightarrow \Box\bigcirc\Box\) from Table 3.1. The forward applicability of the rule is uniquely identified by the substring \(\Box\bigcirc\) on the first two particles, while the backwards applicability of this rule is uniquely identified by the substring \(\bigcirc\Box\) on the second and third particles. The Hamiltonian term checking this rule will be given in (3.27).

Let us now look at an example from a unary clock construction, to see that history states retain a zero-energy for a Hamiltonian of the type (3.19).

**Analogy with [KKR06]**

As an example, we recall [KKR06], where the propagation Hamiltonian was reduced from 3-local to 2-local. There, checking the progress of a unary clock register \(|s\rangle = |1\cdots 1,s,0\cdots 0\rangle\) can be done
with a 3-local Hamiltonian

\[ H_t = (|100\rangle\langle100| + |110\rangle\langle110| - |110\rangle\langle100| - |100\rangle\langle110|)_{t,t+1,t+2}, \]  

(3.20)

uniquely picking the states \(|t\rangle\) and \(|t+1\rangle\) for which the transition rule 100 ↔ 110 applies. Instead, we can make it out of 2-local (and 1-local) terms

\[ H'_t = |10\rangle\langle10|_{t,t+1} + |10\rangle\langle10|_{t+1,t+2} - |1\rangle\langle0|_{t+1} - |0\rangle\langle1|_{t+1}. \]  

(3.21)

with the first two terms uniquely identifying the places where the rule should apply, while the last two (transition) terms are ambiguous in their applicability. The price for the decrease in locality are “mistimed” transitions such as \(|111100\rangle \rightarrow |110100\rangle\) in the unary clock register. However, observe that the expectation value of this Hamiltonian in the uniform superposition of unary clock states is zero, i.e.

\[ \frac{1}{T+1} \sum_{r,s=0}^{T} \langle r | H'_t | s \rangle = 0, \]

because the mistimed transitions in \(H'_t\) take the state out of the legal subspace, to states orthogonal to any of the proper unary clock states. On the other hand, the transitions taking place at proper places are easily shown to result in 0 energy. Thus, the restriction of \(H'_t\) to the legal clock subspace spanned by \(\{|0\rangle, |1\rangle, \ldots, |T\rangle\}\) works exactly as the Hamiltonian \(H_{\text{prop,unary}}^{\text{prop}} = |t\rangle\langle t| + |t+1\rangle\langle t+1| - |t+1\rangle\langle t| - |t\rangle\langle t+1|\) from (3.2). We conclude that a correct history state (a superposition of all legal states) has expectation energy zero under the decreased-locality propagation-checking Hamiltonian (3.21) from [KKR06].

The projector terms in \(H'_t\) thus picked the applicability place uniquely, while the mistimed transitions coming from the last two terms took the state out of the legal subspace (to non-unary clock states). We will now use this insight to construct our \(H_{\text{prop}}\) from 2-local terms. However, our task is more complicated, because the unary-clock propagation rule 100 → 110 for a certain location applied just once in a sequence of proper clock states. In our case, a rule for moving a qubit (or a gate particle) at a given location in the chain could connect configurations \(C_t \leftrightarrow C_{t+1}\) as well as some other configurations \(C_{t'} \leftrightarrow C_{t'+1}\) with the data in the chain shifted by a few positions.
Explicit form of the propagation-checking terms

Instead of writing the propagation Hamiltonian as a sum of terms $H_t$, we choose to write it out as a sum of terms $H_{\text{prop}, i}^{(\text{rule } \rho)}$ corresponding to different rules $\rho$ applied at location pairs $(i, i+1)_\rho$ wherever rule $\rho$ is applicable. This generalization is required because one rule $\rho$ for a pair of particles can facilitate legal transitions between several configuration pairs. The propagation Hamiltonian is then

$$H_{\text{prop}} = \sum_{\rho=1}^{6} \sum_{(i, i+1)_\rho} H_{\text{prop}, i}^{(\text{rule } \rho)}.$$  

(3.22)

and we want its application to a state $|\psi_t\rangle$ (corresponding to a legal configuration $C_t$ with $2 \leq t \leq K$) to result in

$$H_{\text{prop}} |\psi_t\rangle = 2|\psi_t\rangle - |\psi_{t-1}\rangle - |\psi_{t+1}\rangle + \text{illegal but locally detectable states.}$$

(3.23)

The propagation term corresponding to rule 1 ($\uparrow \leftrightarrow U_{t,i}$) in Table 3.1 is simple, as it involves only the sites $i$ and $i+1$ and does not create bad transitions. We want to check the transfer of $\uparrow$ to the right and the application of the gate $U_{t,i}$ (corresponding to the location $i$) to the qubit content of the two neighboring sites. This is done by

$$H_{\text{prop}, i}^{(\text{rule 1})} = |\uparrow\rangle \langle \uparrow|_{i, i+1} + |\downarrow\rangle \langle \downarrow|_{i, i+1} - U_{t,i} |\downarrow\rangle \langle \uparrow|_{i, i+1} - U_{t,i}^\dagger |\uparrow\rangle \langle \downarrow|_{i, i+1},$$

(3.24)

and this term appears only for locations $(i, i+1)$ of the type B in Table 3.5.

We continue with rule 2 ($\uparrow \leftrightarrow \downarrow$) for moving the $\uparrow$ from position $i$ to $i+1$. Depending on the location in the chain, the Hamiltonian term reads

$$H_{\text{prop}, i}^{(\text{rule 2})} = |\downarrow\rangle \langle \uparrow|_{i, i+1} + |P\rangle \langle P|_{i, i+1} - |P\rangle \langle P|_{i, i+1} - |\uparrow\rangle \langle \downarrow|_{i, i+1},$$

(3.25)

(3.26)

with $PO = \odot \odot$ for locations $(i, i+1)$ of type A, $PO = \otimes \odot$ for locations of type C, and
\( PO = \ominus \odot \odot \) for locations of type E.

The propagation terms for rule 4 (\( \blacklozenge \odot \ominus \ominus \iff \blacklozenge \odot \ominus \ominus \), \( \blacklozenge \odot \ominus \odot \iff \blacklozenge \odot \ominus \odot \)) govern the creation of the symbol \( \ominus \). We now involve three particles, but again, only two at a time, leaving the Hamiltonian 2-local:

\[
H_{\text{prop, rule 4}} = |X \ominus \rangle \langle X \ominus |_{i,i+1} + |\ominus \ominus \rangle \langle \ominus \ominus |_{i+1,i+2} - |\blacklozenge \blacklozenge \rangle \langle \blacklozenge \blacklozenge |_{i,i+1}
\]

(3.27)

with \( X = \blacklozenge \) at locations of the type D and \( X = \blacklozenge \) at locations of the type B. Only the projector term identifying the backwards applicability of rule 4 involves a particle pair different from \((i, i+1)\).

Rule 5 (\( \blacklozenge \blacklozenge \ominus \iff \ominus \blacklozenge \blacklozenge \), \( \blacklozenge \blacklozenge \blacklozenge \ominus \iff \ominus \blacklozenge \blacklozenge \blacklozenge \)) pushes \( \ominus \) to the left, and its checking Hamiltonian is again simple:

\[
H_{\text{prop, rule 5}} = |X \ominus \rangle \langle X \ominus |_{i,i+1} + |\ominus \ominus \rangle \langle \ominus \ominus |_{i,i+1} - |\ominus \ominus \rangle \langle \ominus \ominus |_{i,i+1} - |X \ominus \rangle \langle X \ominus |_{i,i+1} - |X \ominus \rangle \langle X \ominus |_{i,i+1}
\]

(3.28)

with \( X = \blacklozenge \) at locations of the type ACE and with \( X = \blacklozenge \) at locations of the type BD.

The Hamiltonian for rule 6 (\( \blacklozenge \blacklozenge \blacklozenge \ominus \iff \blacklozenge \blacklozenge \blacklozenge \ominus \), \( \blacklozenge \blacklozenge \blacklozenge \blacklozenge \ominus \iff \blacklozenge \blacklozenge \blacklozenge \blacklozenge \ominus \)) kills the symbol \( \ominus \) and mirrors the ones for rule 4:

\[
H_{\text{prop, rule 6}} = |X \ominus \rangle \langle X \ominus |_{i,i+1} + |\ominus \ominus \rangle \langle \ominus \ominus |_{i,i+1} - |\ominus \ominus \rangle \langle \ominus \ominus |_{i,i+1} - |X \ominus \rangle \langle X \ominus |_{i,i+1} - |X \ominus \rangle \langle X \ominus |_{i,i+1}
\]

(3.29)

with \( W = \blacklozenge \) at locations of the type D and \( W = \blacklozenge \) at locations of the type B.

Rule 3 is the most complicated one since its definition is 4-local. We reduce the locality by looking only at qudit pairs \((i - 1, i)\) and \((i + 1, i + 2)\) to identify the applicability of this rule to states \(|\psi_i\rangle\) and \(|\psi_{i+1}\rangle\) which can be connected by an application of rule 3. Note that the pair \((i, i + 1)\) has to be of the type ACE.

We begin by writing out the propagation terms corresponding to each of the possible transitions...
Rule 3b \( (\square\square\circ\square \leftrightarrow \square\square\circ\square\circ\square) \) applies to pairs \((i, i + 1)\) of type \(A, \ C, \) and \(E:\)

\[
H_{\text{prop},i}^{(\text{rule 3b})} = |\square\square\rangle\langle \square\square|_{i-1,i} + |\square\circ\rangle\langle \square\circ|_{i+1,i+2}
- |\square\rangle\langle \square|_{i,i+1} - |\circ\rangle\langle \circ|_{i,i+1}.
\] (3.30)

Rule 3a \( (\otimes\square\circ\square \leftrightarrow \otimes\circ\circ\square\circ) \) applies to pairs \((i, i + 1)\) of type \(A\) and \(E:\)

\[
H_{\text{prop},i}^{(\text{rule 3a})} = |\otimes\rangle\langle \otimes|_{i-1,i} + |\circ\rangle\langle \circ|_{i+1,i+2}
- |\otimes\rangle\langle \otimes|_{i,i+1} - |\circ\rangle\langle \circ|_{i,i+1}.
\] (3.31)

Rule 3c \( (\square\square\circ\circ \leftrightarrow \square\square\circ\circ\circ) \) acts on pairs of type \(A\) and \(C:\)

\[
H_{\text{prop},i}^{(\text{rule 3c})} = |\circ\rangle\langle \circ|_{i-1,i} + |\circ\rangle\langle \circ|_{i+1,i+2}
- |\circ\rangle\langle \circ|_{i,i+1} - |\circ\rangle\langle \circ|_{i,i+1}.
\] (3.32)

Finally, rule 3d \( (\otimes\square\circ\circ \leftrightarrow \otimes\circ\circ\square\circ) \) handles a special type of illegal configuration that contains only a single qubit-holding particle. In combination with rules 4-6, it helps to move this qubit until it reaches a locally-detectable illegal configuration. For locations \((i, i + 1)\) of type \(A, \ C\) and \(E,\) we write

\[
H_{\text{prop},i}^{(\text{rule 3d})} = |\otimes\rangle\langle \otimes|_{i-1,i} + |\circ\rangle\langle \circ|_{i+1,i+2}
- |\otimes\rangle\langle \otimes|_{i,i+1} - |\circ\rangle\langle \circ|_{i,i+1}.
\] (3.33)

To obtain the overall Hamiltonian for rule 3, we do not simply sum these four terms, as we would include the terms \(|\otimes\rangle\langle \otimes|_{i-1,i}, \ |\circ\rangle\langle \circ|_{i+1,i+2}, \ |\circ\rangle\langle \circ|_{i-1,i}, \ |\circ\rangle\langle \circ|_{i+1,i+2}\) twice. The function of these projectors is to pick out the ‘before’ and ‘after’ configurations of the corresponding transition rule. Since we want each legal configuration to picked out exactly once,
we include them only in a single copy, i.e.

\[ H_{\text{prop}}^{(\text{rule } 3)} = |\bigcirc|\langle \bigcirc |_{i-1,i} + |\bigcirc|\langle \bigcirc |_{i+1,i+2} \]

\[ + |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i-1,i} + |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i+1,i+2} \]  

\[ - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} \]  

\[ - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} \]  

\[ - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i+1,i+1} \]  

\[ - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} - |\bigcirc\bigcirc|\langle \bigcirc\bigcirc |_{i,i+1} \]  

(3.34)

(3.35)

(3.36)

(3.37)

(3.38)

(3.39)

where the first four lines apply at locations \((i, i+1)\) of the types ACE, and the last two lines apply only at the location types listed (AE and AC). The projector terms in this Hamiltonian term applied to a legal configuration now gives something nonzero only when rule 3 could be applied to this legal configuration – and induces exactly one forward and one backward legal transition. Possibly, it could also induce illegal transitions, which lead to illegal states detectable by \(H_{\text{pen}}\). On the other hand, when rule 3 (in its 4-local form) is not applicable to a given configuration, we get no projection terms, only transitions to illegal states.

Note that we need to fix the prescriptions at the ends of the chain. We do this by omitting the particles with positions with \(i-1 < 1, i+1 > 2nR \) or \(i+2 > 2nR\), e.g., using only a single-particle projector \(|\bigcirc|\langle \bigcirc |_1\) as the first term in (3.34) at the location pair \((1,2)\). Together, all these terms make up \(H_{\text{prop}}\).
Applying the propagation Hamiltonian: examples

Let us see the Hamiltonian for rule 3 in action. We list a few examples, applying it in a location of the type E (with a block boundary on the right), first to legal configurations:

<table>
<thead>
<tr>
<th>a legal configuration</th>
<th>applying $H_{\text{prop},i}^{(\text{rule }3)}$ for $(i, i + 1) = (5, 6)$ gives</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1 = \circ \circ</td>
<td>\times</td>
</tr>
<tr>
<td></td>
<td>$- \circ \circ</td>
</tr>
<tr>
<td></td>
<td>$- \circ \times</td>
</tr>
<tr>
<td>$C_2 = \circ \circ</td>
<td>\circ \times</td>
</tr>
<tr>
<td></td>
<td>$- \circ \circ</td>
</tr>
<tr>
<td></td>
<td>$- \circ \times</td>
</tr>
<tr>
<td>$C_3 = \times</td>
<td>\circ</td>
</tr>
<tr>
<td></td>
<td>$- \times</td>
</tr>
</tbody>
</table>

For the first configuration $C_1$, the pair $(5, 6)$ is indeed where rule 3 should apply. Thus we get a projection from (3.34), and a legal transition from (3.36). There is an additional illegal transition from (3.38), locally detectable by the illegal pair $\times \circ$. The second configuration $C_2$ should transform forward by applying rule 3. The configuration is projected by (3.35), has a legal transition from (3.38) and an extra illegal one from (3.36) with the bad substring $\times \circ$. For the third configuration $C_3$, rule 3 should not apply (the proper-transition producing rule is now rule 5 involving $\circ$) – thus we get nothing from the projection terms (3.34)-(3.35). Instead, we get two transitions to illegal states – (3.36) creates a state with a bad substring $\circ \circ$ and (3.38) makes a state detectable by the pair $\circ \circ$.

Let us look at one more example, checking what $H_{\text{prop},i}^{(\text{rule }3)}$ does to an illegal, allowed but not locally detectable configuration. This special case is crucial for detecting configurations with badly
aligned blocks or with too few/too many qubits.

<table>
<thead>
<tr>
<th>an allowed but illegal configuration</th>
<th>applying $H^{\text{rule 3}}_{\text{prop,4}}$ for the middle pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_4 = \cdots \otimes \otimes</td>
<td>\square\square</td>
</tr>
<tr>
<td></td>
<td>$- \otimes \otimes</td>
</tr>
</tbody>
</table>

(3.40)

The configuration $C_4$ is projected once by (3.35) and an illegal transition from (3.37) makes a configuration with a forbidden pair $\square \square$. Note that we did not obtain a legal transition, even though rule 3 was applicable and gave us a projection term. In Section 3.6.2 we will translate this into a lower bound on the ground state energy for such states.

Classifying the legal and illegal transitions

When $H_{\text{prop}}$ (3.22) acts on a state of the system, it induces changes in the configuration (besides sometimes performing a two-qubit unitary operation). This construction differs from the one in [AGIK09] in that the changes can occur at more than one location along the chain. This is readily apparent when one considers the action of the whole propagation Hamiltonian on the state with a configuration such as $\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots$ (the first line in Table 3.2). We obtain

$$H_{\text{prop}}|\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle = + |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

$$- |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

$$+ |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

$$- |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

$$- |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

$$- |\cdots \otimes \otimes |\square\square||\circ\circ||\circ\circ\cdots\rangle$$

with the first 2 terms coming from (3.26), connected with rule 2 for moving the $\square$ particle, the second 2 terms coming from (3.29), connected with backward application of rule 6 for making $\square$
disappear. These 4 terms are exactly what we would like. However, we also obtain the three transitions to locally detectable states on the 5-7th lines, due to (3.36)-(3.38), connected with rule 3 for moving a qubit. The way the legal and locally detectable terms are produced by \( H_{\text{prop}} \) in our construction obeys certain rules.

We can check that \( H_{\text{prop}} \) acting on a state \( |\psi_t\rangle \) with a legal configuration (i.e. one appearing in a computation as in Table 3.2 with a correct number of qubits, properly aligned between boundaries) produces a superposition \( H_{\text{prop}} |\psi_t\rangle \) that contains

1. The original legal state with amplitude 2 (for two rules that apply to it), except for \( t = 0 \) and \( t = K \), where the amplitude is 1 (only a single rule applies to those two states).

2. Two new legal configurations: one due to a ‘forward’ transition to \( |\psi_{t+1}\rangle \) and one due to a ‘backward’ transition to \( |\psi_{t-1}\rangle \). Note that for \( t = 0 \) and \( t = K \) we only get one legal transition.

3. Some illegal terms, which are all locally detectable with \( H_{\text{pen}} \) (such as the 5th line in (3.41), with the illegal combination \( \circ \boxdot \circ \)).

Points 1 and 2 are a property of our construction with projector terms uniquely picking out only the “active” spots in a given configuration. We discuss the verification of point 3 (verifying that transition terms applied at inappropriate times are always locally detectable) in the next section.

Note that there exist allowed (not containing one of the forbidden pairs) configurations that are not locally detectable (such as the configuration \( C_4 \) in (3.40) with a single qubit). These are either improperly aligned or have an incorrect number of qubits. For some of these states, \( H_{\text{prop}} \) gives only one transition to another allowed state, while it still projects onto the state twice – this will be used to show the energy of such locally undetectable states is still high. For example the configuration from (3.40) is projected twice by the terms in \( H_{\text{prop}} \) corresponding to rules 3 and (the backwards application of) 6.

**Lemma 12.** A configuration that does not contain any of the forbidden pairs (i.e. those penalized by \( H_{\text{pen}} \)) is either one of the legal configurations (configurations that occur in the course of a computation), or (i) has a [qubits] string of incorrect length, or (ii) has a [qubits] string of the right length, but improperly aligned with the block boundaries.
Proof. Careful checking of the allowed pairs at positions \((i, i + 1)\) for odd \(i\) and even \(i\) from Table 3.5 implies the allowed joining of symbol pairs given in Table 3.3. This in turn restricts the legal/allowed configurations to form \((x \cdots x)[\text{qubits}](u \cdots u)\) where [\text{qubits}] is a nonzero string with the structure (3.5)-(3.13).

The only configurations that are not ruled out by these considerations are: (i) the legal configurations, (ii) configurations with a [\text{qubits}] string of the wrong length (not containing exactly \(n\) qubits), (iii) configurations with a [\text{qubits}] string with the right number of qubits, but improperly aligned with the blocks (e.g. \(\cdots \otimes |\underline{\text{□□□}}\otimes\underline{\text{□□□}}\otimes\underline{\text{□〇〇}}\otimes\cdots\) which eventually evolves to \(\cdots \otimes |\underline{\text{□□□}}\otimes\underline{\text{□□□}}\otimes\underline{\text{□〇〇}}\otimes\cdots\) with a bad pair \(|\underline{\text{□〇〇}}\) indicative of a misaligned block). \(\square\)

3.5 Completeness

Suppose there exists a witness, \(|\xi\rangle\), that is accepted by \(V_x\) with probability \(\geq 1 - \epsilon\). Beginning with the initial state \(|\psi_0\rangle\) (3.4) that has \(n\) qubits with qubit content \(|0^{n-m}\rangle \otimes |\xi\rangle\), we get the history state \(|\eta\rangle = \frac{1}{\sqrt{K+1}} \sum_{t=0}^{K} |\psi_t\rangle\) associated with circuit \(\tilde{V}_x\). The configurations occurring in this superposition are exactly the legal configurations. Given that all the ancilla qubits were initially in the \(|0\rangle\) state, \(H_{\text{in}}\) evaluates to zero on \(|\eta\rangle\). Since all the configurations in the superposition are legal, \(H_{\text{pen}}\) also evaluates to zero.

We next note the following facts about the legal configurations to be used in the claim.

Fact 13. Any legal configuration can contain at most one substring on the lefthand side of the transition rules (3.5) This means that to any legal configuration, at most one of the transition rules can apply in the forward direction. Furthermore, a legal configuration can contain at most one substring \(XY_{(j,j+1)}\) (and thus be connected to a single projector term of the type \(|XY\rangle \langle XY|_{j,j+1}\) in all of \(H_{\text{prop}}^{\text{rule}}\) (3.19)).

The first part of this fact can be verified by inspection of the list of legal configurations in Table 3.2 and the transitions that can be applied to them in Table 3.1. To check the second part (about the projector terms), in Table 3.5 we list the substrings \(XY\) identifying active spots in legal configurations from the projector terms of the type \(|XY\rangle \langle XY|_{j,j+1}\) in all of the \(H_{\text{prop}}^{\text{rule}}\). An inspection of the legal sequence again shows that each state in it has only one spot where one of the substrings \(XY\) appears (at a proper location with respect to the boundaries).
Table 3.6: Substrings identifying active spots in legal configurations. We list all of the substrings appearing in the projector terms of the type $|XY\rangle\langle XY|_{j,j+1}$ and $|ZW\rangle\langle ZW|_{k,k+1}$ from all of the $H_{\text{rule}}^{\rho}$ terms (3.24), (3.26), (3.34), (3.27), (3.28) and (3.29). Finding a substring $XY$ of a legal configuration $C_t$ uniquely indicates that the configuration is connected to some configuration $C_{t+1}$ ahead of it. Similarly, finding a substring $ZW$ uniquely locates a backward transition to some $C_{t-1}$.

**Fact 14.** Any legal configuration can contain at most one substring from the righthand side of the transition rules (i.e. at most one transition rule applies to it in the backward direction). Furthermore, a legal configuration can contain at most one substring $ZW_{(k,k+1)}$ (and thus be connected to a single projector term of the type $|ZW\rangle\langle ZW|_{k,k+1}$ in all of $H_{\text{prop}}^{\rho}$ (3.19)).

**Fact 15.** For a legal configuration $C_t$, there can be multiple places containing one of the substrings $NO_{(i,i+1)}$ or $PQ_{(i,i+1)}$ from all of the terms in $H_{\text{rule}}^{\rho}$ (3.19). However, exchanging any $NO \rightarrow PQ$ in $C_t$ leads to locally detectable illegal configurations for all cases except one, which gives the legal (following) configuration $C_{t+1}$. Similarly, exchanging any $PQ \rightarrow NO$ in $C_t$ leads to locally detectable illegal configurations for all cases except one, which gives the legal (preceding) configuration $C_{t-1}$.

We have chosen the $PQ$’s and $NO$’s so that they work properly where they should, and always produce locally detectable illegal configurations when used at “wrong times” (i.e. improper locations). This can be checked by careful inspection of the transition rules and terms in $H_{\text{prop}}^{\rho}$.

**Claim 16.** For any history state $|\eta\rangle$ with an initial configuration $C_0$ from (3.4), $\langle \eta | H_{\text{prop}} | \eta \rangle = 0$.

**Proof.** Let us see what happens when we apply $H_{\text{prop}}$ to a state $|\psi_t\rangle$ with a legal configuration $C_t$. The propagation Hamiltonian is made from terms of the type (3.19), with projection terms built from substrings $XY$ and $ZW$, and transition terms exchanging substrings $NO$ for $PQ$ and vice versa.

First, due to Fact 13, a legal configuration $C_t$ contains only one substring $XY_{(j,j+1)}$, and this projection term will apply, producing $|\psi_t\rangle$. Second, due to Fact 15, the corresponding transition
term \( NO_{i,i+1} \to PQ_{i,i+1} \) will apply, producing the state \(-|\psi_{t+1}\rangle\). Possibly, other transition terms for other substrings \( NO \to PQ \) will apply at a different (or the same) location, but these all produce locally detectable configurations, which are orthogonal to legal ones. Third, due to Fact 14 we get a single projection term because of the unique substring \( ZW_{(k,k+1)} \), producing \(|\psi_t\rangle \) again. Fourth, again due to Fact 15, we get a single legal transition to \(-|\psi_{t-1}\rangle\), and possible other, illegal, detectable states.

In sum, the action of \( H_{\text{prop}} \) on \(|\psi_t\rangle \) (for \( 1 \leq t \leq K - 1 \)) produces, \(-|\psi_{t-1}\rangle + 2|\psi_t\rangle - |\psi_{t+1}\rangle \) in the legal subspace, and a vector that lies in the space of locally detectable illegal configurations. For the endpoint states, we only get \( H_{\text{prop}} |\psi_0\rangle = |\psi_0\rangle - |\psi_1\rangle + \) illegal states, and \( H_{\text{prop}} |\psi_K\rangle = |\psi_K\rangle - |\psi_{K-1}\rangle + \) illegal states. Observe that within the legal subspace, both the rows and columns of the matrix form of \( H_{\text{prop}} \) sum to zero. Looking now at a history state \(|\eta\rangle\), a uniform superposition of legal states (for a given initial state), we obtain \( \langle \eta | H_t | \eta \rangle = 0 \), since \( H_{\text{prop}} |\eta\rangle = 0 \) when restricted to the legal subspace, and the illegal terms produced by \( H_{\text{prop}} \) are orthogonal to \(|\eta\rangle\).

With the propagation Hamiltonian evaluating to zero on a proper history state, we have

\[
\langle \eta | J_{\text{in}} H_{\text{in}} + J_{\text{prop}} H_{\text{prop}} + J_{\text{pen}} H_{\text{pen}} | \eta \rangle = 0.
\]

Consider now the last remaining term in the Hamiltonian, \( \langle \eta | H_{\text{out}} | \eta \rangle \). Since \( H_{\text{out}} \) acts only on the state with \( \Box \) on the last qubit, this term equals \( \frac{1}{K+1} \langle \psi_K | H_{\text{out}} | \psi_K \rangle = \frac{p_0}{K+1} \), where \( p_0 \) is the probability that the output qubit is zero in the final step. Since the computation accepts with probability \( 1 - p_0 \geq 1 - \epsilon \), we have \( \langle \eta | H_{\text{out}} | \eta \rangle \leq \frac{\epsilon}{K+1} \). Finally,

\[
\langle \eta | H | \eta \rangle = \langle \eta | J_{\text{in}} H_{\text{in}} + J_{\text{prop}} H_{\text{prop}} + J_{\text{pen}} H_{\text{pen}} + H_{\text{out}} | \eta \rangle \leq \frac{\epsilon}{K+1}.
\]  (3.42)

This concludes our completeness proof, showing that the energy for a proper history state corresponding to the verifying computation on a well-accepted witness \(|\xi\rangle\) is close to zero. Therefore, the ground state energy of \( H \) is also small, in fact it is upper-bounded by \( a = \frac{\epsilon}{K+1} \), where \( K = (R - 1)(3n^2 + 2n - 1) + 2n \), with \( R \) a polynomial in \( n \).
3.6 Soundness

We now need to show that in the case that there exists no witness that is accepted by $V_x$ with probability greater than $\epsilon$, the ground state energy of $H$ is bounded away from zero.

We partition the set of configurations into minimal sets $S$ that are invariant under the action of $H_{\text{prop}}$. The invariant sets are of three types:

1. Sets that contain legal configurations and locally detectable illegal configurations.

2. Sets that contain only locally detectable illegal configurations.

3. Sets that contain only illegal configurations, some of which are not locally detectable.

As we have seen previously, the action of $H_{\text{prop}}$ on a legal configuration produces legal ‘forward’ and ‘backward’ transitions, besides transitions to illegal configurations.

Illegal configurations that are not locally detectable either have the wrong number of qubits or have the right number of qubits but are incorrectly aligned\footnote{An example of a locally undetectable, badly aligned sequence: $\cdots \otimes \otimes \otimes \otimes \otimes \otimes \otimes \otimes \otimes \cdots$. The block length is 4, and when the $\vec{a}$ particle eventually reaches the front of the [qubits], it will happen away from a sequence boundary – there we’ll be able to detect it.} with the blocks. Since the transition rules do not change the number of two-state sites in a configuration nor change the alignment of the [qubits] string, legal configurations cannot transition to illegal configurations that are not locally detectable. Similarly, illegal configurations that are not locally detectable can only turn into other non-locally detectable illegal configurations, or into locally detectable illegal configurations.

A vector belonging to a subspace of type 2 has energy $\geq J_{\text{pen}}$, due to the presence of at least one locally detectable illegal pair that is penalized by $H_{\text{pen}}$. We now need to show that vectors from spaces of type 1 and 3 have high energy. We do this by repeated use of the Projection Lemma, a technique introduced in \cite{KKR06}. The lemma allows us to bound the ground state energy of our Hamiltonian by restricting it to progressively smaller subspaces of the Hilbert space.

**Lemma 17** (Projection Lemma, \cite{KKR06} Lemma 1). Let $H = H_1 + H_2$ be the sum of two Hamiltonians operating on some Hilbert space $\mathcal{H} = S + S^\perp$. Suppose the Hamiltonian $H_2$ is such that $S$ is a zero eigenspace and the eigenvectors in $S^\perp$ have eigenvalue at least $J > 2\|H_1\|$. Then,

\[
\lambda(H_1|S) - \frac{2\|H_1\|^2}{J - 2\|H_1\|} \leq \lambda(H) \leq \lambda(H_1|S) \tag{3.43}
\]
where $\lambda(A)$ denotes the lowest eigenvalue of an operator $A$.

### 3.6.1 Type 1 subspace

We consider the action of $H$ on $\mathcal{H}_1$, the space spanned by configurations of type 1. We apply the projection lemma with

$$H_1 = J_{\text{in}}H_{\text{in}} + J_{\text{prop}}H_{\text{prop}} + H_{\text{out}}, \quad H_2 = J_{\text{pen}}H_{\text{pen}}.$$  

Let $S_{\text{pen}}$ be the subspace of $\mathcal{H}_1$ that is spanned by legal configurations. Then $S_{\text{pen}} \subseteq \mathcal{H}_1$ is the zero eigenspace of $H_2$. On $S_{\text{pen}}^\perp$, $H_2$ has energy $\geq J_{\text{pen}}$. Since $\|H_1\| \leq \text{poly}(n)$, we can pick $J_{\text{pen}}$ to be some polynomial such that $J_{\text{pen}} > 2\|H_1\|$, allowing us to apply the projection lemma:

$$\lambda(H) \geq \lambda(H_1|_{S_{\text{pen}}}) - 1/8. \quad (3.44)$$

Now we bound the lowest eigenvalue of $H_1|_{S_{\text{pen}}}$,

$$H_1|_{S_{\text{pen}}} = H_{\text{out}}|_{S_{\text{pen}}} + J_{\text{in}}H_{\text{in}}|_{S_{\text{pen}}} + J_{\text{prop}}H_{\text{prop}}|_{S_{\text{pen}}}.$$  

We apply the projection lemma again, with

$$H'_1 = H_{\text{out}}|_{S_{\text{pen}}} + J_{\text{in}}H_{\text{in}}|_{S_{\text{pen}}}, \quad H'_2 = J_{\text{prop}}H_{\text{prop}}|_{S_{\text{pen}}}.$$  

To simplify the analysis of the eigenvalues of $H_{\text{prop}}$, we rotate to a different basis – one in which all the gates from $V_x$ are just the identity operator. We define the unitary matrix $W$:

$$W = \sum_{t=0}^{K} U_t \cdots U_1 \otimes |t\rangle\langle t|$$

where $|t\rangle$ represents the configuration in the $t$-th step of the computation, and $U_t, \ldots, U_1$ are the
first \( t \) unitary operations performed on the qubit content of the particles. Then we have,

\[
W^\dagger H_{\text{prop}}|_{S_{\text{pen}}} W = I \otimes 
\begin{bmatrix}
\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\
-\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 \\
0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \ddots & \\
0 & 0 & -\frac{1}{2} & \ddots & \ddots & : & \\
\vdots & \ddots & 1 & -\frac{1}{2} & \ddots & \\
0 & 0 & \cdots & 0 & -\frac{1}{2} & 1 & \\
0 & 0 & \cdots & 0 & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}_{(K+1) \times (K+1)}.
\]

\[ \text{(3.45)} \]

This matrix has only one zero-eigenvector, namely the valid history state. Therefore \( S_{\text{prop}} \subset S_{\text{pen}} \subset H_1 \) the set of correct history states (disregarding initial state of ancilla qubits). This matrix has second largest eigenvalue \( \geq \frac{1}{2(K+1)^2} \) (see Appendix A.1). Therefore, in \( S_{\text{prop}}^\perp \), \( H_2' \) has minimum energy \( \geq J_{\text{prop}} \cdot \frac{1}{2(K+1)^2} \). Choosing \( J_{\text{prop}} \) so that \( J_{\text{prop}}^2 > 2\|H_1'\| \), the projection lemma gives us:

\[
\lambda(H_1|_{S_{\text{pen}}}) \geq \lambda(H_1'|_{S_{\text{prop}}}) - \frac{1}{8}.
\]

\[ \text{(3.46)} \]

We now apply the projection lemma a third time, to \( H_1'|_{S_{\text{prop}}} \), with

\[
H_1'' = H_{\text{out}}|_{S_{\text{prop}}}, \quad H_2'' = J_{\text{in}} H_{\text{in}}|_{S_{\text{prop}}},
\]

\( H_2'' \) has zero eigenspace \( S_{\text{in}} \subset S_{\text{prop}} \subset S_{\text{pen}} \subset H_1 \), the set of history states with correctly initialized ancilla qubits. Since \( H_2'' \) is in the standard basis and applies to vectors that are history states with 0 on the ancilla input (i.e., in \( S_{\text{in}}^\perp \)), \( H_2'' \) has minimum energy \( O\left(\frac{1}{K+1}\right) \). Therefore, \( J_{\text{in}} \) can be chosen so that \( \frac{J_{\text{in}}}{K+1} > 2\|H_1''\| \). Then

\[
\lambda(H_1'|_{S_{\text{prop}}}) \geq \lambda(H_1''|_{S_{\text{in}}}) - \frac{1}{8},
\]

\[ \text{(3.47)} \]

and \( H_1''|_{S_{\text{in}}} = H_{\text{out}}|_{S_{\text{in}}} \). For any input state \( |\xi,0\rangle \), the circuit \( V_x \) accepts with probability \( < \epsilon \). Therefore, for any \( |\eta\rangle \in S_{\text{in}} \), we have \( \langle \eta|H_{\text{out}}|\eta\rangle > (1 - \epsilon)/(K + 1) \). In particular, this is true for the eigenvector \( |\eta\rangle \) of \( H_{\text{out}} \) with the lowest eigenvalue. Therefore \( \lambda(H_{\text{out}}|_{S_{\text{in}}}) \geq (1 - \epsilon)/(K + 1) \).
Combining (3.47) with (3.44) and (3.46), we have $\lambda(H) \geq \frac{5}{8} - \epsilon$.

Now we look at vectors from subspaces of type 3.

### 3.6.2 Type 3 subspace

A locally undetectable illegal configuration either (i) has the wrong number of qubits, or (ii) has the [qubits] string incorrectly aligned with the blocks.

Consider a (locally undetectable illegal) configuration with a ▶ site. The [qubits] string either crosses a block boundary, or is too short (or both). The ▶ moves right until it either hits the end of the [qubits] string or it hits a block boundary. If the end of the [qubits] string does not coincide with a block boundary, the configuration eventually evolves to contain either ▶⊙ or ▶□, both of which are locally detectable. If the end of the [qubits] string does coincide with the block boundary (this can only happen when the [qubits] string is too short), the qubits get moved over into the next block, where a ▶ is generated again, and, moving right, eventually produces a pair ▶□, which is locally penalized.

A locally undetectable illegal configuration with a ⊙ also eventually evolves into a locally detectable one: the ⊙ moves the [qubits] string to the right until the beginning of the [qubits] string coincides with the beginning of a block, and generates a ▶, at which point the above argument applies.

If our locally undetectable illegal configuration has neither ▶ or ⊙, i.e. its [qubits] substring consists only of ▶s (separated by ⊙s), the qubits begin to move themselves to the right, eventually generating a ⊙ or ▶ flag, at which point, the previous arguments apply: the evolution does indeed result in a locally detectable configuration.

In all the above cases, a locally detectable illegal configuration is reached within polynomially many steps/transitions. To see this, consider a configuration with $n'$ qubits. It takes $\text{poly}(n')$ steps to move the [qubits] string over one block, and by the preceding arguments, a locally checkable configuration must be reached at some point in this ‘round’ of computation. Since $n'$ can be at most $2nR$, this number of steps (which we label $K'$) must be polynomial in $n$. In other words, the transition rules eventually take the state outside $\mathcal{H}_3 = \text{Span}(\text{configurations of type 3})$. We can treat the restriction of $H_{\text{prop}}$ to $\mathcal{H}_3$ in much the same way as we did its restriction to $\mathcal{H}_1$.

We attempt to bound the lowest eigenvalue of $J_{\text{pen}}H_{\text{pen}} + J_{\text{prop}}H_{\text{prop}}$ on $\mathcal{H}_3$ using the projection
lemma, with $H_1 = J_{\text{prop}}H_{\text{prop}}$ and $H_2 = J_{\text{pen}}H_{\text{pen}}$. The zero eigenspace of $H_2$ is the space of illegal states that are not locally detectable, $S_{\text{pen}} \subset \mathcal{H}_3$. $H_2$ has energy $\geq J_{\text{pen}}$ on $S_{\text{pen}}^\perp$. Choosing $J_{\text{pen}}$ to be $\text{poly}(J_{\text{prop}}\|H_{\text{prop}}\|)$,

$$\lambda(H) \geq \lambda(H_1|S_{\text{pen}}) - \frac{1}{8}. \quad (3.48)$$

We now need the lowest eigenvalue of $J_{\text{prop}}H_{\text{prop}}|S_{\text{pen}}$. We rotate bases once again, using the unitary matrix $W$ defined earlier, with the difference that $|t\rangle$ now represents the $t$-th configuration in the sequence of (locally undetectable) illegal configurations that arises from the transition rules and forms the steps of the ‘computation’. This sequence of configurations terminates in a locally detectable illegal configuration on at least one end. (The other end of the sequence could be a locally undetectable illegal configuration from which there are no further transitions.) When we have a transition to a locally illegal configuration, the action of $H_{\text{prop}}$ on the last nonlocally detectable illegal configuration ($|K'\rangle$) in the sequence is to pick this configuration out twice, i.e., there are two projectors onto this configuration in $H_{\text{prop}}|S_{\text{pen}}$, with the result that, in the space of configurations, the last (or first, or both) diagonal element of $W^\dagger H_{\text{prop}}|S_{\text{pen}}W$ is 1 instead of $\frac{1}{2}$.

In other words, $W^\dagger H_{\text{prop}}|S_{\text{pen}}W$ looks like

\[
I \otimes \begin{bmatrix}
  f & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\
  -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 \\
  0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \cdots & \cdots & \cdots \\
  0 & 0 & -\frac{1}{2} & \cdots & \cdots & \cdots & 0 \\
  \vdots & \vdots & \vdots & \iddots & \cdots & \cdots & 0 \\
  0 & 0 & \cdots & 1 & -\frac{1}{2} \\
  0 & 0 & \cdots & 0 & -\frac{1}{2} & g & \ldots & \ldots \\
\end{bmatrix}_{(K'+1) \times (K'+1)}
\]

(3.49)

with $f = 1, g = \frac{1}{2}$ or $f = g = 1$. This is a matrix for a quantum walk on a line with particular boundary conditions.

The least eigenvalue of either of these matrices is $O\left(\frac{1}{(K'+1)^2}\right)$ (see Appendix A.1). Since $K'$ is a polynomial in $n$, choosing $J_{\text{prop}}$ to be an appropriately large polynomial in $n$, we can lower-bound the energy of $H$ on type-3 subspaces by some constant.
Chapter 4

The Q-2-SAT problem on 1 qutrit and \( n-1 \) qubits

The Quantum SAT problem is a version of the local Hamiltonian problem where all the Hamiltonian terms are projectors. If all the particles involved are qubits, Quantum SAT has an efficient algorithm. In this chapter, we consider the two-local Quantum SAT problem on a system of \( n-1 \) qubits and a single qutrit, and show that some special cases of this problem can be efficiently solved.

4.1 Introduction

In the satisfiability problem (SAT) a boolean formula \( \phi(x_1,\ldots,x_n) \) is given and the task is to decide whether or not \( \phi \) has a satisfying assignment, that is, an assignment to the variables \( x_1,\ldots,x_n \in \{0,1\} \) such that the formula evaluates to 1. The complexity of this problem depends on the form of \( \phi \). In the \( k \)-SAT problem, the formulas are in conjunctive normal form where each clause has \( k \) literals. Conjunctive normal form is the AND of a set of clauses where each clause is an OR of literals \( (\gamma_{i1} \lor \cdots \lor \gamma_{ik}) \). There is a transition in complexity from \( k = 2 \) to \( k = 3 \): the 3-SAT problem is NP-complete while 2-SAT is in P. In other words, there is an efficient algorithm to solve 2-SAT while we do not expect there to be one for 3-SAT. The constraint satisfaction problem (CSP) is a generalization of SAT.

In a CSP, the variables can take more than just two values, and the constraints on a given set of variables are specified by the subset of values that those variables are allowed to take. A
well-known example of a CSP is the graph-coloring problem: given a graph $G$ with a set of edges $V = \{v_i\}$, and a set of edges $E = \{(v_j, v_k)\}$, and a set of $k$ labels or ‘colors’, can we label each vertex in the graph such that no two vertices that share an edge are given the same color. In the case of 3-Coloring, the variables are the nodes $v_i$ of the graph to be colored and can take any of three values (the three colors, say R, G, B), and the constraint corresponding to each edge is the set of values that the pair of nodes sharing that edge can not take: ((R, R), (G, G), (B, B)).

A quantum analog of SAT (Q-$k$-SAT) has been defined where an assignment to the variables now holds a quantum state and the clauses are projectors acting on the variables. The variables are $d$-dimensional vector spaces, and can be thought of as $d$-level quantum systems. In Q-$k$-SAT each projector acts on $k$ qubits. The analog of satisfying a clause classically is that the projector applied to the state vector evaluates to zero. Given an instance, which is a set of $k$-local projectors, the question is whether there is one state vector satisfies all the projectors. Bravyi showed that Q-2-SAT has an efficient classical algorithm, as it does in the classical case. Q-3-SAT is NP-complete since it contains 3-SAT as a special case.

Here we study the generalization to quantum CSPs. We are interested in understanding what the complexity is when some number of particles is three level instead of two. If too many particles are three level, then the problem is NP-hard. This can be seen by a reduction from 3-COLORING. For each edge $(u, v)$ in the original graph, a gadget replaces it where three two level nodes are inserted in the middle, and each node has an edge to $u$ and $v$. The constraints on the six new edges enforce that $u$ and $v$ have different colors by labeling the color with a one it the corresponding position.

If only one particle is three level, then the classical CSP is efficiently solvable. It is possible to assign a value to the three level node and get an instance that reduces to 2-SAT. This can be tried for each of the three possible values $\{0, 1, 2\}$ to get the answer. We ask the same question for the quantum CSP. The problem seems to be more difficult since we cannot just try all possible states in $\mathbb{C}^3$. We consider the case with only a single qutrit and $n - 1$ qubits.
4.2 Background

Bravyi [Bra06] defines the Quantum-2-SAT problem and gives an efficient algorithm for it. The input to the problem is a set of two-qubit constraints given by projectors. The algorithm uses the idea that an instance of Quantum-2-SAT on \( n \) qubits can often be reduced to an instance on \( n - 1 \) qubits. This process of reduction has a nice interpretation in terms of penalized subspaces and degrees of freedom. A constraint is a projector \( \{\Pi_{ab}\} \) that acts on \( \mathbb{C}_a^2 \otimes \mathbb{C}_b^2 \). Any vector in this space, under the action of the constraint, is either preserved (an eigenvector of eigenvalue 1) or gets zeroed out (a zero eigenvector). The eigenspace of eigenvalue 1 can be thought of as incurring an energy penalty under the Hamiltonian. The ground state of the system must have a component in the unpenalized subspace of \( \mathbb{C}_a^2 \otimes \mathbb{C}_b^2 \). The rank of the projector determines the dimension of this unpenalized subspace: a projector of rank 4 penalizes the whole Hilbert space of the two qubits and can not have a zero-energy ground state; a projector of rank \( 4 - d \) has an unpenalized subspace of dimension \( d \).

This allows us to restrict our search for a zero-energy ground state to the unpenalized subspaces alone. We can thus reduce the size of our search space. The particular structure of Quantum-2-SAT means that this reduction of the search space corresponds to a smaller instance of Quantum-2-SAT. Since rank-4 projectors do not permit a zero-eigenvector, let us look at how we can ‘reduce’ the size of the instance if we have rank-3 or rank-2 projectors. A rank-3 projector on qubits \( a \) and \( b \) penalizes all but a single vector of the two-qubit space. Any satisfying assignment must be a tensor product of this vector with some state of the remaining \( n - 2 \) qubits. We then need to consider only states on this smaller \( n - 2 \)-qubit space. The situation with a rank-2 projector is quite similar. In this case, the qubits \( a \) and \( b \) have a two-dimensional unpenalized space: an effective qubit. Having reduced the pair of qubits \( a \) and \( b \) to a new qubit \( c \), we have an \( n - 1 \)-qubit instance.

How do we deal with instances that have only rank-one constraints? For this, Bravyi introduces the idea of generating new constraints’. Given any trio of qubits, \( a, b, c \), and constraints \( \Pi_{ab} \) and \( \Pi_{bc} \), we can show that any state on these three qubits that satisfies these two constraints must also satisfy some constraint on the \( a \) and \( c \). This constraint can be constructed from \( \Pi_{ab} \) and \( \Pi_{bc} \). If this constructed constraint is not already specified as part of the problem description, we can add it to our set of constraints, thus generating a new constraint. We can do this for all trios of qubits. If
the addition of new constraints results in there being two or more independent constraints on some pair of qubits (i.e., a constraint of rank two or more), we can reduce the size of our instance. If we ever reach an instance containing a rank-4 constraint, we can declare the constraints unsatisfiable. Otherwise, we reach a point where each pair of qubits has only a rank-one constraint, and no new constraints can be generated. Such an instance is said to be ‘complete’. A satisfying tensor product assignment can always be constructed for every complete instance.

Let us now formally define the Quantum SAT problem on a system of \( n - 1 \) qubits and a single qutrit.

**Definition 18.** The input to the problem is a Hamiltonian \( H \) on a \( n \)-particle system consisting of \((n - 1)\) qubits and 1 qutrit, and a real number \( \epsilon \) that is \( \frac{1}{\text{poly}(n)} \). The Hamiltonian \( H \) is a sum of two-local projectors, \( H = \sum \Pi_{ab} \). We are asked to determine if there exists a zero-eigenstate of \( H \).

More precisely, we must decide between the cases:

1. There exists a state \( |\psi\rangle \) on \( (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^3 \) such that \( H |\psi\rangle = 0 \), or equivalently, \( \Pi_{ab} |\psi\rangle = 0 \), for every pair \( a, b \).

2. For every state \( |\psi\rangle \) on \( (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^3 \), \( \langle \psi | H |\psi\rangle > \epsilon \).

### 4.3 Eliminating higher-rank constraints

We denote the constraint on particles \( a, b \) by \( \Pi_{ab} \), where \( \Pi_{ab} \) is a projector. If the particles \( a \) and \( b \) have dimensions \( d_a \) and \( d_b \), \( \Pi_{ab} \) acts on a \( d_a d_b \)-dimensional space. \( \Pi_{ab} \) can have rank between 1 and \( d_a d_b \). In our case, we have either qubit-qubit constraints, which can have ranks between 1 and 4, or qutrit-qubit constraints, which can have ranks between 1 and 6. Since a \( \Pi_{ab} \) with rank \( d_a d_b \) is unsatisfiable (it penalizes the entire Hilbert space of the two qudits), we only consider instances where qubit-qubit constraints have rank at most 3, and qutrit-qubit constraints have rank at most 5.

We now see how we can reduce any instance of Q-(3,2)-SAT to an instance where all constraints (both qubit-qubit and qutrit-qubit) have rank at most 2, by getting rid of all higher-rank constraints.
4.3.1 Constraints of rank $d_ad_b - 1$

Claim 19. We can eliminate qubit-qubit constraints of rank 3 and qutrit-qubit constraints of rank 5.

We reproduce from Bravyi a property of these projectors that we can use to simplify some instances of our quantum SAT problem: Any two projectors $P$ and $Q$ obey the following equivalence:

\[ \{ P |\psi\rangle = |\psi\rangle, \quad Q |\psi\rangle = |\psi\rangle \} \Leftrightarrow \{ P |\psi\rangle = |\psi\rangle, \quad PQP |\psi\rangle = |\psi\rangle \} . \]

The uses of this property become evident when we consider constraint projectors $\Pi_{ab}$ and $\Pi_{cd}$ acting on pairs of particles $(a,b)$ and $(c,d)$ respectively. If $P = I - \Pi_{ab} = I - |\phi\rangle \langle \phi|_{ab}$ and $Q = I - \Pi_{cd}$ (note that $\Pi_{ab}$ is a projector of rank $d_ad_b - 1$), we have two possibilities:

1. If $(a,b)$ and $(c,d)$ do not overlap, then $\{ P |\psi\rangle = |\psi\rangle, \quad PQP |\psi\rangle = |\psi\rangle \}$ is equivalent to $\{ |\psi\rangle = |\phi\rangle_{ab} \otimes |\psi'\rangle, \quad \Pi_{cd} |\psi'\rangle = 0 \}$ where $|\psi'\rangle$ is a state on the remaining $n - 2$ particles.

2. If $(a,b)$ and $(c,d)$ overlap (say $a = c$), then $\{ P |\psi\rangle = |\psi\rangle, \quad PQP |\psi\rangle = |\psi\rangle \}$ is equivalent to $\{ |\psi\rangle = |\phi\rangle_{ab} \otimes |\psi'\rangle, \quad Q_d |\psi'\rangle = |\psi'\rangle \}$ where $|\psi'\rangle$ is a state on the remaining $n - 2$ particles as before, and $Q_d$ is a one-local constraint implicitly defined by $|\phi\rangle \langle \phi|_{ab} (I - \Pi_{ad}) |\phi\rangle \langle \phi|_{ab} = |\phi\rangle \langle \phi|_{ab} \otimes Q_d$.

What this means is that, in any state that satisfies all the constraints, the two particles $a$ and $b$ must be confined to a (one-dimensional) subspace of their joint Hilbert space. Once we restrict $a$ and $b$ to this subspace, we must update the constraints between $a$ and $b$ and the other particles. The expression for $Q_d$ tells us how to do this. For our specific problem (with only qubits and qutrits), this tells us how to handle qutrit-qubit constraints of rank 5 and qubit-qubit constraints of rank 3.

4.3.2 Constraints of rank $d_ad_b - d_a$

We now consider constraints of other ranks. These can be dealt with in a similar fashion: given a constraint $\Pi_{ab}$ of rank $r$, the unpenalized subspaces of the qudits $a$ and $b$ form an effective particle of dimension $d_ad_b - r$. Labeling this effective qudit $c$, we can derive the new constraints between $c$ and the other qudits.
Claim 20. We can eliminate qubit-qubit constraints of rank 2 and qutrit-qubit constraints of rank 3 or 4.

Let us consider this more general case in some more detail. A constraint projects onto a subspace of dimension \( r \) (\( r \) denotes the rank of the projector), which in our case is at most 6. Let \( \{ |u_1\rangle, \ldots, |u_r\rangle \} \) be some orthonormal basis of the forbidden subspace, and extend this to an orthonormal basis of the whole two-particle space. The constraint is then a projector \( |u_1\rangle \langle u_1| + \cdots + |u_r\rangle \langle u_r| \).

Let \( V \) be defined as the partial isometry described by a matrix with columns corresponding to the basis vector spanning the unpenalized space (\( |u_{r+1}\rangle, \ldots, |u_{d_a d_b}\rangle \)) \( (V : \mathbb{C}^{d_a d_b - r} \rightarrow \mathbb{C}^{d_a} \otimes \mathbb{C}^{d_b}) \), then \( V V^\dagger = |u_{r+1}\rangle \langle u_{r+1}| + \cdots + |u_{d_a d_b}\rangle \langle u_{d_a d_b}| \) is a projector on to the unpenalized subspace, which we label \( c \). Then the constraint-projector can be written \( \Pi_{ab} = I - V_c V_c^\dagger \). Now, for any constraint on some pair of qudits \( (f,g) \), we have the following:

Claim 21. \( \Pi_{fg} |\psi\rangle = 0 \) iff \( V_c^\dagger \Pi_{fg} V_c |\psi'\rangle = 0 \).

Proof. \( 0 = \Pi_{fg} |\psi\rangle = \Pi_{fg} V_c |\psi'\rangle \). Therefore, \( V_c^\dagger \Pi_{fg} V_c |\psi'\rangle = 0 \).

In the other direction, we have \( V_c^\dagger \Pi_{fg} V_c |\psi'\rangle = 0 \). Taking the inner product of this vector with \( |\psi'\rangle \), we get \( \langle \psi'| V_c^\dagger \Pi_{fg} V_c |\psi'\rangle = 0 \). Since \( |\psi\rangle = V_c |\psi'\rangle \), this is the same as saying \( \langle \psi| \Pi_{fg} |\psi\rangle = 0 \). In other words, the inner product of the vector \( \Pi_{fg} |\psi\rangle \) with itself is zero. Therefore, \( \Pi_{fg} |\psi\rangle = 0 \). \( \square \)

Any satisfying state \( |\psi\rangle \) must now be of the form \( |\psi\rangle = V_c |\psi'\rangle \) where \( |\psi'\rangle \) is a state on a smaller space, obtained by tensoring the space \( c \) with the spaces of the other \( n - 2 \) particles. The constraints that must be satisfied by this new, smaller, state are given by \( V_c^\dagger \Pi_{fg} V_c \) where \( \{ \Pi_{fg} \} \) are the constraints in the original instance. We thus have a ‘smaller’ instance.

This smaller instance is not always useful for our purposes. For instance, if we consider qutrit-qubit constraints, this new space (which we can think of as a particle consisting of the unpenalized degrees of freedom of \( a \) and \( b \)) will be of dimension \( d_a d_b - r = 6 - r \), where \( r \) is the rank of the constraint. In general, the new set of constraints only remains an instance of our particular problem if the new particle has dimension \( d_c \) equal to 1, 2, or 3. However, this idea is useful insofar as it allows us to ‘eliminate’ qutrit-qubit constraints of ranks 4 and 3, reducing the instance to one on a smaller number of particles. Similarly, we can apply this claim to eliminate qubit-qubit constraints of rank 2.
4.4 Constraints of rank 1 and 2

Applying the ideas in the previous section repeatedly, we reach either an unsatisfiable instance, or an instance where all qubit-qubit constraints are of rank 1, and all qutrit-qubit constraints are of rank at most 2.

We begin by disposing of an easy special case:

**Lemma 22.** Consider a qutrit-qubit constraint of rank 2, described by specifying two vectors \(|\phi_1\rangle\) and \(|\phi_2\rangle\) in \(\mathbb{C}^3 \otimes \mathbb{C}^2\). If \(|\phi_1\rangle\) and \(|\phi_2\rangle\) have the special form \(|\phi_1\rangle = |u_1\rangle \otimes |v_1\rangle\) and \(|\phi_2\rangle = |u_1\rangle \otimes |v_2\rangle\), where \(|u_1\rangle \in \mathbb{C}^3\) and \(|v_1\rangle, |v_2\rangle \in \mathbb{C}^2\), then we can reduce this instance to an all-qubit instance.

**Proof.** Since we assume in general that a constraint is specified by \(r\) orthogonal vectors, let us assume that \(|v_1\rangle\) and \(|v_2\rangle\) are orthonormal in \(\mathbb{C}^2\). Fixing two vectors \(|u_2\rangle\) and \(|u_3\rangle\) such that \(\{|u_1\rangle, |u_2\rangle, |u_3\rangle\}\) is an orthonormal basis for \(\mathbb{C}^3, |u_1\rangle |v_1\rangle, |u_2\rangle |v_1\rangle, |u_1\rangle |v_2\rangle, |u_2\rangle |v_2\rangle, |u_3\rangle |v_1\rangle, |u_3\rangle |v_2\rangle\) is a basis for the two-particle space. The projector on to the unpenalized space is

\[
I_6 - \Pi_{ab} = |u_2\rangle \langle u_2| \otimes |v_1\rangle \langle v_1| + |u_2\rangle \langle u_2| \otimes |v_2\rangle \langle v_2| + |u_3\rangle \langle u_3| \otimes |v_1\rangle \langle v_1| + |u_3\rangle \langle u_3| \otimes |v_2\rangle \langle v_2| .
\]

Since \(|v_1\rangle\) and \(|v_2\rangle\) are orthonormal vectors in \(\mathbb{C}^2\), we have

\[
I_6 - \Pi_{ab} = (|u_2\rangle \langle u_2| + |u_3\rangle \langle u_3|) \otimes I_2 .
\]

4.4.1 Generating new constraints

Bravyi deals with rank-1 projectors by introducing the idea of ‘completeness’ of a set of constraints. A set of constraints is said to be ‘complete’ when no new constraints can be generated from the existing constraints. If we start with an instance where all constraints are rank-1 and attempt to add new constraints on every edge, we end up either with a complete set (where all the constraints are rank-1 and no new ones can be generated) or with an edge where the constraint has rank greater than 1, in which case we can reduce the size of the instance using the techniques outlined in the previous subsections. We hope to define an analogous notion of completeness for our problem.

A constraint is described by giving the vectors forming a basis for the forbidden subspace. A rank-1 constraint thus corresponds to a single vector \(\sum_{\alpha=0}^2 \sum_{\beta=0}^1 c_{\alpha\beta} |\alpha\rangle_a |\beta\rangle_b\) (a rank-2 constraint can be described by two orthogonal vectors, and can be thought of as two separate rank-1 constraints). A rank-1 constraint can be written as a tensor of order two \(\phi_{\alpha\beta}^{(a,b)}\), where \(\phi_{\alpha\beta}^{(a,b)} = c_{\alpha\beta}\).

**Fact 23.** A vector \(|\psi\rangle\) is orthogonal to two vectors \(|\phi_1\rangle\) and \(|\phi_2\rangle\) if and only if it is orthogonal to any pair of vectors in the span of \(|\phi_1\rangle\) and \(|\phi_2\rangle\).
Now consider a three particle subsystem (particles labelled \(a\), \(b\), and \(c\)), with constraints (written in tensor form) \(\phi^{(a,b)}\) and \(\theta^{(b,c)}\). Bravyi shows that any state \(\psi\) on these particles that satisfies these constraints (i.e., \(\phi_{\alpha\beta}^{(a,b)} \psi_{\alpha\beta\gamma} = 0\) and \(\theta_{\beta\gamma}^{(b,c)} \psi_{\alpha\beta\gamma} = 0\)) must also satisfy the constraint \(\omega_{\alpha\gamma}^{(a,c)} = \phi_{\alpha\beta}^{(a,b)} \epsilon_{\beta\mu} \theta_{\mu\gamma}^{(b,c)}\), where \(\epsilon\) is the fully-antisymmetric tensor of order 2. Note that, for this definition of \(\omega\) to make sense, the index corresponding to the second particle must take two values, i.e., particle \(b\) must be a qubit. Particles \(a\) and \(c\) can be qubits or qutrits or even higher-dimensional particles.

We now have a way of deriving \((a,c)\) constraints, given \((a,b)\) and \((b,c)\) constraints. If, using this method, we succeed in generating a new constraint on some pair of particles \((a,b)\) that increases its rank beyond 1 (for qubit-qubit constraints) or beyond 2 (for qutrit-qubit constraints), we can once again apply the procedures outlined in the previous section to reach either an unsatisfiable instance or an instance where all ranks are once again at most 2.

Let us attempt to define a complete set of constraints as follows: A complete set is one in which (i) all qubit-qubit constraints have rank 1, (ii) all qutrit-qubit constraints have rank at most 2, (iii) for all triples of qubits, no new constraints can be defined, and (iv) for all qutrit-qubit-qubit triples, no new qutrit-qubit constraints can be defined. Note that this definition does not say anything about new qubit-qubit constraints in qutrit-qubit-qubit triples.

### 4.4.2 Complete set with all constraints of rank one

This case has a satisfying tensor-product assignment (the qutrit is labelled \(ctr\) below):

1. Assign arbitrarily to the qutrit, say \(\psi^{(ctr)}\).

2. Assign to any one of its qubit neighbors: \(\psi^{(a)} = \epsilon (\phi^{(ctr,a)})^T \psi^{(ctr)}\). It is easy to check that \(\psi^{(ctr)} \psi^{(a)}\) satisfies the constraint \(\phi^{(ctr,a)}\).

3. Assign to all qubits \(b\) that are neighbors of both \(ctr\) and \(a\) as: \(\psi^{(b)} = \epsilon (\phi^{(a,b)})^T \psi^{(a)}\). Once again, it is easy to check that the \(\phi^{(a,b)}\) constraint is satisfied. Moreover, the \((ctr, b)\) constraint is also satisfied: since the three-particle state \(\psi^{(ctr)} \psi^{(a)} \psi^{(b)}\) satisfies \(\phi^{(ctr,a)}\) and \(\phi^{(a,b)}\), the completeness condition assures us that \(\phi^{(ctr,b)}\) must also be satisfied by this state.

4. Repeat steps 2 and 3 until all neighbors of the center qutrit have been assigned a state.
5. We have now assigned to everything in the neighborhood of the qutrit (call this set of particles $V_{close}$). The rest of the particles $V_{far}$ are as yet unassigned. Bravyi’s paper shows that any assignment to $V_{far}$ satisfies the edges between $V_{close}$ and $V_{far}$. The edges in $V_{far}$ again form a complete set of constraints, so the have a tensor product solution.

4.4.3 Complete set with at least one constraint of rank 2

Exactly one rank-2 constraint

This case again has a satisfying tensor-product assignment. It is slightly different from the previous case because we can no longer assign arbitrarily to the center qutrit. Say the rank-2 constraint is between the qutrit $ctr$ and a qubit $A$. We first show that there is a tensor product assignment to these two particles that satisfies the rank-2 constraint. The rank-2 constraint is described by two orthonormal vectors $p = (p_1, p_2, p_3, p_4, p_5, p_6)^T$ and $q = (q_1, q_2, q_3, q_4, q_5, q_6)^T$. We need to find vectors $(a, b, c)^T$ and $(c, d)^T$, such that

\[
\begin{align*}
    p_1 ad + p_2 ae + p_3 bd + p_4 be + p_5 cd + p_6 ce &= 0 \\
    q_1 ad + q_2 ae + q_3 bd + q_4 be + q_5 cd + q_6 ce &= 0.
\end{align*}
\]

We can rewrite this as:

\[
\begin{align*}
    d(p_1 a + p_3 b + p_5 c) + e(p_2 a + p_4 b + p_6 c) &= 0 \\
    d(q_1 a + q_3 b + q_5 c) + e(q_2 a + q_4 b + q_6 c) &= 0.
\end{align*}
\]

If we had values for $a$, $b$, and $c$, we could solve these two equations for $c$ and $d$. We wish to assign values to $a$, $b$, $c$ such that these equations have a non-trivial solution for $c$ and $d$. A vector $(a, b, c)$ obeying the following constraints will fit the bill:

\[
\begin{align*}
    (p_1 - q_1)a + (p_3 - q_3)b + (p_5 - q_5)c &= 0 \\
    (p_2 - q_2)a + (p_4 - q_4)b + (p_6 - q_6)c &= 0.
\end{align*}
\]
But this is a system of two equations in three variables, thus always has a solution. So if we pick some particular solution \((a, b, c)\), we can plug in to the previous set of equations and solve for \((d, e)\), to get a tensor product assignment to qutrit \(ctr\) and qubit \(A\). From here on, the assignment procedure is similar to the one in the previous subsubsection.

**Complete set with more than one qutrit-qubit constraint of rank two**

Here we assume that the instance has been fully reduced: all remaining qubit-qubit constraints are of rank 1, all remaining qutrit-qubit constraints are of rank at most 2, and any possible bridge constraints have been generated and added to the instance. Now we show that any such instance with at most two rank-2 qutrit-qubit constraints has a tensor product solution.

**Claim 24.** Consider a fully reduced instance with exactly two rank-2 constraints between the center qutrit (denoted \(ctr\)) and some pair of qubits \(A\) and \(B\). Such an instance, if satisfiable, always has a tensor product solution.

**Complete set with more than one qutrit-qubit constraint of rank two**

Here we assume that the instance has been fully reduced: all remaining qubit-qubit constraints are of rank 1, all remaining qutrit-qubit constraints are of rank at most 2, and any possible bridge constraints have been generated and added to the instance. Now we show that any such instance with at most two rank-2 qutrit-qubit constraints has a tensor product solution.

**Claim 25.** Consider a fully reduced instance with exactly two rank-2 constraints between the center qutrit (denoted \(ctr\)) and some pair of qubits \(A\) and \(B\). Such an instance, if satisfiable, always has a tensor product solution.

Call every qutrit-qubit edge of rank 2 a ‘petal’. Consider the subgraph formed by the qubits that have a rank-2 constraint with the qutrit. The edges of this subgraph are the rank-1 constraints between the qubits. Each connected component of this subgraph is termed a ‘cluster’. If our instance has only one cluster, we can assign a satisfying tensor-product state thus: Assign to the qutrit \((ctr)\) and one of the qubits \((A)\) as in the previous subsubsection, and to qubits \(B\) that neighbor both \(ctr\) and \(A\) according to \(\psi^B = \epsilon(\phi^{A,B})^T \psi^A\). By completeness, the \((ctr, B)\) constraint (whether of rank 1 or 2) is also satisfied.
Assuming that we can solve three particle instances with just two rank-2 constraints, we have a way of finding a satisfying tensor product assignment for all instances with at most two clusters.

**Lemma 26.** There exists a satisfying tensor product assignment for every instance with at most two clusters.

**Proof.** If our instance has only one cluster, we begin the assignment by picking an arbitrary petal $\Pi_{ctr,A}$ from the cluster, and finding a tensor product assignment to the center qutrit and the qubit $A$. We can extend this assignment to the rest of the cluster as follows: to every qubit $B$ that neighbors both the qutrit and some already-assigned qubit $A'$, we assign the state $\psi(B) = \epsilon(\phi(A',B))T\psi(A')$ (where $\phi_{A',B}$ is the rank-1 constraint between $A'$ and $B$). This procedure is repeated until every qubit in the cluster has been assigned.

If the instance has two clusters, the assignment procedure is very similar. We begin by picking two arbitrary petals, one from each cluster, and finding a satisfying tensor product assignment for the resulting two-petal three-particle instance. Given a tensor product assignment to the center qutrit and one of the qubits in the cluster, the assignment can be extended to every other qubit in the cluster. \qed

Now we must show that three particle instances with just two rank-2 constraints do indeed have tensor product solutions. Suppose the two rank-2 constraints are given by pairs of vectors $(p,q)$ and $(r,s)$. A satisfying tensor product assignment is a vector of the form $(a,b,c)_{ctr}^T \otimes (d,e)_A^T \otimes (f,g)_B^T$.

\[
\begin{align*}
 p_{1a}d + p_{2a}e + p_{3b}d + p_{4b}e + p_{5c}d + p_{6c}e &= 0 \\
 q_{1a}d + q_{2a}e + q_{3b}d + q_{4b}e + q_{5c}d + q_{6c}e &= 0 \\
 r_{1a}f + r_{2a}g + r_{3b}f + r_{4b}g + r_{5c}f + r_{6c}g &= 0 \\
 s_{1a}f + s_{2a}g + s_{3b}f + s_{4b}g + s_{5c}f + s_{6c}g &= 0.
\end{align*}
\]

Separating the odd and even components of the constraint vectors ($p_{odd} = (p_1, p_3, p_5)$, $p_{even} = (p_2, p_4, p_6)$, etc.) and denoting by $v$ the vector $(a, b, c)$, we can rewrite the constraint equations...
more compactly as:

\[(p_{\text{odd}} \cdot v)d + (p_{\text{even}} \cdot v)e = 0\]
\[(q_{\text{odd}} \cdot v)d + (q_{\text{even}} \cdot v)e = 0\]
\[(r_{\text{odd}} \cdot v)f + (r_{\text{even}} \cdot v)g = 0\]
\[(s_{\text{odd}} \cdot v)f + (s_{\text{even}} \cdot v)g = 0\]

We thus have two sets of equations: a pair of linear equations in the variables \(d\) and \(e\), and a pair of linear equations in the variables \(f\) and \(g\). We need non-zero vectors \((d, e)\) and \((f, g)\) that will satisfy these equations. The only way that the pair of equations in \(d\) and \(e\) (similarly for the equations in \(f\) and \(g\)) can have non-zero solution is if the two equations are not linearly independent, i.e., if the corresponding determinants are zero:

\[
\begin{vmatrix}
    p_{\text{odd}} \cdot v & p_{\text{even}} \cdot v \\
    q_{\text{odd}} \cdot v & q_{\text{even}} \cdot v \\
    r_{\text{odd}} \cdot v & r_{\text{even}} \cdot v \\
    s_{\text{odd}} \cdot v & s_{\text{even}} \cdot v
\end{vmatrix} = 0.
\]

Each of these equations gives us a quadratic equation in the three variables \(a, b, c\). For example, the first determinant is \((p_1q_2 - p_2q_1)a^2 + (p_3q_4 - p_4q_3)b^2 + (p_5q_6 - p_6q_5)c^2 + (p_1q_4 + p_3q_2 - p_2q_3 - p_4q_1)ab + (p_3q_6 + p_5q_4 - p_4q_5 - p_6q_3)bc + (p_1q_6 + p_5q_2 - p_2q_5 - p_6q_1)ca\).

Does such a pair of quadratic equations always have a non-zero solution (the all-zero vector \(v\) is always a solution)? To answer this, we turn to the theory of Gröbner bases. Computing the Gröbner bases for a set of polynomials gives us information about the number of common zeroes of the polynomials. In particular, it can tell us whether the set of equations has a finite number of zeroes or infinitely many. Using the computer algebra software, Mathematica, to compute the Gröbner basis for a pair of polynomials of the form \(x_1a^2 + x_2b^2 + x_3c^2 + x_4ab + x_5bc + x_6ca, y_1a^2 + y_2b^2 + y_3c^2 + y_4ab + y_5bc + y_6ca\), we obtain a set of polynomials. The theory of Gröbner bases tells us that a set of polynomials with variables \(\{z_\alpha\}_\alpha\) has a finite number of zeroes if and only if, for each variable \(z_\alpha\), there is some polynomial in the Gröbner basis whose initial monomial...
(the initial monomial is well-defined because the Gröbner bases computation uses some ordering on monomials) is a power of $z_\alpha$ (not involving any other variable $z_\beta$). Since this is not true for the Gröbner basis of our pair of quadratic polynomials \([B.2]\), we can conclude that the set of equations has infinitely many zeroes, and in particular, has a non-zero solution.

Given the existence of a satisfying non-zero \((a, b, c)\), we know that there are also corresponding non-zero vectors \((d, e)\) and \((f, g)\) such that the two-petal case has a satisfying tensor product assignment. Since any such assignment can be extended to a tensor product assignment for a complete instance with two clusters, we may conclude that any complete two-cluster instance has a tensor product solution.
Chapter 5

Discussion and Open Problems

In this dissertation, we have explored the 2-local Hamiltonian problem along two directions. Our first result showed the hardness of estimating the ground energy of a 1D system, while the second result showed an efficient algorithm for special instances of the Quantum 2SAT problem on a system that included one qutrit.

5.1 1D 2-local Hamiltonians

The one-dimensional version of the 2-local Hamiltonian problem is important because it is the simplest possible topology for such a system. By constraining our system interactions to nearest-neighbor on a line, we are looking at a very restricted version of the Local Hamiltonian problem. Since the Local Hamiltonian problem has been shown to be QMA-complete even with locality 2, to search for a tractable version of the problem, we need to look at more constrained versions. However, AGIK09’s proof of the QMA-completeness of the 1D problem with 13-state qudits showed that merely restricting the problem to 1D is not enough to make it tractable.

However, the complexity of the 1D problem could depend on the dimensionality of the particles. In particular, it could be the case that decreasing the dimensionality makes the problem less hard. The effective heuristics used by physicists to handle systems of qubits on a line suggest that perhaps the problem is tractable for the case of 2-state particles. As a step towards settling this question, we showed that the dimensionality of the particles could be brought down to 8 without affecting the complexity: the problem remains QMA-complete.
Most of the Local Hamiltonian QMA-completeness proofs to date, including ours, use a common set of tools. A QMA verifier circuit is encoded into a Hamiltonian that has the effect of checking the computation of this verifier. In 1D, this amounts to implementing a locally checkable computation on a line of qudits. One approach to reducing the dimensionality of qudits is to use increasingly efficient encodings of the circuit. We accomplish this by eliminating some of the states from the 13-state construction (at the cost of complicating the bounds on the Hamiltonian’s ground energy) and by using the parity of a qudit as an additional ‘state’ (while doubling the number of qudits required). It remains to be seen if more efficient encodings can bring down the qudit-dimensionality even further. At the other end of the complexity spectrum, it could happen that the 1D problem has an efficient algorithm (classical or quantum) for some qudit dimensionality between 7 and 2, inclusive. The complexity of the 1D problem for qudits of dimensionality less than 8 remains an open problem. Besides the obvious relevance to quantum complexity, i.e., exploring the ‘boundary’ cases of the Local Hamiltonian problem, investigating these cases is also motivated by the fact that systems of interest to physicists tend to be consist of qudits of lower dimensionality.

### 5.2 Quantum SAT

Quantum SAT can be thought of as either a specialization of the Local Hamiltonian problem, or as a generalization of the classical SAT problem. Though the Quantum 2SAT problem on qubits is efficiently decideable, Quantum 2SAT on a system of both qubits and qutrits is NP-hard. This follows from the observation that 3-Coloring can be reduced to classical SAT on a system of 2-level and 3-level (classical) variables. However, a classical system of \( n \) two-level variables and \( \log(n) \) three-level variables is still efficiently decideable: we can simply enumerate all possible assignments to the three-level variables in polynomial time. The question is, then, how does this generalize to the quantum version? We attempted to address this issue by examining the case where the system had only a constant number of qutrits. In particular, can introducing just a single qutrit to a system of qubits change the complexity of quantum SAT?

We showed that special instances of the problem can be efficiently decided. We could deal with any instance that involved the qutrit in at most two rank-2 constraints. While some instances with three such ‘petals’ were clearly unsatisfiable, computation showed other instances did have a
solution. However, a completely general solution evaded us. One question that we were unable to answer, and that might have helped towards a general solution was: Can qubit-qubit constraints be ‘generated’ from two qutrit-qubit constraints? This question is further complicated by the fact that we are considering systems where the qutrit-qubit constraints can be of rank 2. Answering this question could also throw light on how the number of petals in a satisfiable instance is related to the presence of such ‘cross-linking’ qubit-qubit constraints, and possibly lead to an upper bound on the number of petal clusters possible in a satisfiable instance, thereby limiting the number of cases that need to be analysed.

A fully general solution, if achieved, would still leave open a few questions: Can such an algorithm be adapted to instances with a constant (> 1) number of qutrits? How about if we replace the single qutrit with a higher-dimensional qudit? Furthermore, how many qubits can be replaced by qutrits/qudits before the complexity of the problem changes?

5.3 Other Directions

While we have so far looked at the problem of determining the ground energy of a Local Hamiltonian system, the field of Hamiltonian Complexity has opened up other types of problems. An immediately related problem is that of describing ground states of local Hamiltonian systems. A fundamental fact about quantum systems is that their state can not, in general, be efficiently described: the state of an $n$-qubit system is a $2^n$-dimensional vector. However, it was shown that the ground state of 2-local Commuting Hamiltonians admits an efficient description [BY05].

Another result related to the description of ground states is the 1D Area Law [Has07, AALV11]. Area laws place bounds on the entanglement entropy of ground states, and while they have been conjectured for arbitrary dimension, only the 1D case has been proved. Characterizing the classes of local Hamiltonians that have an efficiently describable ground state is important because such systems hold out hope of being tractable for analysis and computation.

Another problem that has been gaining attention in the quantum complexity world is the Quantum PCP conjecture. The classical PCP theorem offers a new characterization of NP, and says that many optimization problems that are hard to solve, such as MAX-3SAT, are also hard to approximate. In the quantum world, we have seen that it is QMA-hard to estimate the energy
of a $k$-local Hamiltonian with a $1/poly$ promise gap. A quantum PCP theorem would give a way of efficiently mapping such a $k$-local Hamiltonian instance to another $k$-local Hamiltonian with a constant promise gap.
Bibliography


Appendix A

8-state particles on a line

A.1 Eigenvalues

Here we analyze the eigenvalues of the three matrices in Section 3.6. Our matrices are of the form:

\[
\begin{bmatrix}
  f & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\
  -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 \\
  0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \ddots & & \\
  0 & 0 & -\frac{1}{2} & \ddots & \ddots & \vdots & \\
  \vdots & & \ddots & \ddots & 0 & \\
  0 & 0 & \cdots & 1 & -\frac{1}{2} & \\
  0 & 0 & \cdots & 0 & -\frac{1}{2} & g
\end{bmatrix}_{(L+1)\times(L+1)}
\]  

(A.1)

where, in subspaces of type 1, \( f = g = \frac{1}{2} \), and in subspaces of type 3, either (i) \( f = g = 1 \) or (ii) \( f = 1 \) and \( g = \frac{1}{2} \) (we could also have \( f = \frac{1}{2} \) and \( g = 1 \), but this doesn’t change the eigenvalues).

We wish to solve the eigenvalue equation \( Mx = \lambda x \), where \( M \) is our matrix, and \( x = (x_0, x_1, x_2, \ldots, x_L)^T \).

It is easy to see that \( x \) must satisfy the equations

\[-\frac{1}{2}x_{j-1} + x_j - \frac{1}{2}x_{j+1} = \lambda x_j \quad \text{for } 1 \leq j \leq L - 1.
\]  

(A.2)
We use the ansatz
\[ x_j = A \cos k(j + c) + B \sin k(j + c). \]  \hspace{1cm} (A.3)
where \( A, B, k \) and \( c \) are reals. Plugging this into (A.2):
\[
-(\cos k) (A \cos k(j + c) + B \sin k(j + c)) = (\lambda - 1) (A \cos k(j + c) + B \sin k(j + c))
\]
\[
\lambda = 1 - \cos k
\]

Any vector \( x \) with \( x_j = A \cos k(j + c) + B \sin k(j + c) \) satisfies (A.2) with \( \lambda = 1 - \cos k \). Now we apply the ‘boundary conditions’ in the first case \( (f = g = \frac{1}{2}) \). The eigenvectors and eigenvalues are
\[
\lambda_m = 1 - \cos \left( \frac{m\pi}{L + 1} \right),
\]
\[
|\psi_m\rangle = \sum_{j=0}^{L} \cos \left( \frac{m\pi}{L + 1} (j + \frac{1}{2}) \right) |j\rangle \quad \text{for} \; m = 0, 1, \ldots, L.
\]

We can check this by plugging into the boundary condition equations
\[
\frac{1}{2} x_0 - \frac{1}{2} x_1 = (1 - \cos k)x_0,
\]
\[
-\frac{1}{2} x_{L-1} + \frac{1}{2} x_L = (1 - \cos k)x_L.
\]
The lowest eigenvalue in this case is 0 (when \( m = 0 \)). The second-lowest eigenvalue is
\[
1 - \cos \left( \frac{\pi}{L + 1} \right) > \left( \frac{1}{L + 1} \right)^2 \left( \frac{\pi^2}{2!} - \frac{\pi^4}{4!(L + 1)^2} \right)
\]
\[
= \Omega \left( \frac{1}{(L + 1)^2} \right)
\]
i.e., \( 1/poly(L) \), where \( L \) is the number of steps, as promised.

Now we consider the next set of boundary conditions \( f = g = 1 \) (subspace of type 3). We get
eigenvalues and eigenvectors:

$$\lambda_m = 1 - \cos \left( \frac{(m + 1)\pi}{L + 1} \right),$$

$$|\psi_m\rangle = \sum_{j=0}^{L} \cos \left( \frac{(m + 1)\pi}{L + 2} (j + 1) \right) |j\rangle$$

for $m = 0, 1, \ldots, L$.

This is again easily checked by plugging into the equations

$$x_0 - \frac{1}{2} x_1 = (1 - \cos k)x_0,$$

$$-\frac{1}{2} x_{L-1} + x_L = (1 - \cos k)x_L.$$

The lowest eigenvalue here is $\lambda_0 = 1 - \cos \left( \frac{\pi}{L+2} \right)$, which is $\Omega \left( \frac{1}{(L+2)^2} \right)$.

The final set of boundary conditions to consider is $f = 1, g = \frac{1}{2}$. The eigenvalues and eigenvectors in this case are:

$$\lambda_m = 1 - \cos \left( \frac{(2m + 1)\pi}{2L + 3} \right),$$

$$|\psi_m\rangle = \sum_{j=0}^{L} \sin \left( \frac{(2m + 1)\pi}{2L + 3} (j + 1) \right) |j\rangle$$

for $m = 0, 1, \ldots, L$.

The lowest eigenvalue here is $\lambda_0 = 1 - \cos \left( \frac{\pi}{2L+3} \right) = \Omega \left( \frac{1}{(2L+3)^2} \right)$. 
Appendix B

Quantum SAT

B.1 Notes on the Bravyi paper

B.1.1 Equation 1 in Bravyi

Claims that for two projectors $P$ and $Q$, \{ $P \ket{\psi} = \ket{\psi}$, $Q \ket{\psi} = \ket{\psi}$ \} $\Leftrightarrow$ \{ $P \ket{\psi} = \ket{\psi}$, $PQP \ket{\psi} = \ket{\psi}$ \}.

Proof. $PQP \ket{\psi} = P(Q \ket{\psi}) = \ket{\psi}$. But $P \ket{\psi} = \ket{\psi}$. Therefore, we must have $Q \ket{\psi} = \ket{\psi}$.

\hfill $\square$

B.1.2 Rank 2 constraints

Projector $\Pi_{ab}$ on qubits $a$ and $b$ has rank 2. The qubits $a$ and $b$ can then be rolled into a single qubit which we shall denote $c$. Extend the two vectors projected on by the constraint $\Pi_{ab}$ to a basis of the two-qubit space $\mathbb{C}^2 \otimes \mathbb{C}^2$. Let $V$ be the 4-by-2 matrix with columns corresponding to the unpenalized vectors. The projector on to the unpenalized subspace is then $VV^\dagger$, and $\Pi_{ab}$ can be written as $I - VV^\dagger$. Also note that $V^\dagger V = I$. Any satisfying state $\ket{\psi}$ must be of the form $V_c \ket{\psi'}$.

The constraints $\Pi_{fg}$ must now be updated. The new constraints pop out of the following claim:

Claim 27. $\Pi_{fg} \ket{\psi} = 0$ iff $V_c^\dagger \Pi_{fg} V_c \ket{\psi'} = 0$.

Proof. $0 = \Pi_{fg} \ket{\psi} = \Pi_{fg} V_c \ket{\psi'}$. Therefore, $V_c^\dagger \Pi_{fg} V_c \ket{\psi'} = 0$.  

78
In the other direction, we have \( V_c^\dagger \Pi_f g V_c |\psi'\rangle = 0 \). Taking the inner product of this vector with \(|\psi\rangle\), we get \( \langle \psi | V_c^\dagger \Pi_f g V_c |\psi'\rangle = 0 \). Since \(|\psi\rangle = V_c |\psi'\rangle\), this is the same as saying \( \langle \psi | \Pi_f g |\psi\rangle = 0 \). In other words, the inner product of the vector \( \Pi_f g |\psi\rangle \) with itself is zero. Therefore, \( \Pi_f g |\psi\rangle = 0 \). □

The new constraint on qubits \( c \) and \( g \) is thus \( Q_{cg} |\psi'\rangle = 0 \), where \( Q_{cg} = V_c^\dagger \Pi_a g V_c \), or more explicitly: \( Q_{cg} = (V_c^\dagger \otimes I_g)(\Pi_a g \otimes I_b)(V_c \otimes I_g) \)

**B.2 Gröbner bases**

Computing a Gröbner basis for a set of polynomials can be thought of as a generalization of Gaussian elimination. Given a field \( K \), we have the corresponding ring of polynomials in \( n \) variables over the field, \( K[x_1, \ldots, x_n] \). Now consider a set of polynomials \( F \) over \( K \): this set generates an ideal \( \langle F \rangle \).

A term order \( \prec \) on \( K[x_1, \ldots, x_n] \) is an ordering imposed on the set of all monomials. If we fix a term order, each polynomial has a unique initial term. The set of initial terms in some ideal \( I \) in \( K[x_1, \ldots, x_n] \) generate an ideal of their own, called the initial ideal, which we denote \( in_\prec(I) \).

A Gröbner basis of the ideal \( I \) is a subset \( G \) of \( I \) whose initial terms generate the initial ideal of \( I \). The Gröbner basis for some set of polynomials \( F \) is useful because it has the same set of solutions as \( F \), and moreover, reveals properties of the set of solutions that are not readily seen from \( F \). In particular, the Gröbner basis enables us to count the number of solutions: the set of solutions is finite if and only if every unknown \( x_i \) appears to some power in \( in_\prec(I) \).

The Gröbner basis for the set of polynomials \( \{ \alpha_1 a^2 + \alpha_2 b^2 + \alpha_3 c^2 + \alpha_4 ab + \alpha_5 bc + \alpha_6 ca, \beta_1 a^2 + \beta_2 b^2 + \beta_3 c^2 + \beta_4 ab + \beta_5 bc + \beta_6 ca \} \) in variables \( a, b, c \) under the degree reverse lexicographic ordering is \( \{ \alpha_1 a^2 + \alpha_2 b^2 + \alpha_3 c^2 + \alpha_4 ab + \alpha_5 bc + \alpha_6 ca, \beta_1 a^2 + \beta_2 b^2 + \beta_3 c^2 + \beta_4 ab + \beta_5 bc + \beta_6 ca, \alpha_2 \beta_1 b^2 + \alpha_3 \beta_1 c^2 + \alpha_4 \beta_1 ab + \alpha_5 \beta_1 bc + \alpha_6 \beta_1 ac - \alpha_1 \beta_2 b^2 - \alpha_1 \beta_3 c^2 - \alpha_1 \beta_4 ab - \alpha_1 \beta_5 bc - \alpha_1 \beta_6 ac \} \). Clearly, no power of the variable \( c \) appears as the first term of any of the Gröbner basis polynomials, indicating that this pair of quadratic equations has an infinite set of solutions.
Vita
Sandeep Narayanaswami

Education

- The Pennsylvania State University, PhD in Physics, 2013
- Indian Institute of Technology Madras, Bachelor of Technology in Engineering Physics, 2008

Experience

- Research Assistant, Pennsylvania State University, Spring 2010 - Summer 2013
- Teaching Assistant, Pennsylvania State University, Fall 2008 - Summer 2010

Published Work

Sean Hallgren, Daniel Nagaj, and Sandeep Narayanaswami. The local Hamiltonian problem on a line with eight states is QMA-complete. *Quantum Information and Computation*, Vol.13 No.9 & 10 (pp0721-0750), September 2013.