Undecidability of the Spectral Gap

J.B. Houben

Supervisor: G.L.M. Cornelissen

Utrecht University

July 28, 2016

Abstract

We treat the results of the 2015 paper by Cubitt et al. entitled "Undecidability of the Spectral Gap", and the constructions required to arrive at these results. This paper showed that a specific quantum mechanical problem (the "spectral gap problem") is algorithmically undecidable, using techniques from computer science. We also treat the theory from quantum mechanics and computer science that is required to understand these results and constructions.

Contents

Chapter 1

Introduction

In a 2015 paper by Cubitt et al., entitled "Undecidability of the Spectral Gap" ([CPW15b]), it was mathematically proven that a specific problem of interest in quantum mechanics (the "spectral gap problem") is algorithmically undecidable, using techniques from computer science. A short version of this paper was published in Nature ([CPW15a]). The purpose of this thesis is to make the results of this paper accessible to mathematics undergraduate students. We attempt to do this by introducing the theory necessary to understand the results and the required constructions, and by explicitly showing these constructions and their motivations. This thesis was written in the first half of 2016 as a bachelor thesis for the mathematics and physics programmes at Utrecht University, under the supervision of prof. dr. Gunther Cornelissen, the head of the department of mathematics.

In chapter 2 we will treat the concepts and notation from quantum mechanics which are used in this thesis. Then we formally introduce the problem in chapter 3. Next, we prove a simplified version of the result, and show how this result differs from the stronger result, and why this is relevant. In the next chapter, we describe the problems that arise when trying to prove the stronger result, and give a rough overview of how these problems will be overcome. Then, in chapter 6, we introduce the concepts from computer science which will be necessary to understand the remainder of this thesis. This chapter also allows us to more rigorously state some things which had been intuitively described in earlier chapters. Then, in chapters 7 to 9, we describe the ingredients of the construction which allows us to prove our main result. Finally, these results will be stated in chapter 10.

Chapter 2

A brief introduction to quantum mechanics

In this chapter we will treat the concepts and notation from quantum mechanics which are used in this thesis but may be unfamiliar to a reader without a background in physics. The relevant concepts from computer science are treated in chapter 6.

The quantum mechanical concepts which are most central to this thesis are that of a *Hilbert space* and a *Hamiltonian*. We will first define Hilbert spaces. Then we will treat the bra-ket notation which will be used throughout this thesis. Next, we will treat the tensor product which will be useful in the construction of Hilbert spaces. We then move on to the concept of observables in quantum mechanics, which allows us to define the Hamiltonian. Finally, we give a physical example.

2.1 Hilbert spaces

Definition 1 (Hilbert space)**.** *A* Hilbert space $\mathscr{H}$ *is a real or complex inner product space that is also a complete metric space with respect to the metric induced by the inner product.*

We will only consider finite-dimensional complex Hilbert spaces. It is important to note that physicists use a slightly different definition of inner products than mathematicians: for physicists, the inner product is *antilinear* in its first argument and *linear* in its second argument, rather than the other way around. So, writing λ^* for the complex conjugate of λ,

$$\langle \lambda x, y \rangle = \lambda^* \langle x, y \rangle, \tag{2.1}$$

and

$$\langle x, \lambda y \rangle = \lambda \langle x, y \rangle, \tag{2.2}$$

for all $x, y \in \mathscr{H}$ and $\lambda \in \mathbb{C}$. This is also the convention that will be used in this thesis.

The possible states of a quantum mechanical system are represented by the unit vectors of the Hilbert space[1]. These unit vectors are referred to as "state vectors" or simply "states". Two unit vectors represent the same state if they differ only by a constant factor. This factor, necessarily having norm 1, is called a phase factor.

[1] There is some subtlety here: in the infinite-dimensional case, it is sometimes useful to consider non-normalizable states, even though they are not technically part of the Hilbert space. For example, it is often useful to consider functions of the form $\psi_p(x) = e^{ipx}$ when considering the Hilbert space of square-integrable-functions, even though such functions are not actually square-integrable themselves. In this case, the functions ψ_p are eigenfunctions of the momentum operator. However, since we will only consider finite-dimensional Hilbert spaces, this will not be relevant to our purposes.

We will sometimes use the term "superposition". When we speak of a superposition of states a and b, we mean any state $x \neq a, b$ which is a linear combination of a and b. Generally, a and b will be elements of some distinguished orthonormal basis for the Hilbert space.

2.2 Dirac's bra-ket notation

Throughout this thesis we will use the bra-ket notation introduced by Paul Dirac in 1939 ([Dir39]). In this notation, vectors are written as "kets":

$$|\psi\rangle \in \mathscr{H}. \tag{2.3}$$

Here, "ψ" is simply a label, not a mathematical object in itself. It is common to use Greek letters and integers as labels, but sometimes non-alphanumeric symbols or even words are used (see section 2.5 for an example of the former). When integers are used as labels, small Latin letters are often used to index these labels, such as in equation (2.13).

The inner product between a ket $|\varphi\rangle$ and a ket $|\psi\rangle$ will be written as

$$\langle\varphi|\psi\rangle\,. \tag{2.4}$$

From now on, we will use this notation unless there is a need to be explicit (about the space in which the inner product is computed, for instance). This notation for the inner product now motivates the introduction of the "bra" $\langle\varphi| \in \mathscr{H}^*$ to represent a covector ($\mathscr{H}^*$ denotes the dual space of $\mathscr{H}$):

$$\begin{aligned} \langle\varphi| : \mathscr{H} &\to \mathbb{C}, \\ |\psi\rangle &\mapsto \langle\varphi|\,|\psi\rangle = \langle\varphi|\psi\rangle\,. \end{aligned} \tag{2.5}$$

The bra $\langle\varphi|$ is commonly referred to as the adjoint (or Hermitian conjugate) of the ket $|\varphi\rangle$. This is motivated by the canonical isomorphism from $\mathscr{H}$ to the space $L(\mathbb{C}, \mathscr{H})$ of linear operators from $\mathbb{C}$ to $\mathscr{H}$:

$$\begin{aligned} \Phi : \mathscr{H} &\to L(\mathbb{C}, \mathscr{H}), \\ |\varphi\rangle &\mapsto (\lambda \mapsto \lambda\,|\varphi\rangle). \end{aligned} \tag{2.6}$$

Equipping $\mathbb{C} = \mathbb{C}^1$ with the standard inner product[2], and taking the adjoint of $\Phi(|\varphi\rangle)$ we obtain a map

$$\Phi(|\varphi\rangle)^\dagger : \mathscr{H} \to \mathbb{C}. \tag{2.7}$$

By the definition of the adjoint, this map satisfies

$$\big\langle \Phi(|\varphi\rangle)\lambda, |\psi\rangle \big\rangle_{\mathscr{H}} = \big\langle \lambda, \Phi(|\varphi\rangle)^\dagger\,|\psi\rangle \big\rangle_{\mathbb{C}}. \tag{2.8}$$

This simplifies to

$$\lambda^*\,\langle\varphi|\psi\rangle = \lambda^*\big(\Phi(|\varphi\rangle)^\dagger\,|\psi\rangle\big). \tag{2.9}$$

Setting λ to 1, we see that $\Phi(|\varphi\rangle)^\dagger$ is indeed $\langle\varphi|$. So, under the canonical isomorphism Φ, we have that $|\varphi\rangle^\dagger$ is $\langle\varphi|$, for any $|\varphi\rangle \in \mathscr{H}$.

2.2.1 Linear operators

A linear operator A naturally acts on kets, but can also act on bras in the following way:

$$\begin{aligned} \langle\varphi|\,A : \mathscr{H} &\to \mathbb{C}, \\ |\psi\rangle &\mapsto \big(\langle\varphi|\,A\big)\,|\psi\rangle = \langle\varphi|\,\big(A\,|\psi\rangle\big). \end{aligned} \tag{2.10}$$

[2] That is, $\langle x, y\rangle_{\mathbb{C}} = x^* y$ for $x, y \in \mathbb{C}$.

Note that linear operators act on kets on the right, but on bras on the left. The "associativity" apparent in the above equation allows us to simply write

$$\langle\varphi|A|\psi\rangle\,. \tag{2.11}$$

The bra-ket notation allows us to naturally write linear operators as outer products. For example, the operator $|\varphi\rangle\langle\psi|$ is defined as follows:

$$\begin{aligned} |\varphi\rangle\langle\psi| : \mathscr{H} &\to \mathscr{H}, \\ |\chi\rangle &\mapsto |\varphi\rangle\langle\psi|\;|\chi\rangle = \langle\psi|\chi\rangle\,|\varphi\rangle\,. \end{aligned} \tag{2.12}$$

All linear operators can be written this way: let A be a linear operator, and let $\{|1\rangle\,,\ldots,|n\rangle\}$ be an orthonormal basis for the space it acts on. It is easily checked that

$$A = \sum_{i,j=1}^{n} \langle i|A|j\rangle\,|i\rangle\langle j|\,. \tag{2.13}$$

We will also write

$$\mathbb{1} : V \to V \tag{2.14}$$

for the identity map on arbitrary vector spaces. Of course, this notation is not specific to the bra-ket notation.

2.3 Tensor product

The tensor product is of particular interest in quantum mechanics. It provides the mathematical mechanism for "combining" two quantum mechanical systems. To formally define the tensor product of Hilbert spaces, we first define the free vector space. We will restrict our attention to the finite-dimensional case, since this will be sufficient for our purposes.

Definition 2 (Free vector space)**.** *Let S be a finite set and K a field. The* free vector space $F(S,K)$ of S over K *is the vector space over the scalar field K "spanned by the elements of S". Formally, $F(S,K)$ is the space of functions from S to K*

$$F(S,K) = \{f : S \to K\}\,, \tag{2.15}$$

equipped with the natural sum and scalar product. That is, for all $f,g \in F(S,K)$ and $\lambda \in K$,

$$f + g = (s \mapsto f(s) + g(s)), \tag{2.16}$$

$$\lambda \cdot f = (s \mapsto \lambda \cdot f(s)). \tag{2.17}$$

We now identify each element $s \in S$ with the Dirac-delta function

$$\delta_s(x) = \begin{cases} 1_K, & \text{if } s = x, \\ 0_K, & \text{otherwise}, \end{cases} \tag{2.18}$$

where 1_K and 0_K are the multiplicative and additive identity of K respectively.

It is easily checked that $F(S,K)$ is indeed a vector space over K, which has $\{\delta_s\}_{s\in S}$ as a basis.

Definition 3 (Tensor product of vector spaces)**.** *Let V and W be finite-dimensional vector spaces over a field K. Let $F = F(V \times W, K)$. We now define an equivalence relation $\sim$ on F. For all $v, v' \in V$, $w, w' \in W$ and $\lambda \in K$,*

$$(v, w) + (v', w) \sim (v + v', w), \tag{2.19}$$
$$(v, w) + (v, w') \sim (v, w + w'), \tag{2.20}$$
$$\lambda(v, w) \sim (\lambda v, w), \tag{2.21}$$
$$\lambda(v, w) \sim (v, \lambda w). \tag{2.22}$$

Then $V \otimes W = F/\sim$. The sum and scalar product on $F/\sim$ are defined through representatives of the equivalence classes:

$$[u] + [u'] = [u + u'], \tag{2.23}$$
$$\lambda \cdot [u] = [\lambda \cdot u], \tag{2.24}$$

for any $[u], [u'] \in F/\sim$ and $\lambda \in K$. It is easily checked that these operations are indeed well-defined.

For ease of notation, we write $v \otimes w$ for the element $[(v, w)]$.

It is easily seen that tensor products are associative up to a canonical isomorphism. This allows us to unambiguously write expressions such as

$$U \otimes V \otimes W. \tag{2.25}$$

In some cases we will write $V^{\otimes n}$ as a shorthand for $\bigotimes_{i=1}^{n} V$.

Note that if $\mathcal{V}$ is a basis for a (finite-dimensional) vector space V, and $\mathcal{W}$ a basis for a vector space W, then

$$\{v \otimes w \,|\, v \in \mathcal{V}, w \in \mathcal{W}\} \tag{2.26}$$

is a basis for $V \otimes W$. In particular, $\dim(V \otimes W) = \dim(V) \cdot \dim(W)$.

Definition 4 (Tensor product of Hilbert spaces)**.** *Let $\mathscr{H}_1$ and $\mathscr{H}_2$ be finite-dimensional complex Hilbert spaces. Then the Hilbert space $\mathscr{H}_1 \otimes \mathscr{H}_2$ is defined by equipping the tensor product vector space as defined above with the unique inner product*[3] *satisfying*

$$\left\langle v \otimes w, v' \otimes w' \right\rangle_{\mathscr{H}_1 \otimes \mathscr{H}_2} = \langle v, v' \rangle_{\mathscr{H}_1} \cdot \langle w, w' \rangle_{\mathscr{H}_2}, \tag{2.27}$$

for all $v, v' \in \mathscr{H}_1$ and $w, w' \in \mathscr{H}_2$. Since $\mathscr{H}_1 \otimes \mathscr{H}_2$ is finite-dimensional, it immediately follows that it is complete, and thus indeed a Hilbert space.

2.3.1 Bra-ket notation

When using the bra-ket notation, elements of a tensor product can be written as

$$|\varphi\rangle \otimes |\psi\rangle. \tag{2.28}$$

In many cases, more compact notation will be utilized:

$$|\varphi\rangle \otimes |\psi\rangle = |\varphi\rangle\, |\psi\rangle = |\varphi, \psi\rangle. \tag{2.29}$$

Where confusion may arise, kets will be labelled with the space they live in. In this thesis we will use superscripts to refer to the space of a particular particle in an ensemble of particles, and subscripts to refer

[3] Obtained by extending equation (2.27) through linearity in the second argument and antilinearity in the first argument.

to spaces which have a particular purpose or describe a particular property of a particle. Furthermore, superscript labels will always be parenthesized to reduce possible confusion with powers.

$$|\varphi\rangle \otimes |\psi\rangle = |\varphi\rangle^{(1)} |\psi\rangle^{(2)} . \tag{2.30}$$

It is not uncommon to treat the tensor product as being commutative (which is of course true up to a canonical isomorphism). This allows us to write

$$|\varphi\rangle \otimes |\psi\rangle = |\varphi\rangle^{(1)} |\psi\rangle^{(2)} = |\psi\rangle^{(2)} |\varphi\rangle^{(1)} . \tag{2.31}$$

The bra corresponding to the composite ket $|\varphi\rangle \otimes |\psi\rangle$ is written as

$$\langle\varphi| \otimes \langle\psi| . \tag{2.32}$$

All of the flexibilities in how we write kets also apply to how we write bras, although the order of the labels (in this case, φ then ψ) is kept consistent between the kets and corresponding bras.

2.3.2 Tensor products of linear maps

It will be useful to be able to "lift" linear maps to product spaces. To this end, we define the tensor product of linear maps.

Definition 5 (Tensor product of linear maps)**.** *Let $A : X_1 \to Y_1$ and $B : X_2 \to Y_2$ be linear maps. Then $A \otimes B : X_1 \otimes X_2 \to Y_1 \otimes Y_2$ is defined by*

$$(A \otimes B)(x_1 \otimes x_2) = (Ax_1) \otimes (Bx_2), \tag{2.33}$$

and extending by linearity. We further identify A, when interpreted as a map on $X_1 \otimes X_2$ with

$$A \otimes \mathbb{1} : X_1 \otimes X_2 \to Y_1 \otimes X_2, \tag{2.34}$$

and B, when interpreted as a map on $X_1 \otimes X_2$ with

$$\mathbb{1} \otimes B : X_1 \otimes X_2 \to X_1 \otimes Y_2. \tag{2.35}$$

2.4 Observables

In quantum mechanics measurable (or "observable") quantities of a quantum system are given by specific linear operators. For instance, say we have a system which describes a particle (an electron, say). So we have some Hilbert space $\mathscr{H}$. Now suppose we are interested in the momentum of this particle. Then there is some linear, self-adjoint operator $P : \mathscr{H} \to \mathscr{H}$ that represents the momentum of this particle[4]. What does it mean for this operator to "represent" the momentum of the particle? Well if we denote the eigenvalues of P by $\{p_i\}$ and the corresponding normalized eigenkets by $\{|p_i\rangle\}$ then the following statements hold:

- Upon measuring the momentum of the particle we will obtain one of[5] the eigenvalues p_i of P, and after measurement the system will "collapse" into the corresponding eigenstate $|p_i\rangle$.
- Given some (normalized) state $|\varphi\rangle \in \mathscr{H}$, the probability of measuring momentum p_i when the system is in state $|\varphi\rangle$ is $|\langle p_i|\varphi\rangle|^2$.
- As a consequence of the previous statement, given some (normalized) state $|\varphi\rangle \in \mathscr{H}$, the expected value upon measuring the momentum when the system is in state $|\varphi\rangle$ is $\langle\varphi|P|\varphi\rangle$.

[4] In some cases, such an operator can be derived using the principles of quantum mechanics and some classical reasoning. In other cases, it can only be derived experimentally. Either way, all that matters to us is that such an operator exists.

[5] In quantum mechanics, measurements are probabilistic events.

Note that because the operators representing observables are self-adjoint, their eigenvalues are real, and there exists an orthonormal basis of eigenstates (for a proof of this statement see theorems 9 and 10 in chapter 1 of [Sha94], for example).

This finally allows us to define the Hamiltonian, a concept which will be absolutely central to this thesis.

Definition 6 (Hamiltonian)**.** *A* Hamiltonian *H is a self-adjoint linear operator on a Hilbert space $\mathscr{H}$, representing the total energy (kinetic and potential) of the system.*

In light of the above, the eigenvalues of (the operator representing) an observable are of great interest. This motivates the following definition:

Definition 7 (Spectrum)**.** *Given a linear operator $A : V \to V$ on a finite-dimensional vector space V, the* spectrum of A*, denoted* $\operatorname{spec} A$*, is the set of eigenvalues of A.*

In the case of the Hamiltonian, there is some additional relevant terminology.

Definition 8 (Ground state and ground state energy)**.** *Given some Hamiltonian, the smallest eigenvalue is called the* ground state energy*, and the corresponding eigenvectors (there may be multiple because of degeneracy) are called* ground states*.*

Definition 9 (Excited states and excitation energies)**.** *Given some Hamiltonian, all eigenvalues which are greater than the ground state energy are referred to as* excitation energies*. The corresponding eigenvectors are called* excited states*.*

Furthermore, we will use the terms "energies" and "eigenvalues" interchangeably when considering a Hamiltonian.

2.5 Physical example

We will now give an example of a physical system to demonstrate the concepts of this chapter. This example is based on the material in chapter 14 of [Sha94]. Consider a system consisting of two electrons in a uniform magnetic field. To keep things simple, we will ignore any interaction between the particles, as well as their movement through space. Instead, we will only focus on the interaction between the electrons and the magnetic field.

We will first focus on just one electron. Since we ignore the spatial degrees of freedom, we are left with just the particle's *spin*. Spin is a quantum mechanical property analogous to angular momentum. In particular, it leads to the electron having a magnetic moment and thus interacting with the magnetic field. The Hilbert space $\mathscr{H}$ of such a system is a two-dimensional complex vector space (with the standard inner product). There is a particular linear operator S_z corresponding to the observable quantity "spin along the z-axis". It has eigenstates $|\uparrow\rangle$ and $|\downarrow\rangle$ with corresponding eigenvalues $+\frac{\hbar}{2}$ and $-\frac{\hbar}{2}$ respectively. Here, $\hbar$ is the reduced Planck constant

$$\hbar = \frac{h}{2\pi} \approx 1.05 \times 10^{-34}\,\mathrm{J\,s}. \tag{2.36}$$

The states $|\uparrow\rangle$ and $|\downarrow\rangle$ form an orthonormal basis for the Hilbert space. Explicitly,

$$S_z = \frac{\hbar}{2}\big(\,|\uparrow\rangle\langle\uparrow| - |\downarrow\rangle\langle\downarrow|\,\big). \tag{2.37}$$

For simplicity, we will assume that the magnetic field is directed along the positive z-axis. We will denote the strength of the magnetic field by B. In this case, the Hamiltonian is given by

$$H = -\gamma B S_z = -\frac{\hbar}{2}\gamma B\big(\,|\uparrow\rangle\langle\uparrow| - |\downarrow\rangle\langle\downarrow|\,\big), \tag{2.38}$$

where γ is the gyromagnetic ratio of the electron.

Now let us consider the two-particle case. We will take two copies of the Hilbert space introduced above, labelling them as $\mathscr{H}^{(1)}$ and $\mathscr{H}^{(2)}$. We now have a basis

$$\{|\uparrow,\uparrow\rangle, |\uparrow,\downarrow\rangle, |\downarrow,\uparrow\rangle, |\downarrow,\downarrow\rangle\}. \tag{2.39}$$

We also have two spin operators:

$$S_z^{(1)} = \frac{\hbar}{2}\big(\,|\uparrow\rangle\langle\uparrow|^{(1)} - |\downarrow\rangle\langle\downarrow|^{(1)}\,\big), \tag{2.40}$$

$$S_z^{(2)} = \frac{\hbar}{2}\big(\,|\uparrow\rangle\langle\uparrow|^{(2)} - |\downarrow\rangle\langle\downarrow|^{(2)}\,\big). \tag{2.41}$$

Note that in a ket-bra expression, we have labelled the combination instead of the individual kets and bras. That is, we write $|\uparrow\rangle\langle\uparrow|^{(1)}$ instead of $|\uparrow\rangle^{(1)}\langle\uparrow|^{(1)}$. We also make use of the "lifting" identification from definition 5.

The complete Hamiltonian is now given by

$$H = -\gamma B(S_z^{(1)} + S_z^{(2)}). \tag{2.42}$$

It is easily checked that the eigensystem is given by

$$\big\{(|\uparrow,\uparrow\rangle, -\hbar\gamma B), (|\uparrow,\downarrow\rangle, 0), (|\downarrow,\uparrow\rangle, 0), (|\downarrow,\downarrow\rangle, \hbar\gamma B)\big\}. \tag{2.43}$$

Sine $B > 0$ and for electrons $\gamma < 0$, the ground state energy is $\hbar\gamma B$ with ground state $|\downarrow,\downarrow\rangle$.

Chapter 3

The spectral gap

Suppose we have an $L \times L$ lattice of particles. Let $\Lambda(L) = \{1, \ldots, L\}^2$ denote the set of lattice sites, and let $\mathcal{E} \subset \Lambda(L)^2$ denote the set of (oriented) edges of the lattice. We arbitrarily choose the orientation of $\mathcal{E}$ such that $(i, j) \in \mathcal{E}$ implies that j lies north or east of i. We associate to each particle $i \in \Lambda(L)$ a Hilbert space $\mathscr{H}^{(i)} \simeq \mathbb{C}^d$ of dimension d (the local dimension). To each particle we further associate a local Hamiltonian $h_1^{(i)}$, and to each pair $(i, j) \in \mathcal{E}$ of neighbouring particles a Hamiltonian $h^{(i,j)}$. This gives us a Hamiltonian

$$H^{\Lambda(L)} = \sum_{(i,j)\in\mathcal{E}} h^{(i,j)} + \sum_{i\in\Lambda(L)} h_1^{(i)} \tag{3.1}$$

on the total Hilbert space $\mathscr{H}^{\Lambda(L)} = \bigotimes_{i\in\Lambda(L)} \mathscr{H}^{(i)}$.

We will restrict our attention to translationally invariant Hamiltonians. That is, under identification of Hilbert spaces we demand that $h_1^{(i)} = h_1^{(j)}$ for any $i, j \in \Lambda(L)$ and $h^{(i,j)} = h^{(i',j')}$ for any $(i, j), (i', j') \in \mathcal{E}$ for which $i' - i = j' - j$.

Now let $\lambda_0 \leq \lambda_1$ be the two smallest eigenvalues of $H^{\Lambda(L)}$, where $\lambda_0 = \lambda_1$ iff λ_0 is degenerate. We are interested in the behaviour of the quantity $\Delta = \lambda_1 - \lambda_0$ (the spectral gap) in the thermodynamic limit $L \to \infty$. In particular, we want to determine if the spectrum is *gapped*, or *gapless*, in the following sense:

Definition 10 (Gapped Hamiltonian)**.** *A family $\{H^{\Lambda(L)}\}$ of Hamiltonians is* gapped *if there is a constant $\gamma > 0$ and system size L_0 such that for all $L > L_0$, the ground state energy $\lambda_0(H^{\Lambda(L)})$ is non-degenerate and $\Delta(H^{\Lambda(L)}) \geq \gamma$.*

Definition 11 (Gapless Hamiltonian)**.** *A family $\{H^{\Lambda(L)}\}$ of Hamiltonians is* gapless *if there is a constant $c > 0$ such that for all $\varepsilon > 0$ there is an $L_0 \in \mathbb{N}$ such that for all $L > L_0$ and for any point x in $[\lambda_0(H^{\Lambda(L)}), \lambda_0(H^{\Lambda(L)}) + c]$ there is a $\lambda \in \operatorname{spec} H^{\Lambda(L)}$ such that $|x - \lambda| < \varepsilon$.*

Note that a gapped system is necessarily not gapless and vice versa, but that it is possible for a system to be neither gapped nor gapless. In what follows we will show that there exists an infinite set of families of Hamiltonians, each conforming to the constraints laid out above, and with each family being either gapped or gapless, but for which there exists no algorithm[6] which can determine in which of these two categories the family falls. That is, the problem of deciding whether a family of Hamiltonians is gapped or gapless is undecidable.

[6]Intuitively, an algorithm is a precise and finite recipe for performing some task. See section 6.1 for a more formal definition.

Chapter 4

Unconstrained local dimension d

In order to show that a problem is undecidable, we must provide infinitely many instances of that problem. After all, if only finitely many instances of the problem exist, a lookup table trivially provides an "algorithm" which decides each instance of the problem. With the constraints laid out in the previous chapter, an instance of the problem is determined by the local dimension d, a $d \times d$ matrix encoding the local Hamiltonian, and a $d^2 \times d^2$ matrix encoding the interaction Hamiltonian. In this chapter, we will exploit the freedom we have in choosing d to generate infinitely many instances of the problem, and show that deciding whether a system is gapped or gapless is undecidable. Physically speaking, this is not a very interesting result; in a physical context the local dimension will generally be fixed, and only the $\sim d^4$ parameters of the Hamiltonian can be varied. For example, in the system introduced in section 2.5 the local dimension $d = 2$, which is a fundamental property of electrons. In contrast, the magnetic field $\mathbf{B}$ can easily be altered in the lab, and this will give rise to different parameters in the Hamiltonian.

The approach we will take in this chapter is to reduce the spectral gap problem to the Wang tiling problem. The Wang tiling problem (also known as the domino problem) was introduced by Hao Wang in 1961 ([Wan61]). A *Wang tile* is a unit square with edges coloured with one of finitely many colours. A finite set $\mathcal{T}$ of Wang tiles is said to tile the plane $\mathbb{Z}^2$ if there exists a mapping $\mathbb{Z}^2 \to \mathcal{T}$ such that abutting edges of adjacent tiles have the same colour (rotations and reflections are not allowed). In 1966 Robert Berger ([Ber66]) showed that there exists no algorithm which, given any set of tiles as input, decides whether or not this set can tile the plane. Or in other words, the tiling problem is undecidable.

The treatment in this chapter is based on section 5.1 from [CPW15b].

4.1 Hamiltonians with undecidable ground state energy

The first ingredient of our construction will be a Hamiltonian for which determining the ground state energy is equivalent to solving the tiling problem. More formally, for each set of tiles $\mathcal{T}$ we will construct a family of Hamiltonians $\{H^{\Lambda(L)}\}$ such that there exists an $L_0 \in \mathbb{N}$ such that for each $L > L_0$, $\lambda_0(H^{\Lambda(L)}) = 0$ if T tiles the plane, and $\lambda_0(H^{\Lambda(L)}) \geq 1$ otherwise.

Let $\mathcal{T} = \{1, \ldots, T\}$ denote a set of Wang tiles as described above. For $(i, j) \in \mathcal{E}$, let $C^{(i,j)} \subset \mathcal{T}^2$ denote the set of pairs of tiles (m, n) which are incompatible when placed on adjacent sites i and j. For any $i \in \Lambda(L)$, we choose a local Hilbert space $\mathscr{H}_c^{(i)} \simeq \mathbb{C}^T$ (i.e. $d = T$) with orthonormal basis $\{|1\rangle^{(i)}, \ldots, |T\rangle^{(i)}\}$. We then choose the following interaction Hamiltonian

$$h_c^{(i,j)} = \sum_{(m,n) \in C^{(i,j)}} |m\rangle\langle m|^{(i)} \otimes |n\rangle\langle n|^{(j)} \tag{4.1}$$

We choose $h_1^{(i)} = 0$, so that the total Hamiltonian is simply

$$H_c^{\Lambda(L)} = \sum_{(i,j)\in\mathcal{E}} h_c^{(i,j)}. \tag{4.2}$$

Note that if we consider the matrix of $H_c^{\Lambda(L)}$ with respect to the natural basis (arising from the tensor product of the $\{|m\rangle^{(i)}\}$ bases), it is a diagonal matrix with entries in $\mathbb{N}_0$. It follows immediately that $\operatorname{spec} H_c^{\Lambda(L)} \subset \mathbb{N}_0$.

If $\mathcal{T}$ tiles the plane, then for every $L \in \mathbb{N}$, there exists an assignment $\Lambda(L) \to \mathcal{T}$ which does not violate any of the constraints $C^{(i,j)}$. This assignment trivially leads to a quantum state with an energy of zero. Since $\operatorname{spec} H_c^{\Lambda(L)} \geq 0$, this state is the ground energy state, and $\lambda_0(H_c^{\Lambda(L)}) = 0$. If, however, $\mathcal{T}$ does not tile the plane, then there is some $L_0 \in \mathbb{N}$ for which no such assignment exists. Since any deviation from a valid tiling will have some positive energy contribution, this means that $\lambda_0(H_c^{\Lambda(L_0)}) > 0$. And since $\operatorname{spec} H_c^{\Lambda(L_0)} \subset \mathbb{N}_0$, this implies that $\lambda_0(H_c^{\Lambda(L_0)}) \geq 1$. The same now holds for all $L > L_0$, since a system of size $L > L_0$ trivially includes a system of size L_0.

4.2 A gapless Hamiltonian

If we hope to show that the spectral gap problem is undecidable, we will certainly need a gapless Hamiltonian. In particular, we will make use of the work by Fernández-González et al. ([FG+15]), which shows that there exists a Hamiltonian

$$H_q^{\Lambda(L)} = \sum_{(i,j)\in\mathcal{E}} h_q^{(i,j)}, \tag{4.3}$$

such that for $L \to \infty$, $\operatorname{spec} H_q^{\Lambda(L)}$ approaches a dense subset of $\mathbb{R}^+$. We can even restrict the entries of $H_q^{\Lambda(L)}$ to the rational numbers.

4.3 Combining the ingredients

For our first attempt at showing that the spectral gap is undecidable, we will simply add this Hamiltonian to our tiling Hamiltonian from the previous section. That is, we define $\mathscr{H}^{(i)} = \mathscr{H}_c^{(i)} \otimes \mathscr{H}_q^{(i)}$, and

$$h^{(i,j)} = h_c^{(i,j)} \otimes \mathbb{1}_q^{(i,j)} + \mathbb{1}_c^{(i,j)} \otimes h_q^{(i,j)}, \tag{4.4}$$

$$H^{\Lambda(L)} = \sum_{(i,j)\in\mathcal{E}} h^{(i,j)}. \tag{4.5}$$

It is easily checked that this gives us

$$\operatorname{spec} H^{\Lambda(L)} = \operatorname{spec} H_c^{\Lambda(L)} + \operatorname{spec} H_q^{\Lambda(L)} = \{\, \lambda + \mu \mid \lambda \in \operatorname{spec} H_c^{\Lambda(L)},\, \mu \in \operatorname{spec} H_q^{\Lambda(L)} \,\}. \tag{4.6}$$

Indeed, if $|\lambda\rangle_c$ and $|\mu\rangle_q$ are eigenkets of $H_c^{\Lambda(L)}$ and $H_q^{\Lambda(L)}$ with eigenvalues λ and μ, then $|\lambda\rangle_c \otimes |\mu\rangle_q$ is an eigenket of $H^{\Lambda(L)}$ with eigenvalue $\lambda + \mu$. Since $H_c^{\Lambda(L)}$ and $H_q^{\Lambda(L)}$ are both Hermitian, their eigenvectors span the Hilbert space on which they operate. Thus the set of eigenvectors we found for for $H^{\Lambda(L)}$ spans the product space. So this must be the complete set of eigenvectors.

So if $\mathcal{T}$ tiles the plane, then in the thermodynamic limit the spectrum of $H^{\Lambda(L)}$ is contained and dense in $[0, \infty)$, and otherwise it is contained and dense in $[\lambda_0, \infty)$, with $\lambda_0 \geq 1$. Whether or not $\mathcal{T}$ tiles the plane essentially acts as a 'switch' which kicks up the spectrum of our Hamiltonian. However, in both cases the Hamiltonian is gapless. In order to introduce a gap, we want to add a ground state with energy zero which persists regardless of whether $\mathcal{T}$ tiles the plane.

To achieve this, we add a new state to our Hilbert space: $\mathscr{H}^{(i)} = \mathscr{H}_0^{(i)} \oplus \mathscr{H}_c^{(i)} \otimes \mathscr{H}_q^{(i)}$, with $\mathscr{H}_0^i = \text{span}\{|0\rangle\}$. The state $|0\rangle$ will be our ground state with energy zero. If we leave the Hamiltonian unaltered, this is exactly what happens; the state $|0, \ldots, 0\rangle$ will be a ground state with energy zero regardless of whether $\mathcal{T}$ tiles the plane. However, there is a problem. Consider a state where every particle is in the $|0\rangle$ state, except for two neighbouring particles which are in states $|m\rangle_c^{(i)} \otimes |\alpha\rangle_q^{(i)} = |m, \alpha\rangle^{(i)}$ and $|n, \beta\rangle^{(j)}$. Suppose that $(m, n) \notin C^{(i,j)}$. That is, the tiles m and n at sites i and j are compatible. Suppose further that the $|\alpha\rangle$ and $|\beta\rangle$ state have an interaction energy $\epsilon \in (0, 1)$. Since all other particles are in the $|0\rangle$ state, this is the only interaction, and the energy of the entire state is ϵ. This means that there could potentially be eigenstates which fill the gap we are trying to create.

This motivates the following modification to our Hamiltonian: we will add an energy penalty of 1 to interactions between the $|0\rangle$ state and any other state. This leads to the following Hamiltonian:

$$h^{(i,j)} = |0\rangle\langle 0|^{(i)} \otimes \mathbb{1}_{cq}^{(j)} + \mathbb{1}_{cq}^{(i)} \otimes |0\rangle\langle 0|^{(j)} \tag{4.7}$$

$$+ h_c^{(i,j)} \otimes \mathbb{1}_q^{(i,j)} \tag{4.8}$$

$$+ \mathbb{1}_c^{(i,j)} \otimes h_q^{(i,j)}, \tag{4.9}$$

$$H^{\Lambda(L)} = H_0 + H_c + H_q = \sum_{(i,j)\in\mathcal{E}} h^{(i,j)}, \tag{4.10}$$

where H_0, H_c and H_q are defined by summing over (4.7), (4.8) and (4.9) separately.

Before we present a formal statement and proof of our result, it is worth noting that the above Hamiltonian is *frustration-free*. That is, the total Hamiltonian $H^{\Lambda(L)}$ can be minimized by minimizing each of the local Hamiltonians $h^{(i,j)}$; the state $|0, \ldots, 0\rangle$ simultaneously minimizes the total and local Hamiltonians. By restricting our attention to frustration-free Hamiltonians, we can prove a stronger result; the smaller the decision problem for which we show undecidability the stronger this statement is.

We now present a formal statement and proof of our result.

Theorem 1. *Let $\mathcal{T}$ denote a set of Wang tiles, and consider the corresponding family of frustration-free Hamiltonians $\{H^{\Lambda(L)}\}$ as described above. Then the following statements hold:*

(i) If $\mathcal{T}$ tiles the plane, then $\operatorname{spec} H^{\Lambda(L)} \to \mathbb{R}^+$ *as $L \to \infty$, i.e. the Hamiltonian is gapless and the spectrum becomes dense in $\mathbb{R}^+$.*

(ii) If $\mathcal{T}$ does not tile the plane, then there is an $L_0 \in \mathbb{N}$ such that for all $L > L_0$, $H^{\Lambda(L)}$ has a unique ground state and a spectral gap of size at least one, i.e.

$$\operatorname{spec} H^{\Lambda(L)} \backslash \{0\} \geq 1. \tag{4.11}$$

Proof. We will associate with each element of our product basis a signature $\sigma \in \{0, \ldots, T\}^{L^2}$. With $|0\rangle^{(i)}$ we will associate $\sigma_i = 0$, and with $|k, \alpha\rangle^{(i)}$ we will associate $\sigma_i = k$. This gives us a natural decomposition of our Hilbert space:

$$\mathscr{H}^{\Lambda(L)} = \bigotimes_{i\in\Lambda(L)} \mathscr{H}^{(i)} \simeq \bigoplus_{\sigma} \mathscr{H}_\sigma, \tag{4.12}$$

where $\mathscr{H}_\sigma$ is the space spanned by all basis states with signature σ.

Under this decomposition the Hamiltonian is block-diagonal. That is,

$$H^{\Lambda(L)} = \bigoplus_{\sigma} H_\sigma. \tag{4.13}$$

Consequently, $\operatorname{spec} H^{\Lambda(L)} = \bigcup_\sigma \operatorname{spec} H_\sigma$.

We distinguish three cases, and identify the spectra coming from each of these cases:

(i) $\sigma = (0, \ldots, 0)$: it is clear that this yields an eigenvalue of 0.

(ii) $\sigma \neq (0, \ldots, 0)$ but $\sigma_i = 0$ for some i: for any unit basis element $|\psi_\sigma\rangle$ with such a signature, we have $\langle\psi_\sigma|H_0|\psi_\sigma\rangle \geq 1$. Since $H_c, H_q \geq 0$, this implies $\operatorname{spec} H_\sigma \geq 1$. Denote the union of all such spectra by S.

(iii) $\sigma \in \{1, \ldots, T\}^{L^2}$: in the subspace spanned by states with such a signature, $H_0 = 0$, so $H^{\Lambda(L)} = H_c^{\Lambda(L)} \otimes \mathbb{1}_q^{\Lambda(L)} + \mathbb{1}_c^{\Lambda(L)} \otimes H_q^{\Lambda(L)}$, and $\operatorname{spec} H^{\Lambda(L)} = \operatorname{spec} H_c^{\Lambda(L)} + \operatorname{spec} H_q^{\Lambda(L)}$.

Combining these spectra we get

$$\operatorname{spec} H^{\Lambda(L)} = \{0\} \cup \left(\operatorname{spec} H_c^{\Lambda(L)} + \operatorname{spec} H_q^{\Lambda(L)}\right) \cup S. \tag{4.14}$$

So, if $\mathcal{T}$ tiles the plane then $0 \in \operatorname{spec} H_c^{\Lambda(L)}$, so $\operatorname{spec} H^{\Lambda(L)} \to \mathbb{R}^+$ as $L \to \infty$.

If $\mathcal{T}$ does not tile the plane then there is an $L_0 \in \mathbb{N}$ such that for all $L > L_0$, $\operatorname{spec} H_c^{\Lambda(L)} \geq 1$, and thus $\operatorname{spec} H^{\Lambda(L)} \backslash \{0\} \geq 1$. On the other hand, $\operatorname{spec} H_q^{\Lambda(L)} \to \mathbb{R}^+$. Furthermore, the ground state energy of 0 is reached by the unique ground state $|0, \ldots, 0\rangle$. So we have a spectral gap of at least 1. □

Note that in the above construction $\|h^{(i,j)}\| \leq 1 + \|h_q^{(i,j)}\|$. By rescaling $h_q^{(i,j)}$, we can demand $\|h^{(i,j)}\| \leq 1 + \epsilon$ for any $\epsilon > 0$. Further note that all entries in our Hamiltonian are rational. This allows us to state the following corollary:

Corollary 2 (Undecidability of the spectral gap for unconstrained local dimension)**.** *Let $\epsilon > 0$. Consider all families $\{H^{\Lambda(L)}\}$ of frustration-free translationally invariant nearest neighbour Hamiltonians on 2D square lattices that are described by a hermitian matrix h with rational entries and operator norm smaller than $1 + \epsilon$. There is no algorithm that upon input of h decides whether it describes a gapped or gapless system, even under the promise that one of them is true and that in the gapped case the gap is at least of size 1.*

Chapter 5

Constrained local dimension d

We will now turn our attention to the more interesting case of constrained local dimension d. The only remaining freedom is in the $\sim d^4$ parameters of the local and interaction Hamiltonian. We will prove our result using a direct reduction to the halting problem. We will fix some universal Turing machine M. For each $n \in \mathbb{N}$, we will construct a Hamiltonian whose ground state energy depends on whether M halts on input n. We will then use a construction similar to the one in section 4.3 to "lift" the undecidability of this ground state energy into the undecidability of the spectral gap. In this construction we will encounter a number of challenges:

(i) It is not clear how we should tie the computation performed by a universal Turing machine (or any Turing machine) to the ground state energy of a Hamiltonian.

(ii) We will have to encode the countably infinite set of possible inputs n into the finite set of parameters of our Hamiltonian. In fact, we have to encode an essentially global property (the input n) into our Hamiltonian, which can only encode local (nearest-neighbour) interactions.

(iii) Though there are known approaches (such as [GI09]) to problem (i), in all such approaches the energy gap between the halting and non-halting case disappears as the system size tends to infinity. In these approaches, the size of the system is directly related to the run-time of the computation, so restricting the size of the system to some fixed part of the grid (as the grid grows to an infinite plane) would also restrict us to finite calculations, which is unacceptable.

We will tackle problem (i) using an approach by Gottesman and Irani ([GI09]), which allows us to encode the computation of an arbitrary Quantum Turing machine in a translationally invariant Hamiltonian on a one-dimensional chain of Hilbert spaces. As explained above, having only one such chain will not provide us with the necessary energy gap as the system size tends to infinity. We will therefore make sure we have infinitely many chains, with finite but arbitrary length. We will use a modified version of the quasi-periodic Robinson tiling (introduced in [Rob71]) to provide us with these chains, on which the Turing machines will "run". As we increase the size of our grid, the Robinson tiling will provide us with more and more chains, and the (maximum) size of these chains will increase without bound. So, for any input n on which M halts, we will eventually get a chain which is long enough to encode the computation of M on n. We will make sure that this gives us some positive energy contribution. As our system grows further, we will get additional and longer chains. These longer chains will also be able to encode the computation of M on n, and therefore they will also give a positive energy contribution. As indicated above, this energy contribution will tend to zero as the size of the chain tends to infinity. But crucially, we never lose (or modify) the energy contribution of the smallest chain that was able to encode the computation. Therefore, the energy gap between the halting and non-halting case does not disappear in the thermodynamic limit, solving problem (iii).

We will solve problem (ii) by using the same construction by Gottesman and Irani to encode a quantum Turing machine (QTM) which performs the phase estimation algorithm. This algorithm will allow us to

write the phase of some given unitary (which we can encode in our Hamiltonian) to the tape in binary form (which the universal Turing machine can use as its input). The quantum phase estimation algorithm is a well-known quantum algorithm. The challenge will be to execute this algorithm with a QTM (as opposed to doing it using a quantum circuit, for which it is usually defined).

In the next chapter we will treat the concepts of quantum computation which we will require in our construction. This will also allow us to define more rigorously some of the concepts used above. In chapter 7 we will show how the quantum phase estimation can be performed. In chapter 8 we will show how the construction by Gottesman and Irani allows us to encode arbitrary quantum computations in local one-dimensional Hamiltonians. Then, in chapter 9, we will briefly treat the Robinson Tiling, and state some results which will be necessary for our construction. And finally, in chapter 10, we will combine the ingredients to obtain our main result.

Chapter 6

Quantum computation

The purpose of this chapter will be to give a brief overview of both classical and quantum models of computation. This will allow us to define precisely the halting problem and the concept of a universal Turing machine, both of which will be central ingredients in proving our undecidability result. We will also develop the tools required to perform the quantum phase estimation which will be treated in the next chapter. The language developed in this chapter will also be useful when we discuss how quantum computation can be encoded in a local Hamiltonian in chapter 8.

6.1 Classical theory of computation

Our goal is to show that there exists no algorithm which can solve the spectral gap problem as defined in chapter 3. To do this, we must first define what we mean by an *algorithm*. The standard approach to making this definition is by use of the concept of *Turing machines*. In this approach an algorithm is defined as any sequence of operations which can be simulated by a Turing machine (defined below). The motivation for this definition is a central hypothesis known as the *Church-Turing-Thesis*:

> The class of functions computable by a Turing machine corresponds exactly to the class of functions which we would naturally regard as being computable by an algorithm. [NC10, p. 125]

Definition 12 (Turing machine)**.** *A (deterministic) Turing machine (TM) is defined by a triplet* (Σ, Q, δ) *where* Σ *is a finite alphabet with an identified blank symbol* $\#$*,* Q *is a finite set of states with an identified initial state* q_0 *and final state* $q_f \neq q_0$*, and* δ *– the transition function – is a function*

$$\delta : Q \times \Sigma \to \Sigma \times Q \times \{L, R\}. \tag{6.1}$$

The TM has a two-way infinite tape of cells indexed by $\mathbb{Z}$*, which can each hold a single alphabet symbol* $\sigma \in \Sigma$*, and a single read/write tape head that moves along the tape. A* configuration *of the TM is a complete description of the contents of the tape, the location of the tape head and the current internal state* $q \in Q$*. At any time, only a finite number of tape cells may contain non-blank symbols.*

For any configuration c*, the successor configuration* c' *is defined by applying the transition function to the current state and the symbol under the tape head, replacing them by those specified in the transition function and moving the head left (L) or right (R) according to* δ*.*

By convention, the initial configuration satisfies the following conditions: the head is in cell 0*, called the* starting cell*, and the machine is in state* q_0*. We say that the initial configuration has* input $x \in (\Sigma \backslash \{\#\})^*$ *if* x *is written on the tape in positions* $0, 1, 2, \ldots$ *and all the other tape cells are blank.*[7] *The TM* halts *on input* x *if it eventually reaches the final state* q_f*. The number of steps* a *it takes to halt on input* x *is its running time on input* x*. If a TM halts, then its output is the string in* Σ^* *consisting of those tape contents*

[7] We write A^* for the set of all finite sequences (i.e. *strings*) of elements in A.

from the leftmost non-blank symbol to the rightmost non-blank symbol, or the empty string if the entire tape is blank. [CPW15b, p. 42]

It is clear that the set of all possible Turing machines (under identification of relabled sets Σ and Q) is countable. Therefore, we could in theory associate to each Turing machine a natural number, and vice versa. Let us arbitrarily fix such an association. For any Turing machine M, we will denote the associated natural number as T_M and call it the *Turing number* for M. It also clear that a similar correspondence exists between the possible inputs to some particular TM and the natural numbers. This allows us to make the following definition:

Definition 13 (Universal Turing machine)**.** *A universal Turing machine (UTM) is a Turing machine which, given as its input the encoding of the Turing number T_M for some Turing machine M followed by some designated separator symbol $|$ and an arbitrary input string x, calculates the output of M on x.*

The construction of a UTM is beyond the scope of this text, since for our purposes only its existence is relevant. It is worth pointing out, however, that a modern computer is essentially a UTM, except that a computer only has a finite "tape".

6.1.1 The halting problem

The proof of our main result will essentially be a direct reduction to the *Halting problem*, which Turing has shown to be undecidable ([Tur37]):

Theorem 3. *There is no Turing machine which, given natural numbers t and x, decides whether the Turing machine with Turing number t halts on the input encoded by x.*

Again, a proof of this theorem is outside of the scope of this text. For a sketch of the proof refer to [NC10, p. 130]. Note that because of the existence of a UTM, and using our definition of an algorithm, theorem 3 can be restated as follows:

Theorem 3 (Restated)**.** *For any fixed UTM M, there is no algorithm which, given a natural number x, decides whether M halts on the input encoded by x.*

This is essentially what we will be using in our proof. We will fix some UTM M, and for each natural number n we will construct a family of Hamiltonians $\{H^{\Lambda(L)}(n)\}$, with local dimension depending only on M, such that $\{H^{\Lambda(L)}(n)\}$ is gapped if M halts on n, and gapless if M does not halt on n. With the above theorem, this immediately implies the undecidability of the spectral gap for constrained local dimension.

6.2 Extension to quantum computation

While the Church-Turing Thesis states that any (naturally regarded) computable function can be computed by a Turing machine, it makes no claim about the efficiency of such a simulation. There is a stronger form of the Church-Turing thesis which does make such claims. For this, we first define a *probabilistic* Turing machine:

Definition 14 (Probabilistic Turing machine)**.** *A probabilistic Turing machine (PTM) is defined exactly as in definition 12, except that δ is altered such that it associates with each state-symbol-pair two (not necessarily distinct) internal states instead of just one. These two states are interpreted as each occurring with probability $\frac{1}{2}$ in each transition. [NC10, p. 127].*

The class of functions computable by a probabilistic Turing machine is exactly the same as the class of functions computable by a regular Turing machine. It is certainly no smaller, since any TM can be trivially extended to a PTM. And since we can simulate any PTM on a regular TM by simply stepping through all possible outcomes of each transition, it is also no larger. However, such a simulation may be very inefficient.

Thus the class of functions which can be *efficiently* simulated by a PTM may well be larger. This notion is made exact by the strong form of the Church-Turing Thesis, which states that any computation can be efficiently performed with a PTM:

> Any model of computation can be simulated on a probabilistic Turing machine with at most a polynomial increase in the number of elementary operations required. [NC10, p. 140]

This hypothesis has been cast into doubt by the discovery of quantum algorithms which efficiently solve problems for which no efficient solution on a PTM is known. An example of such algorithm is Shor's algorithm for factoring integers. This motivates the introduction of yet another model of computation. All definitions and theorems in this section, unless noted otherwise, are based on the definitions and theorems in chapter 6 in [CPW15b].

Definition 15 (Quantum Turing Machine)**.** *A quantum Turing machine (QTM) is defined by a triplet (Σ, Q, δ) where Σ is a finite alphabet with an identified blank symbol $\#$, Q is a finite set of states with an identified initial state q_0 and final state $q_f \neq q_0$, and δ – the quantum transition function – is a function*

$$\delta : Q \times \Sigma \to \widetilde{\mathbb{C}}^{\Sigma \times Q \times \{L,R\}}. \tag{6.2}$$

Here, $\widetilde{\mathbb{C}}$ is the set of all complex numbers α such that there is a deterministic algorithm which computes the real and imaginary part of a to within 2^{-n} in time polynomial in n.

The QTM has a two-way infinite tape of cells indexed by $\mathbb{Z}$ and a single read/write tape head that moves along the tape. We define configurations exactly as for deterministic TMs.

Let $\mathcal{S}$ be the inner product space of finite complex linear combinations of configurations of our QTM M with the Euclidean norm. We call each element $\phi \in \mathcal{S}$ a superposition of M. The QTM M defines a linear operator $U_M : \mathcal{S} \to \mathcal{S}$, called the time evolution operator of M as follows: if M starts in configuration c with current state p and scanned symbol σ, then after one step M will be in the superposition of configurations $\psi = \sum_i \alpha_i c_i$, where each non-zero α_i is the amplitude of $|\tau\rangle|q\rangle|d\rangle$ in the transition $\delta(p, \sigma)$ (which we will denote as $\delta_t(p, \sigma, \tau, q, d)$) and c_i is the new configuration obtained by writing τ, changing the internal state to q and moving the head in the direction d. Extending this map to the entire $\mathcal{S}$ through linearity gives the linear time evolution operator U_M.

Let x be any valid input to the QTM (defined exactly the same as for deterministic TMs). Let $\phi_0 \in \mathcal{S}$ be the superposition that has support only on the initial configuration with input x. We say that the QTM halts with running time T if $U_M^T |\phi_0\rangle$ has support only on final configurations (that is, configurations with internal state q_f), and $U_M^t |\phi_0\rangle$ has no support on final configurations for any $t < T$. [BV97, p. 1425].

For us, the utility of this new model of computation will not be its potentially greater capacity for efficient computation, but its use in extracting information out of the parameters of our Hamiltonian to serve as the input of a UTM (see chapter 7).

It will be convenient to allow our (Q)TMs to keep the head still in addition to being able to move it left and right.

Definition 16 (Generalised TM and QTM)**.** *A generalised TM or QTM is defined exactly as a standard TM (definition 12) or standard QTM (definition 15) except that the set of head movement directions is $\{L, N, R\}$ instead of just $\{L, R\}$. The definition of successor states is amended such that when $d = N$ the head is kept still.*

It is easily seen that this extension does not influence the class of functions which can be computed or computed efficiently. From now on we will assume that all our TMs and QTMs are of this form.

We will also define various classes of deterministic and quantum Turing machines:

Definition 17 (Reversibility)**.** *A deterministic TM M is reversible if each configuration has at most one predecessor. That is, if $\mathcal{C}$ is the set of configurations of M (reachable or not), and $f_M : \mathcal{C} \to \mathcal{C}$ is the function that describes the evolution of M on arbitrary configurations, then*

$$\forall c \in \mathcal{C} \quad |\{c' \in \mathcal{C} \mid f_M(c') = c\}| \leq 1. \tag{6.3}$$

Definition 18 (Well-formed)**.** *We say that a QTM is well-formed if its time evolution operator is an isometry, that is, it preserves the Euclidean norm.*

Note that any reversible TM is also a well-formed QTM where the quantum transition function $\delta(p,\sigma) = |q\rangle|\tau\rangle|d\rangle$ if $\delta(p,\sigma) = (q,\tau,d)$ for the reversible TM and 0 otherwise.

Definition 19 (Normal form)**.** *A well-formed QTM or reversible TM $M = (\Sigma, Q, \delta)$ is in normal form if*

$$\forall \sigma \in \Sigma \quad \delta(q_f, \sigma) = |\sigma\rangle|q_0\rangle|N\rangle\,. \tag{6.4}$$

Definition 20 (Unidirectional)**.** *A QTM $M = (\Sigma, Q, \delta)$, or (partial) transition function δ, is unidirectional if each state can be entered from only one direction. In other words, if $\delta_t(p_1, \sigma_1, \tau_1, q, d_1)$ and $\delta_t(p_2, \sigma_2, \tau_2, q, d_2)$ are both non-zero, then $d_1 = d_2$. The definition is extended to deterministic TMs in the obvious way.*

Theorem 4. *A TM $M = (\Sigma, Q, \delta)$ is reversible iff*

(i) M is unidirectional, and

(ii) δ is injective.

Furthermore, any partial transition function satisfying the above conditions can always be completed to the transition function of a reversible TM.

Proof. See theorem B.1 and corollary B.3 in [BV97]. □

We have an analogous theorem for QTMs:

Theorem 5. *A unidirectional QTM $M = (\Sigma, Q, \delta)$ is well-formed iff its quantum transition function δ satisfies the following conditions:*

(i) Normalisation

$$\forall p, \sigma \in Q \times \Sigma \quad \|\delta(p,\sigma)\| = 1. \tag{6.5}$$

(ii) Orthogonality

$$\forall (p_1, \sigma_1) \neq (p_2, \sigma_2) \in Q \times \Sigma \quad \langle \delta_r(p_1, \sigma_1) | \delta_r(p_2, \sigma_2) \rangle = 0, \tag{6.6}$$

where $\delta_r : Q \times \Sigma \to \mathbb{C}^{\mathbb{Q} \times \Sigma}$ is the reduced transition function obtained from δ by ignoring the direction (i.e. projecting $\delta(q,\sigma)$ on $\mathbb{C}^{\mathbb{Q}\times\Sigma}$).

Furthermore, a partial quantum transition function satisfying these conditions can always be completed to a well-formed transition function.

Proof. See theorem 19 in [CPW15b]. □

Theorem 6. *Let M be a well-formed, normal form unidirectional QTM. Then the time evolution operator U_M of M is a unitary.*

Proof. See the aside in the proof of theorem 19 in [CPW15b]. □

Definition 21 (Properness)**.** *A QTM behaves properly (or is proper) on a subset $\mathcal{X}$ of initial superpositions if whenever initialised in $\phi \in \mathcal{X}$, the QTM halts in a final superposition where each configuration has the tape head in the starting cell, the head never moved to the left of the starting cell, and the QTM never enters a configuration in which the head is in a superposition of different locations.*

Similarly, we say that a deterministic TM behaves properly (or is proper) on $\mathcal{X} \subset (\Sigma \backslash \{\#\})^*$ if the head never moves to the left of the starting cell, and it halts on every input $x \in \mathcal{X}$ with its tape head back in the starting cell.

All TMs and QTMs that will be used in our construction will be proper on a set $\mathcal{X}$, which will usually be clear from context. Importantly, every function computable by a TM is computable by a proper TM. In particular, there exist proper universal Turing machines.

6.3 QTM toolbox

We will show how new (Q)TMs can be constructed out of existing (Q)TMs by dovetailing two machines, implementing one machine as the subroutine of another machine, or reversing a machine. We will also give examples of some basic Turing machines. Again, the results from this chapter are taken from chapter 6 in [CPW15b], unless stated otherwise.

First, we will define the reversal of a QTM.

Definition 22 (Reversal)**.** *Let M_1 and M_2 be two QTMs with the same alphabet. We say that M_2 reverses the computation of M_1 if identifying the final state of M_1 with the initial state of M_2 and the initial state of M_1 with the final state of M_2 gives the following. For any input x on which M_1 halts, if c_x and ϕ_x are the initial configuration and final superposition of M_1 on input x, then M_2 halts on initial superposition ϕ_x with its final superposition having support only on c_x. [BV97, p. 1430] Replacing superposition with configuration, we get an analogous definition for the reversal of a deterministic TM.*

Lemma 7 (Reversal lemma)**.** *If M is a well-formed, normal form, unidirectional QTM (resp. normal form reversible TM) then there is a well-formed, normal form, unidirectional QTM (resp. normal form reversible TM) $M^\dagger$ that reverses the computation of M while taking two extra time steps and using the same amount of space. Moreover, if M is proper on $\mathcal{X}$, then $M^\dagger$ is proper on the set of final superpositions (resp. configurations) of M started on $\mathcal{X}$.*

We will give the construction of $M^\dagger$, but we will not prove that it has the stated properties. For a detailed proof of those claims, see lemma 23 in [CPW15b].

Suppose $M = (\Sigma, Q, \delta)$ is a well-formed, normal form, unidirectional QTM. Let $M^\dagger = (\Sigma, Q', \delta')$, where $Q' = Q \cup \{q_0', q_f'\}$ with q_0' and q_f' the new initial and final state respectively, and δ' is defined below. Here, for any state $q \in Q$, d_q is the unique direction in which that state can be entered, and $-d_q$ is the opposite direction.

(i) For each $\sigma \in \Sigma$, $\delta'(q_0', \sigma) = |\sigma\rangle|q_f\rangle\,|-d_{q_f}\rangle$.

(ii) For each $q \in Q \backslash \{q_0\}$ and each $\tau \in \Sigma$,

$$\delta'(q, \tau) = \sum_{p,\sigma} \delta_t(p, \sigma, \tau, q, d_q)^* \,|\sigma\rangle|p\rangle|-d_p\rangle\,. \tag{6.7}$$

(iii) For each $\sigma \in \Sigma$, $\delta'(q_0, \sigma) = |\sigma\rangle|q_f'\rangle|d_{q_0}\rangle$.

(iv) For each $\sigma \in \Sigma$, $\delta'(q_f', \sigma) = |\sigma\rangle|q_0'\rangle|N\rangle$.

Since reversible TMs are a special case of well-formed QTMs, this construction can easily be extended to reversible TMs.

Lemma 8 (Dovetailing lemma)**.** *Let M_1 and M_2 be well-formed normal form, unidirectional QTMs (resp. normal form reversible TMs) with the same alphabet, so that M_1 is proper on $\mathcal{X}_1$, M_2 is proper on $\mathcal{X}_2$, and $\mathcal{X}_2$ contains all final superpositions (resp. configurations) of M_1 started on $\mathcal{X}_1$. Then there is a well-formed unidirectional QTM (resp. normal form reversible TM) M which carries out the computation of M_1 followed by the computation of M_2 and that is also proper on $\mathcal{X}_1$.*

Proof. See lemma 4.9 in [BV97]. □

For the following lemma, we will consider multi-track TMs. A k track TM is simply a TM with alphabet $\Sigma = \Sigma_1 \times \cdots \times \Sigma_k$, where $\Sigma_1, \ldots, \Sigma_k$ are the alphabets of the individual tracks. Note that unlike in some other definitions in the literature, our multi-track TMs will not be allowed to move their heads independently on each track. Intuitively, there is a single head which "spans" all the tracks.

Lemma 9 (Subroutine lemma)**.** *Let M_1 be a two-track, normal form, reversible TM, and M_2 a two-track, normal form, reversible TM (or well-formed, normal form, unidirectional QTM) with the same alphabet and the following properties:*

(i) M_1 is proper on initial configurations in $\mathcal{X}_1$ and M_2 is proper on $\mathcal{X}_2$.

(ii) When started on $\mathcal{X}_1$, M_1 leaves the second track untouched and when started on $\mathcal{X}_2$, M_2 leaves the first track untouched.

(iii) There is a state q on M_1 that, when started on $\mathcal{X}_1$, can only be entered with the head in the starting cell.

(iv) $\mathcal{X}_2$ contains all the output superpositions (with q_f replaced by q_0) of k consecutive executions of M_2 started from an initial configuration in $\mathcal{X}_1$ for all $0 \leq k \leq r$, where $r \in \mathbb{N} \cup \{\infty\}$ is the maximum number of times that q is entered when M_1 runs on input $\mathcal{X}_1$.

Then there is a normal form, reversible TM (or well-formed, normal form unidirectional QTM) M which behaves properly on $\mathcal{X}_1$ and acts as M_1 except that each time it would enter state q, it instead runs machine M_2.

Proof. See lemma 4.8 in [BV97]. □

We will now explicitly construct a simple reversible TM. We will also state the existence of a number of other reversible TMs, but we will not provide a construction for them. Please refer to lemmas 25 to 30 in [CPW15b] for the constructions.

Lemma 10 (Copying machine)**.** *There is a two-track, normal form, reversible TM* Copy *with alphabet $\Sigma \times \Sigma$ that, on input s written on the first track, behaves properly, copies the input to the second track and runs for time $2|s| + 1$, using $|s| + 1$ space.*

Proof. We simply step the head right, copying the symbol from the first track to the second. However, we defer copying the starting cell until the end of the computation, so that we can locate the starting cell again (without stepping to the left of the starting cell, which would violate properness). It is straightforward to verify that the following normal form transition function implements Copy:

	$[\sigma, \#]$	$[\#, \#]$	$[\tau, \tau]$
q_0	$([\sigma, \#], q_1, R)$	$([\#, \#], q_f, N)$	
q_1	$([\sigma, \sigma], q_1, R)$	$([\#, \#], q_2, L)$	
q_2	$([\sigma, \sigma], q_f, N)$		$([\tau, \tau], q_2, L)$
q_f	$([\sigma, \#], q_0, N)$	$([\#, \#], q_0, N)$	$([\tau, \tau], q_0, N)$

$\forall \sigma, \tau \in \Sigma \backslash \{\#\}$ (6.8)

This partial transition function meets the two conditions of theorem 4, so it can be completed to give a reversible TM. □

Lemma 11 (Shift-right machine)**.** *There exists a normal form, reversible TM* Shift *with alphabet Σ that, on input s behaves properly, shifts s one cell to the right and runs for time $2|s| + 2$ using space $|s| + 2$.*

Lemma 12 (Equality machine)**.** *There is a three-track, normal form, reversible TM* Eql *with alphabet $\Sigma \times \Sigma \times \{\#, 0, 1\}$ that, on input $s; t; b$, where s and t are arbitrary input strings and b is a single bit, behaves properly and outputs $s; t; b'$, where $b' = \neg b$ if $s = t$ and $b' = b$ otherwise. Further more,* Eql *runs for time $2|s| + 1$ using space $|s| + 1$.*

The following machines will make use of numbers. Most of these machines will operate on binary numbers written in little-endian order. That is, numbers written in base 2, using designated symbols $0, 1 \in \Sigma$, with the least significant bit being the first bit on the tape. When describing the input or output of such machines, we will simply use "n" to refer to the tape configuration representing the number n in little-endian binary. Some machines will operate on unary numbers. That is, numbers written by simply repeating a designated symbol $1 \in \Sigma$ that many times. We will use "$1n$" to refer to the tape configuration representing the number n in unary.

Lemma 13 (Increment and decrement machines)**.** *There exist normal form, reversible TMs* Inc *and* Dec *with alphabet* $\{\#, 0, 1\}$ *that, on little-endian binary input* n *(with* $n > 1$ *for* Dec*), behave properly and output* $n + 1$ *or* $n - 1$ *respectively. Both machines run for time* $O(\log n)$ *and use at most* $|n| + 2$ *space.*

Lemma 14 (Looping Lemma)**.** *There is a two-track, normal form, reversible TM* Loop2 *with alphabet* $\{\#, 0, 1\}$*, which has the following properties. On input* $n; m$ *with* $n < m$ *both little-endian binary numbers,* Loop2 *behaves properly, runs for time* $O((m-n)\log m)$*, uses space* $|m|+2$ *and halts with its tape unchanged. Moreover,* Loop2 *has a special state* q *such that on input* $n; m$*, it visits state* q *exactly* $m - n$ *times, each time with its tape head back in the start cell.*

There is also a one-track, normal form, proper, reversible TM Loop *which, on input* $m \geq 1$*, behaves as* Loop2 *on input* $0; m$*.*

Lemma 15 (Binary addition and subtraction machines)**.** *There exist two-track, normal form, reversible TMs* Add *and* Sub *with alphabet* $\{\#, 0, 1\}$*, which have the following properties. On input* $n; m$ *with* n *and* m *both little-endian binary numbers and* $m \geq 1$*,* Add *behaves properly and outputs* $n + m; m$*. On input* $n; m$ *with* $n > m$*,* Sub *behaves properly and outputs* $n - m; m$*. Both TMs run for time* $O(m \log m \log n)$ *and use* $\max(|n|, |m|) + 2$ *space.*

Lemma 16 (Unary to binary converter)**.** *There exists a two-track, normal form, reversible TM* UtoB *with alphabet* $\{\#, 1\} \times \{\#, 0, 1\}$ *with the following properties. On input* $1n; \#$ *(*n *written in unary on the first track),* UtoB *behaves properly and outputs* $1n; n$ *(*n *written in little-endian binary on the second track). Furthermore,* UtoB *runs for* $O(n^2 \log n)$ *steps and uses* $n + 1$ *space.*

Chapter 7

Quantum phase estimation

In this chapter we will introduce the quantum phase estimation algorithm. This introduction will be based on the material in chapter 5 of [NC10]. We will also explain its role in our construction. Finally, we will sketch the construction of a QTM which can perform this algorithm exactly, using the lemmas developed in section 6.3.

The goal of the quantum phase estimation algorithm is the following. Given a unitary matrix $U_\varphi = \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{2\pi i\varphi} \end{smallmatrix}\right)$ and its second eigenvector $|1\rangle$, to determine the phase φ to a certain (fixed) precision. If φ can be written as a fraction with as its denominator a sufficiently small power of 2, the estimation can be performed exactly (that is, φ is recovered exactly). Otherwise, the algorithm will output an approximation $\widetilde{\varphi}$ which is close to φ with high probability. For our purposes, only the former will be relevant.

The quantum phase estimation algorithm is usually implemented using *quantum circuits*. Quantum circuits are an alternative model of quantum computation. A quantum circuit can be represented as a graph (a directed acyclic graph, in fact) where the edges are "wires" running from the input to the output through "quantum gates" (the nodes in the graph). Each wire represents a 2-dimensional Hilbert space (a qubit), and the total Hilbert space is the tensor product of these spaces. The quantum gates are unitary matrices that operate on the incoming wires, with the result of the application being represented by the outgoing wires.

7.1 The quantum phase estimation circuit

The quantum circuit for the quantum phase estimation algorithm is shown in figure 7.1. The Hilbert space for this circuit consists of n output qubits and one ancillary qubit. The output qubits are initialized in the $|0\rangle$ state, and the ancillary qubit is initialized in the $|1\rangle$ state, which is the eigenstate of U for which we want to find the phase.

There are three types of gates in this circuit: the Hadamard gate (represented by H), the controlled U^{2^k} gates, and the inverse quantum Fourier transform on n qubits (F_n^{-1}). The Hadamard gate simply applies the unitary matrix $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ to its operand. This rotates the output qubits from the state $|0\rangle$ into the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

For any two-level unitary operator V, the controlled V-gate applies V to its operand (the wire going "into" the gate) if its control qubit (the wire with the $\bullet$) is $|1\rangle$, and leaves its operand fixed if its control qubit is $|0\rangle$. This transformation is extended to superpositions through linearity. So, if we write $|\phi\rangle_c |\psi\rangle_u$ for the state with control qubit $|\phi\rangle$ and operand qubit $|\psi\rangle$, then the controlled U^{2^k}-gate maps $|+\rangle_c |1\rangle_u$ to

$$\frac{1}{\sqrt{2}}\left(|0\rangle_c |1\rangle_u + |0\rangle_c e^{2\pi i(2^k\varphi)} |1\rangle_u\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(2^k\varphi)} |1\rangle\right)_c |1\rangle_u . \tag{7.1}$$

Thus, after consecutively applying controlled $U^{2^{n-k}}$-gates to the ancillary bit with output qubit k as the

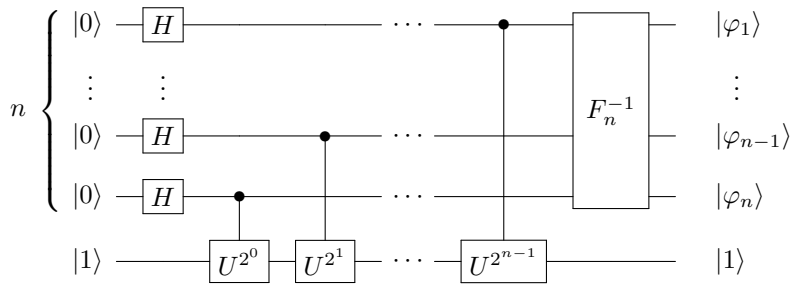


Figure 7.1: The quantum circuit for the quantum phase estimation algorithm.

control qubit, our quantum system is in the state

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i(2^{n-1}\varphi)}|1\rangle\right)_1\left(|0\rangle + e^{2\pi i(2^{n-2}\varphi)}|1\rangle\right)_2\cdots\left(|0\rangle + e^{2\pi i(2^{0}\varphi)}|1\rangle\right)_n|1\rangle_u\,. \tag{7.2}$$

If we now assume that φ can be expressed exactly in n bits as $0.\varphi_1\varphi_2\ldots\varphi_n$, then the above can be rewritten as[8]

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.\varphi_n}|1\rangle\right)_1\left(|0\rangle + e^{2\pi i 0.\varphi_{n-1}\varphi_n}|1\rangle\right)_2\cdots\left(|0\rangle + e^{2\pi i 0.\varphi_1\ldots\varphi_n}|1\rangle\right)_n|1\rangle_u\,. \tag{7.3}$$

The quantum Fourier transform is defined as the following transformation on an orthonormal basis $|0\rangle,\ldots,|N-1\rangle$.

$$|j\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi i jk/N} \tag{7.4}$$

For $N = 2^n$ it is convention to decompose the Hilbert space into a product of two-level systems with bases $|0\rangle_i, |1\rangle_i$ as follows:

$$|j\rangle = \left|\sum_{k=1}^{n} j_k 2^{n-k}\right\rangle = |j_1\rangle_1|j_2\rangle_2\cdots|j_n\rangle_n = |j_1, j_2, \ldots, j_n\rangle\,, \tag{7.5}$$

where $j_1,\ldots,j_n \in \{0,1\}$ is the unique binary expansion of j. It is straightforward to show that using this convention the quantum Fourier transform can be written as

$$|j\rangle = |j_1,\ldots,j_n\rangle \mapsto \frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)_1\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)_2\cdots\left(|0\rangle + e^{2\pi i 0.j_1\ldots j_n}|1\rangle\right)_n\,. \tag{7.6}$$

Comparing (7.6) to (7.3), we see that after performing the inverse Fourier transform on the n output qubits, we obtain the state

$$|\varphi_1\rangle_1|\varphi_2\rangle_2\cdots|\varphi_n\rangle_n|1\rangle_u\,, \tag{7.7}$$

so that the output qubits now contain the exact binary expansion of φ.

[8]Here, $0.x_1x_2\ldots x_n$ is defined as $\sum_{k=1}^{n} x_k 2^{-k}$, with $x_1,\ldots,x_n \in \{0,1\}$.

7.2 The role of the quantum phase estimation algorithm

The local dimension of the Hamiltonian that we will construct in chapter 8 depends on the particular QTM that is encoded (more precisely, it depends on the size $|\Sigma|$ of the alphabet and the size $|Q|$ of the set of internal states). Therefore, if we fix the local dimension d, we can not encode infinitely many Turing machines. Luckily, we only have to encode one UTM to encode all possible computations. As long as we can supply this UTM with a sufficient set of inputs, that is. That requires encoding arbitrary input (a binary string, say) in the parameters of our Hamiltonian, while keeping the local dimension fixed. This is where the quantum phase estimation algorithm comes into play. For each $n \in \mathbb{N}$, we will construct a QTM P_n which performs this algorithm, where the phase φ is defined as the rational number whose binary decimal expansion contains the digits of n in reverse order after the decimal. The output of P_n (which is the binary expansion of n) will then be used as the input to our UTM. We will construct the machines P_n in such a way that they share their set of internal states and alphabet. This will allow us to encode all of the P_n in Hamiltonians with the same local dimension.

7.3 Construction of a quantum phase estimation QTM

Nishimura and Masanao have shown ([NO02], [NO09]) that the quantum circuit model and the QTM model are computationally equivalent. In particular, they show that any quantum circuit can be simulated with a QTM. However, no claims are made about the set of internal states of such a QTM. Therefore, we will explicitly construct the QTMs described in section 7.2. In particular, we will prove the following theorem.

Theorem 17 (Phase-estimation QTM)**.** *There exists a family of well-formed, normal form, unidirectional QTMs P_n indexed by $n \in \mathbb{N}$ with the following properties:*

(i) Both the alphabet and the set of internal states are identical for all P_n; only the transition rules differ.

(ii) On input $N \geq |n|$ written in unary, P_n behaves properly, halts deterministically after $O(poly(N)2^N)$ steps, uses $N+3$ space, and outputs the binary expansion of n (padded to N digits with leading zeroes). (Here, $|n|$ denotes the length of the binary expansion of n.)

(iii) For each choice of states p, q, alphabet symbols σ, τ, and directions D, the transition amplitude $\delta_t(p, \sigma, \tau, q, D)$ is one of the elements of the set

$$\left\{0, 1, \pm\frac{1}{\sqrt{2}}, e^{\pi i\varphi}, e^{\pi i 2^{-|\varphi|}}\right\}, \tag{7.8}$$

where $\varphi \in \mathbb{Q}$ is defined as the rational number whose binary decimal expansion contains the digits of n in reverse order after the decimal, and $|\varphi| = |n|$ is the length of the binary expansion of φ.

The detailed proof of theorem 17 can be found as theorem 11 in [CPW15b]. Here we will only give a sketch of the construction.

The role of the input N is to indicate how many bits should be used in estimating n, which directly relates to the amount of space (the length of the tape) the QTM will use. The QTM will behave as desired when $N > |n|$. We will make use of several tracks:

(a) A "quantum track" with alphabet $\{\#, 0, 1\}$. This track will be used to store the qubits involved in the computation. This is also the track which contains the input $1N$ (recall that this is N written in unary) and will contain the output.

(b) An auxiliary classical "input track" with alphabet $\{\#, 0, 1\}$ which will contain N (the input) but written in binary.

(c) An auxiliary (non-classical) "mark track" with alphabet $\{t, c, m_0, m_1, \#\}$ which will be used to mark the qubits to which the controlled-U_φ operations have to be applied, and to store an auxiliary qubit $|m\rangle$.

(d) An auxiliary "cU looping track" with alphabet $\{\#, 0, 1\}$ which will be used to control the number of times the U_φ gate has to be applied.

(e) An auxiliary "outer looping track" with alphabet $\{\#, 0, 1\}$ which will be used to iterate over the control qubits.

Additional auxiliary tracks may be used during certain stages of the algorithm, but these are not relevant for the global state. The quantum phase estimation algorithm proceeds in five stages:

(1) A preparation stage, which will initialize both the output qubits and the ancillary qubit, and apply the Hadamard gates to the output qubits. This stage will also initialize the input track (b).

(2) The controlled-unitary stage during which control-U_φ operations are applied between the output qubits and the ancillary qubit. Here, the output qubits are iterated in the opposite order from what has been described in section 7.1. This will lead to the binary expansion of φ being outputted in reverse order, which is exactly the binary expansion of n.

(3) A stage in which we locate the least significant bit of the output.

(4) The quantum Fourier transform stage, in which the inverse quantum Fourier transform is applied to the significant output qubits (identified in the previous stage).

(5) A reset stage, which resets all the auxiliary systems used during the computation to their initial configuration.

We will construct QTMs for each of these stages separately, and use the dovetailing lemma (lemma 8) to chain them together.

7.3.1 Preparation stage

During the preparation stage, we first run the UtoB machine from lemma 16 to write the input in binary form to track (b). We then want to initialize the first cell, which will be the ancillary qubit, to $|1\rangle$, and the next N cells, which will be the output qubits, to $|+\rangle$. Noting that the first N cells are in the $|1\rangle$ state, such a QTM is very straightforward to construct. An explicit transition function can be found as equation 51 in [CPW15b].

7.3.2 Controlled-unitary stage

We will first need a QTM which performs the controlled-U_φ operation k times. This is stated in the following lemma.

Lemma 18 (Controlled-U QTM)**.** *For any single-qubit unitary U, there exists a well-formed, normal form, unidirectional QTM cU^k with the following properties. The QTM operates on the tracks (a), (c) and (d). The input consists of a number $k \geq 1$ written in binary on the cU looping track (d) in little-endian order, a configuration containing a single t and a single c within the first n tape cells on the mark track (c) (and all other cells blank), and an arbitrary n-qubit state on the quantum track.*

On such input, the QTM applies the controlled-U operation k times between the control and target qubits on the quantum track marked by c and t, and then halts, having run for time $O(kn + k \log k)$, used at most $\max(n, |k|) + 2$ space, behaving properly and leaving the configurations of the looping and mark tracks unchanged.

Again, for a detailed proof, refer to lemma 31 of [CPW15b]. Here, we will sketch the construction of the above QTM. The machine will be constructed out of two simpler QTMs M_1 and M_2 by dovetailing them together in the sequence $M_1, M_2, M_1^\dagger$.

- The machine M_1 will scan right until it encounters a c on the mark track, at which point it will copy the the qubit on the quantum track to an internal qubit (implemented as the superposition of two internal states). It will then copy this internal qubit to the starting cell on the mark track (implemented as the superposition of $|m_0\rangle$ and $|m_1\rangle$).

- The machine M_2 will loop k times over a machine M' (using the TM Loop from lemma 14 and the subroutine lemma 9). The machine M' will copy the qubit on the mark track to an internal qubit, then scan right until it finds t on the mark track, and apply a single controlled-U operation between the internal qubit (acting as the control qubit) and the target qubit on the quantum track. It then moves back to the starting cell and copies its internal qubit to the qubit on the mark track (this will carry the phase kick from the controlled-U operation to the qubit on the mark track).

- Finally, running $M_1^\dagger$ will carry the qubit on the mark track (with its phase kicks) to the control qubit on the quantum track, while returning the mark track to its original state.

Next, we will need a machine which iterates over the output qubits, applying the $cU_\varphi^{2^{k-1}}$ QTM between the kth output bit and the ancillary qubit (as noted above, this is in the opposite order of what has been described in section 7.1). This machine will shift the c on the mark track (c), double the number on the cU looping track (d), and then execute the cU^k machine from lemma 18 using the subroutine lemma 9. This sequence of steps is then looped using the track (e) as the looping track (again, using the Loop machine from lemma 14 and the subroutine lemma 9). The construction of such a machine is straightforward.

We still have to initialise the auxiliary tracks to the right state to run the above machine. The mark track will be initialized with a t in the starting cell and a c on the cell corresponding to the last qubit on the quantum track. The cU looping track will be initialized to have a 1 in its starting cell. And finally, we copy N from the input track (b) to the outer looping track using the Copy machine from lemma 10. Again, this construction is completely straightforward.

Finally, we run a simple machine which resets the auxiliary tracks to their initial (blank) configuration. Some care must be taken to do this reversibly.

After this stage, the quantum track is in the stage

$$\frac{1}{2^{N/2}} |1\rangle \left(|0\rangle + e^{2\pi i 0.\varphi_N} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.\varphi_{N-1}\varphi_N} |1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i 0.\varphi_1 \ldots \varphi_N} |1\rangle\right). \tag{7.9}$$

(Compare this to equation (7.3).) Note that the normalisation factor is split up as a factor $\frac{1}{\sqrt{2}}$ on each of the output qubits.

7.3.3 Locating the LSB

A straightforward implementation of the inverse quantum Fourier transform would require the use of controlled-$U_{2^{-N}}$ gates. We cannot appeal to lemma 18, because N, and therefore the required unitary, is part of the input. However, we know that φ has an $|n|$-digit binary expansion, and that $N \geq |n|$. This means that $\varphi_{|n|+1}, \ldots, \varphi_N = 0$. Referring to equation (7.9), we see that the first $N - |n|$ output qubits are in the state $|+\rangle$. Ignoring these qubits, we are left with $|n|$ output qubits which are exactly in the state which would be obtained after performing the controlled-unitary stage on $|n|$ qubits. If we simply perform an $|n|$-qubit inverse Fourier transformation on these qubits, we recover the binary expansion of n. Setting the remaining qubits to $|0\rangle$, we will have performed the exact same computation as an N-qubit inverse Fourier transform, but we have not used gates which are dependent on the input.

In order to perform the trick described above, we need to locate the least significant bit (LSB). Noting that this bit will necessarily be 1 (i.e. $\varphi_{|n|} = 1$), we can see that the corresponding output qubit will be in the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ after the controlled-unitary stage. All previous output qubits will be in the state $|+\rangle$, so we can simply step right over the quantum tracks, applying Hadamard gates (see section 7.1) until we obtain a $|1\rangle$. For the details, please refer to section 6.5 of [CPW15b].

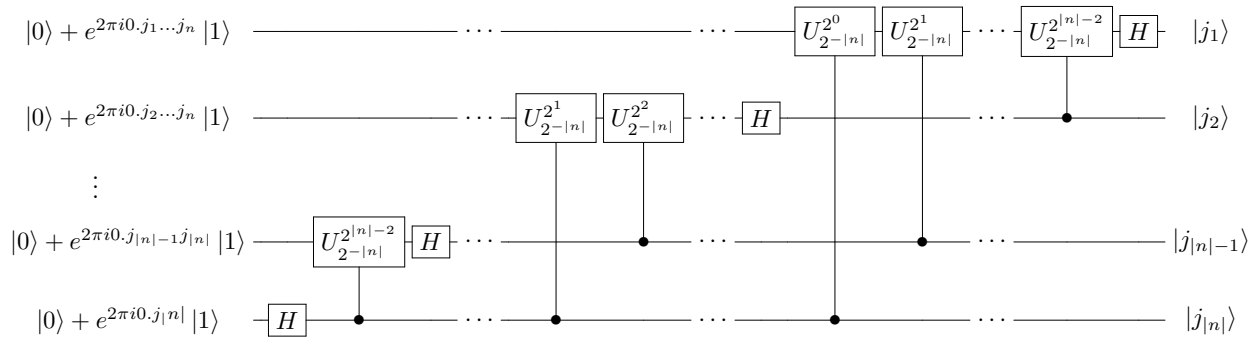


Figure 7.2: The quantum circuit for the inverse quantum Fourier transformation (based on figure 5.1 from [NC10]).

7.3.4 QFT stage

The quantum circuit for the inverse quantum Fourier transform is given in figure 7.2. A QTM that simulates this circuit can be constructed in a manner similar to the construction in section 7.3.2. For the details, please refer to section 6.6 of [CPW15b].

7.3.5 Reset stage

The computation is now essentially complete, with the output having been written to the quantum track. However, the auxiliary tracks are not yet in the blank state, as they should be to conform to the output specification of theorem 17. So, during this stage, the auxiliary tracks will be reset to their blank state in a reversible way. Again, please refer to section 6.7 of [CPW15b] for the details.

Chapter 8

Encoding quantum computation

In this chapter we will show how the evolution of a QTM can be encoded in a local Hamiltonian. We will give an overview of the construction in chapter 7 of [CPW15b], which is itself based on the construction in [GI09]. Please refer to those articles for detailed proofs of the statements in this chapter, as we will omit those here.

The main idea is to construct a Hamiltonian which has as its unique ground state a *computational history state*. This computational history state then encodes the evolution of the QTM. This encoding is realized by using a Hilbert space which is the product of two Hilbert spaces $\mathscr{H}_C$ and $\mathscr{H}_Q$. The Hilbert space $\mathscr{H}_C$ is called the *clock register* and only acts as a time label for the second Hilbert space. This second Hilbert space $\mathscr{H}_Q$ must be able to represent a complete configuration of the QTM. The complete computational history state is then a superposition over all the states reached by the QTM labelled by their time:

$$|\psi\rangle_{CQ} = \frac{1}{\sqrt{d_C}} \sum_{t=1}^{d_C} |t\rangle_C \, |\psi_t\rangle_Q \,, \tag{8.1}$$

where $d_C = \dim \mathscr{H}_C$, $\{|t\rangle\}$ is an orthonormal basis for $\mathscr{H}_C$, and $|\psi_t\rangle = \prod_{i=1}^{t} U_i \, |\psi_0\rangle$. The state $|\psi_0\rangle \in \mathscr{H}_Q$ represents the initial configuration of the QTM, and $U_t : \mathscr{H}_Q \to \mathscr{H}_Q$ is a unitary representing the t-th step of the quantum computation.

The specific QTM we will encode is the machine P_n from theorem 17 dovetailed with some UTM. The initial configuration $|\psi_0\rangle$ will be the configuration corresponding to an input N written in unary, for some integer N (also see (ii) in theorem 17). The integer N will be the length of the encoded tape up to a small constant offset which will become clear later.

The Hilbert space $\mathscr{H}_C \otimes \mathscr{H}_Q$ that the Hamiltonian acts on will itself be a finite chain of local Hilbert spaces, just like the complete Hilbert space $H^{\Lambda(L)}$ is a grid of local Hilbert spaces. That is,

$$\mathscr{H}_C \otimes \mathscr{H}_Q = \left(\bigotimes_{i=1}^{L} \mathscr{H}_C^{(i)} \right) \otimes \left(\bigotimes_{i=1}^{L} \mathscr{H}_Q^{(i)} \right), \tag{8.2}$$

where L is the length of the chain. These chains will be embedded in the grid using the modified Robinson tiling described in chapter 9. The length of the chain will correspond to the length of the tape the QTM runs on. Of course, for correct simulation of the QTM we theoretically need an infinite tape. However, any QTM which terminates will only use a finite amount of tape. So if we can encode arbitrary lengths of tapes, we will still be able to simulate any halting QTM. This does mean that we need to have infinitely many encodings (for different tape lengths) within the same grid. This, in turn, implies that the local Hamiltonian (i.e. $h_1^{(i)}$ and $h^{(i,j)}$) cannot be dependent on the tape length. Instead, only the Hilbert space will depend on the tape length.

Both the $\mathscr{H}_C$ chain and $\mathscr{H}_Q$ chain are made up of different "tracks". Mathematically, both $\mathscr{H}_C^{(i)}$ and

$\mathscr{H}_Q^{(i)}$ are themselves tensor products of different spaces, with each of those spaces representing a single "cell" of a single track of the tape. The chain $\mathscr{H}_C$ will consist of three different tracks:

(1) An "oscillator" track. On this track, an "arrow" will travel back and forth, driving the calculations on the other tracks.

(2) A track representing the head and internal state of a counter TM. This counter TM (which simply increments a number in some base) is reversible and will act as the actual clock. Every time the oscillator on track (1) completes an oscillation (i.e. the "arrow" has travelled back and forth along the track once), one step of the counter TM will be performed. Because it is reversible, even just performing a single step will bring the TM into a new unique configuration, which is sufficient to act as a unique time label (so we don't even need to complete the incrementation of the binary number each time).

(3) A track representing the tape of the counter TM.

The chain $\mathscr{H}_Q$ will also consist of three different tracks:

(4) A track representing the head and internal state of the QTM that is actually being encoded. Every time the oscillator on track (1) completes an oscillation, one step of the QTM will be performed.

(5) A track representing the tape of the QTM.

(6) A track which will be used to 'waste time' to run out the computation after the QTM has already reached a halting state.

In addition to the cells described above, both the $\mathscr{H}_C$ chain and the $\mathscr{H}_Q$ chain will also get a beginning marker $\left|\textcircled{<}\right\rangle$ and an end marker $\left|\textcircled{>}\right\rangle$. So the total Hilbert spaces are:

$$\mathscr{H}_C^{(i)} = \operatorname{span}\left\{\left|\textcircled{<}\right\rangle, \left|\textcircled{>}\right\rangle\right\} \oplus \left(\mathscr{H}_{(1)}^{(i)} \otimes \mathscr{H}_{(2)}^{(i)} \otimes \mathscr{H}_{(3)}^{(i)}\right), \tag{8.3}$$

$$\mathscr{H}_Q^{(i)} = \operatorname{span}\left\{\left|\textcircled{<}\right\rangle, \left|\textcircled{>}\right\rangle\right\} \oplus \left(\mathscr{H}_{(4)}^{(i)} \otimes \mathscr{H}_{(5)}^{(i)} \otimes \mathscr{H}_{(6)}^{(i)}\right), \tag{8.4}$$

where the labels (1) to (6) refer to the tracks above.

We will define a basis for each of the sites $\mathscr{H}_{(j)}^{(i)}$. The products of these states, together with the beginning and end markers will form a basis for the single site Hilbert space $\mathscr{H}^{(i)} = \mathscr{H}_C^{(i)} \otimes \mathscr{H}_Q^{(i)}$. The product of *these* states will form a basis for the entire chain. We will refer to the states of any such basis (depending on context) as *standard basis states*.

8.1 Legal and illegal states

The interaction Hamiltonian $h^{(i,j)}$ will have two types of terms: *penalty terms* and *transition rule terms*. Penalty terms have the form $|ab\rangle\langle ab|$ where a, b are standard basis states. This adds a positive energy contribution to any configuration containing a to the left of b. We call ab an *illegal pair*. We will also use single-site illegal states. These are easily implemented in terms of illegal pairs by adding penalty terms $|ax\rangle\langle ax|$ and $|xa\rangle\langle xa|$ for all standard basis states x. We can also implement illegal pairs and illegal single-site states for a subset of the tracks by repeating the above process with all single-site states in which the relevant subset appears. We now define legal and illegal states (definition 35 from [CPW15b]).

Definition 23 (Legal and Illegal States)**.** *We call a standard basis state* legal *if it does not contain any illegal pairs, and* illegal *otherwise.*

At this point, it is necessary to introduce another concept from computer science: regular expressions. A regular expression on some set of symbols Σ defines a subset of Σ^* (a formal language). If E is a regular expression, we will use $L(E)$, the "regular set", to denote the language defined by E. Regular expressions are built using a recursive process with three basis steps:

- The regular expression $\emptyset$ represents the empty language: $L(\emptyset) = \emptyset$.
- The regular expression ϵ represents the language containing just the empty string: $L(\epsilon) = \{\epsilon\}$.
- For $\sigma \in \Sigma$, the regular expression σ represents the language containing just the string σ: $L(\sigma) = \{\sigma\}$.

Regular expressions can be combined using several operations:

- If E and F are regular expressions, then $L(E + F) = L(E) \cup L(F)$.
- If E and F are regular expressions, then EF represents the concatenation of $L(E)$ and $L(F)$, i.e. $L(EF) = L(E)L(F) = \{ef \mid e \in L(E), f \in L(F)\}$.
- If E is a regular expression, then E^* represents the closure of $L(E)$, i.e. $L(E^*) = L(E)^*$. Recall that the closure operation A^* generates all strings that concatenate a finite number (possibly zero) elements of A.

Parentheses are used to indicate precedence where necessary. We will use the common shorthand $[E_1 E_2 \cdots E_n]$ to refer to $E_1 + E_2 + \cdots + E_n$. We say that an element $s \in \Sigma^*$ *matches* a regular expression E if $s \in L(E)$. See a textbook on formal language theory (such as [HMU06]) for more information.

This allows us to define bracketed states (definition 34 from [CPW15b]):

Definition 24 (Bracketed state)**.** *We call standard basis states that match the regular expression* $\textcircled{<}\textcircled{\cdot}^*\textcircled{>}$ bracketed states. *Here,* $\textcircled{<}, \textcircled{>}$ *stand for the states* $\left|\textcircled{<}\right\rangle_{\text{Tracks (1)-(3)}} \left|\textcircled{<}\right\rangle_{\text{Tracks (4)-(6)}}$ *and* $\left|\textcircled{>}\right\rangle_{\text{Tracks (1)-(3)}} \left|\textcircled{>}\right\rangle_{\text{Tracks (4)-(6)}}$ *respectively, and* $\textcircled{\cdot}$ *stands for any single-site state other than* $\textcircled{<}, \textcircled{>}$. *We denote by* S_{br} *the subspace spanned by the bracketed states.*

We will also make us of the following lemma, which is lemma 5.2 from [GI09], and lemma 36 from [CPW15b].

Lemma 19 (Regular Expressions)**.** *For any regular expression over the standard basis of single-site states in which each state appears at most once, we can use penalty terms to ensure that any legal standard basis state for the system is a substring of a string in the corresponding regular set.*

By only using regular expressions which start with a $\textcircled{<}$ and end with a $\textcircled{>}$, and restricting our attention to the space S_{br}, we can be sure that any legal state is in fact a complete string in the regular set, not just a substring. Actually enforcing that the ground states live in S_{br} will happen through the specific implementation of these chains into the grid $\mathscr{H}^{\Lambda(L)}$, which will be treated in the last chapters.

The other terms in the interaction Hamiltonian, besides the penalty terms, are the transition rule terms. Transition rule terms have the form $\frac{1}{2}(|\psi\rangle - |\varphi\rangle)(\langle\psi| - \langle\varphi|)$, where $|\psi\rangle, |\varphi\rangle$ are states on the same pair of adjacent sites. This term causes any state with amplitude on a configuration containing $|\psi\rangle$ to have a positive energy contribution, unless it has equal amplitude on a configuration containing $|\varphi\rangle$, and vice versa. Thus any zero-energy eigenstate has equal amplitude on configurations containing $|\psi\rangle$ and $|\varphi\rangle$. So these configurations are "linked", and we can think of these terms as implementing a transition rule $|\psi\rangle \to |\varphi\rangle$ (which we will call the "forwards" direction) or a transition rule $|\varphi\rangle \to |\psi\rangle$ (which we will call the "backwards" direction).

Note that this introduction of transition rules only creates linked sets of configurations, it does not actually implement any sort of time-dependence or order. This is instead achieved by being able to identify the configurations with a time because of their label in $\mathscr{H}_C$.

8.2 The oscillator track

The role of the oscillator track (1) is to "sweep" along the chain repeatedly. First to enforce the starting configuration for the counter TM and the QTM, and then to drive the evolution of both of these machines. The symbols for this track are:

- Two different "right sweeping arrows" $\overrightarrow{0}, \overrightarrow{1}$. The former will be used in the initialization sweep, and the latter in the subsequent sweeps.
- A number of different "left sweeping arrows" $\overleftarrow{0}, \overleftarrow{(i)}, \overleftarrow{1}$, for $i \in 1, \ldots, K$. Here, K is a constant which indicates how many cells should be left empty at the end of the tape during the initialization sweep. The arrows $\overleftarrow{0}$ and $\overleftarrow{(i)}$ are used in the initialization sweep, and the arrow $\overleftarrow{1}$ is used in the subsequent sweeps.
- Two empty symbols $\bigcirc, \bigcirc\!\!\bigcirc$. The former will be used for the space to the left of the arrow, and the latter will be used for the space to the right of the arrow. The reason there are two empty symbols is that this allows us to enforce that there is only one arrow.

Each of these symbols label orthonormal kets, and the Hilbert space $\mathscr{H}^{(i)}_{(1)}$ is the span of these.

The legal configurations for track (1) are the standard basis states matching the regular expression

$$\langle \bigcirc^* \left[\overrightarrow{0}\,\overrightarrow{1}\,\overleftarrow{0}\,\overleftarrow{(i)}\,\overleftarrow{1}\right] \bigcirc\!\!\bigcirc^* \rangle. \tag{8.5}$$

By lemma 19, we can enforce this using penalty terms.

8.3 The counter TM tracks

The role of the counter TM head and state track (2) is to keep track of the position of the head of the counter TM, as well as its internal state. The counter TM performs a step when the oscillator arrow (i.e. $\overrightarrow{1}$) sweeps right over the cell that currently contains the head and state of the TM. The symbols for this track are:

- Two empty symbols $\bigcirc, \bigcirc\!\!\bigcirc$. The former will be used for the space to the left of the head, and the latter will be used for the space to the right of the head. Just like with track (1), we use two different empty symbols to enforce that there is only one head.
- States $p \in P$, where P is the set of internal states of the counter TM. We partition P into $P = P_L \cup P_N \cup P_R$, according to the direction these states are entered from. This is a valid partitioning because the counter TM is unidirectional.
- Primed states $p' \in P'_R$, which duplicate the states in P_R. This duplication is required to make sure the counter TM performs only one step when the head is moved right, along with the sweeping arrow.

Again, each of these symbols label orthonormal kets, and the Hilbert space $\mathscr{H}^{(i)}_{(2)}$ is the span of these. We will denote the set of all states as $P' = P \cup P'_R$.

The legal configurations for this track are the standard basis states matching the regular expression

$$\langle \bigcirc^* p \bigcirc\!\!\bigcirc^* \rangle, \tag{8.6}$$

where $p \in P'$.

The role of the counter TM tape track (3) is to store the tape of the counter TM. The set of symbols for this track is simply the set of tape symbols $\Xi = \{\vdash, \#, 0, \ldots, \zeta - 1\}$ of the counter TM. The constant ζ depends on the size of the encoded QTM's alphabet Σ and set of internal states Q: $\zeta = |\Sigma \times Q|$. The only restriction for this track is that it should always be in a bracketed state.

8.4 Initialization sweep

During the initialization sweep, a $\overrightarrow{0}$ starts in the left-most cell, travels right until it reaches the $>$, then transitions into $\overleftarrow{(0)}$. This arrow then travels left, incrementing the counter by one at every step. Once it reaches K, it transitions into $\overleftarrow{0}$, which then continues travelling left until it reaches the $<$. We add the restriction $L - 3 \geq K$ to make sure that this entire evolution can indeed happen. When this is done, the arrow transitions into $\overrightarrow{1}$, and the initialisation sweep is done. This is implemented using the following transition rules on track (1):

$$\begin{aligned} &\overrightarrow{0}\,\bigcirc \to \bigcirc\,\overrightarrow{0}, \quad \overrightarrow{0}\,> \to \overleftarrow{(0)}\,>, \quad \bigcirc\,\overleftarrow{(i)} \to \overleftarrow{(i+1)}\,\bigcirc, \\ &\bigcirc\,\overleftarrow{(K)} \to \overleftarrow{(0)}\,\bigcirc, \quad \bigcirc\,\overleftarrow{0} \to \overleftarrow{0}\,\bigcirc, \quad <\,\overleftarrow{0} \to >\,\overrightarrow{1}. \end{aligned} \tag{8.7}$$

The initialisation sweep will force the counter TM tracks (2) and (3) to be in initial configurations matching the regular expressions

$$<\,p_\alpha\,\bigcirc^*\,> \text{ and } <\,\vdash \#^*\,>, \tag{8.8}$$

respectively. Here, p_α is the initial state of the counter TM. We do this by adding illegal pairs for anything other than the desired configuration. We add an illegal pair to forbid track (2) from being anything other than p_α when $\overrightarrow{0}$ is over it at the beginning of the chain:

$$\begin{array}{|c|c|}\hline < & \overrightarrow{0} \\ \hline & \neg p_\alpha \\ \hline \end{array}. \tag{8.9}$$

This diagram describes any state where $\left|<\right\rangle_{(1),(2)}$ is immediately to the left of $\left|\overrightarrow{0}\right\rangle_{(1)} \otimes |\varphi\rangle_{(2)}$, where $|\varphi\rangle$ has no support on $|p_\alpha\rangle$, and the tracks other than (1) and (2) can be anything. We will use this notation throughout this chapter. Anywhere but at the start of the chain, we forbid track (2) from being anything other than $\bigcirc$:

$$\begin{array}{|c|c|}\hline \neg< & \overrightarrow{0} \\ \hline & \neg\bigcirc \\ \hline \end{array}. \tag{8.10}$$

Similarly, we forbid track (3) from starting with anything other than $\vdash$:

$$\begin{array}{|c|c|}\hline < & \overrightarrow{0} \\ \hline & \neg\vdash \\ \hline \end{array}, \tag{8.11}$$

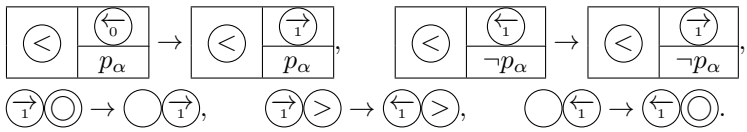
and being anything other than # in the other positions:

$$\begin{array}{|c|c|}\hline \neg< & \overrightarrow{0} \\ \hline & \neg\# \\ \hline \end{array}. \tag{8.12}$$

Because of these illegal pairs, any standard basis state containing a $\overrightarrow{0}$, $\overleftarrow{0}$ or $\overleftarrow{(i)}$ on track (1), but for which track (2) and (3) do not match regular expression (8.8), will evolve (either forwards or backwards) under the oscillator transition rules into an illegal configuration, in at most $2(L-2)$ steps (the number of steps the initialisation sweep takes). On the other hand, if tracks (2) and (3) do match the regular expression (8.8), then a standard basis state containing one of those arrows will evolve backwards until it reaches the initial configuration which has the form:

$$\begin{array}{|c|c|c|c|}\hline & \overrightarrow{0} & \bigcirc\cdots\bigcirc & \\ \cline{2-3} < & p_\alpha & \bigcirc\cdots\bigcirc & > \\ \cline{2-3} & \vdash & \#\ \ldots\ \# & \\ \hline \end{array}. \tag{8.13}$$

At the end of the initialisation sweep, the $\overleftarrow{0}$ will reach the left end of the track. At that point, it will turn around, transitioning into a $\overrightarrow{1}$, which then keeps travelling up and down the track. This is implemented by the following transition rules:



(8.14)

We refer to one complete cycle of the arrow moving up and down the track as one "oscillation".

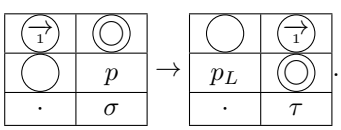
8.5 The counter TM

The counter TM will continually increment a base ζ-number (see section 8.3). To construct such a TM, we first take the Inc machine from lemma 13. We can easily modify this machine to work in base ζ instead of base 2. We also add a "start-of-tape" symbol $\vdash$. For the details, see equation (75) of [CPW15b]. We then modify this machine to make it loop forever. To this end, new initial and final states p_α and p_ω are added, and we make the original final state p_f transition back into the original initial state p_0. Some care must be taken to do all of this reversibly (see equation (76) of [CPW15b] for the details). The result of this is a well-formed, normal form, reversible base-ζ counter TM. When started from the tape configuration consisting of a $\vdash$ in the first cell followed by an all-blank tape (representing the number 0), this TM will loop indefinitely, incrementing the number written on the tape by 1 in each complete iteration.

Now we will add transition rules that will let the oscillator drive the computation of the counter TM. Let δ be the transition function for the counter TM, and recall that P and Ξ are the set of internal states and the alphabet of the counter TM respectively. For each transition $\delta(p, \sigma) = (\tau, p_N, N)$, where $p \in P, p_N \in P_N$ and $\sigma, \tau \in \Xi$, we add the following transition rules, which are described using our diagrammatic notation from before, now referring to tracks (1) to (3):

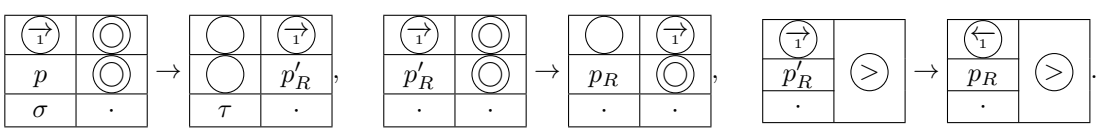
(8.15)

Here, $\cdot$ can refer to any state, and such a state is left unmodified by the transition. Similarly, for each transition $\delta(p, \sigma) = (\tau, p_L, L)$, where $p \in P, p_L \in P_L$ and $\sigma, \tau \in \Xi$, we add the following transition rule:



(8.16)

Note that because we are restricted to using pairs in our restriction rules (which ultimately stems from the fact that our Hamiltonian only has interaction terms between two neighbouring sites), we have to perform left-moving transitions as we approach the head, not as we move over it. Finally, for each transition $\delta(p, \sigma) = (\tau, p_R, R)$, where $p \in P, p_R \in P_R$ and $\sigma, \tau \in \Xi$, we add the following transition rules:



(8.17)

Note that we make use of an additional primed state for right-moving transitions. This is done to prevent a second calculation step from being performed immediately after a right-moving transition.

It is important to note that when these transition rules apply, they do so instead of the transition rules in (8.14) not in addition to.

Together, these transition rules ensure that whenever a $\overset{\rightarrow}{\underset{1}{}}$ sweeps along the head of the counter TM (i.e. the cell in track (2) which contains a state $p \in P$), exactly one step of the computation of the counter TM will be performed, in either one or two steps of the oscillator. This will continue until the counter TM runs out of space on the tape, at which points no transition rules will apply any more, and the TM stops.

We have to ensure that the chain of configurations (defined by transition rules) which corresponds to the evolution of the counter TM on its desired input is the only chain which does not contain any illegal states. To this end, we will declare certain configurations, which the counter TM will never enter when started on its desired input, to be illegal. We will declare these states illegal using our diagrammatic notation from before, now referring to tracks (2) and (3), unless stated otherwise.

First, since the counter TM will never re-enter the initial state p_α, we can declare all configurations with state p_α and the head not in the starting cell to be illegal:

$$\begin{array}{|c|c|}\hline \neg < & \begin{array}{c} p_\alpha \\ \hline \cdot \end{array} \\ \hline \end{array}. \tag{8.18}$$

Similarly, any configuration in state p_α with the head adjacent to a non-blank symbol will be declared illegal:

$$\begin{array}{|c|c|}\hline p_\alpha & \cdot \\ \hline \cdot & \neg\# \\ \hline \end{array}. \tag{8.19}$$

We will also never see the left sweeping arrow $\overset{\leftarrow}{\underset{1}{}}$ return to the beginning over the state p_α, so we also declare that configuration illegal:

$$\begin{array}{|c|c|}\hline < & \begin{array}{c} \overset{\leftarrow}{\underset{1}{}} \\ \hline p_\alpha \end{array} \\ \hline \end{array}. \tag{8.20}$$

Note that the diagram above applies to tracks (1) and (2), not track (2) and (3).

The state p_ω is never entered by the TM, so we declare any configuration in such a state to be illegal:

$$\begin{array}{|c|c|}\hline p_\omega & \cdot \\ \hline \cdot & \cdot \\ \hline \end{array}. \tag{8.21}$$

Next, we will note that the transition table for the counter TM (table (76) in [CPW15b]) is partial, just like table (6.8). Some of the transitions are never used, and were not defined. For each (p, σ) for which no transition rule is defined, we declare configurations in which the counter TM is in state p and the head is reading σ to be illegal:

$$\begin{array}{|c|c|}\hline p & \cdot \\ \hline \sigma & \cdot \\ \hline \end{array}. \tag{8.22}$$

Similarly, for each triplet $(p_L, \tau_L, L), (p_N, \tau_N, N)$ or (p_R, τ_R, R) which is not *entered* by any defined transition rule (i.e. which is not in the image of the partial transition function δ), we declare illegal pairs:

$$\begin{array}{|c|c|}\hline p_L & \cdot \\ \hline \cdot & \tau_L \\ \hline \end{array}, \qquad \begin{array}{|c|c|}\hline p_N & \cdot \\ \hline \tau_N & \cdot \\ \hline \end{array}, \qquad \begin{array}{|c|c|}\hline \cdot & p_R \\ \hline \tau_R & \cdot \\ \hline \end{array}. \tag{8.23}$$

Since the counter head never needs to move more than one cell to the right of a non-blank tape symbol, we also declare the following illegal pair for every $p \in P$:

$$\begin{array}{|c|c|}\hline \cdot & p \\ \hline \# & \# \\ \hline \end{array}. \tag{8.24}$$

The counter TM will never pad the numbers it writes with leading zeroes, so the tape never has a 0 to the left of a #. We therefore declare this combination to be illegal:

$$\begin{array}{|c|c|}\hline \cdot & \cdot \\ \hline 0 & \# \\ \hline \end{array}. \tag{8.25}$$

Similarly, the tape never has a 0 in the right-most cell:

$$\begin{array}{|c|c|}\hline \cdot & \\ \hline 0 & > \\ \hline\end{array}. \tag{8.26}$$

Also note that the primed states p'_R which were required to implement the right-moving transitions (see equation (8.17)) only ever appear below $\overset{\rightarrow}{1'}$, so we can declare all other appearances of a primed state illegal:

$$\begin{array}{|c|c|}\hline \neg\overset{\rightarrow}{1'} & \cdot \\ \hline p'_R & \cdot \\ \hline\end{array}. \tag{8.27}$$

Finally, when started from the desired input, the tape of the counter TM will always match the following regular expression:

$$\vdash [0\ 1 \ldots (\zeta - 1)]^* \#^*. \tag{8.28}$$

We can enforce this regular expression using illegal pairs by invoking lemma 19.

All these restrictions together ensure the validity of the following.

Definition 25 (Well-formed state)**.** *We say that a standard basis state on tracks (1) to (3) is* well-formed *if it is a bracketed state, its track (1) configuration matches the regular expression* (8.5)*, and its track (2) configuration matches the regular expression* (8.6)*.*

We have already ensured that any standard basis state in S_{br} that does not match the desired regular expressions has an energy penalty of at least 1. Thus only well-formed states can have zero energy. We also have the following result:

Lemma 20 (Well-formed transitions)**.** *For any well-formed standard basis-state, at most one transition rule applies in the forward direction, and at most one in the backwards direction. Furthermore, the set of well-formed states is closed under the transition rules.*

Proof. See lemma 39 of [CPW15b]. □

Now let $|\varphi_0\rangle$ denote the standard basis state of track (1) to (3) described in equation (8.13). Let $|\varphi_t\rangle$ be the standard basis state obtained by applying t transitions (in the forwards direction) to $|\varphi_0\rangle$, which is well defined thanks to lemma 20. Now the following result holds.

Lemma 21 (Evolve-to-illegal)**.** *Evolving any $|\varphi_t\rangle$ forwards or backwards in time according to the transition rules will never reach an illegal state. All other well-formed basis states will evolve either forwards or backwards to an illegal configuration after $O(L^2)$ transitions.*

Proof. See lemma 40 of [CPW15b]. □

8.6 Encoding the QTM

Now that the counter TM has been fully encoded, we can turn our attention to encoding the actual QTM. This will involve the tracks (4) to (6), the $\mathscr{H}_Q$ chain. This encoding will be very similar to the encoding of the counter TM which has been described in the previous sections. Only this time, we will have to concern ourselves with superpositions of standard basis states, and the computation steps of the QTM will be driven by the *left* sweeping arrow $\overset{\leftarrow}{1}$ on the oscillator track, instead of the right sweeping arrow.

Track (4) will fulfil a role similar to that of track (2). Its set of symbols and the regular expression it is restricted to are defined in the same way, except this time using the set Q of internal states of the QTM, instead of the set P. We also duplicate the *left*-moving states $q'_L \in Q'_L$ instead of the right moving states, to account for the different direction of the sweeping arrow.

Track (5) will fulfil a role similar to that of track (3). Its set of symbols is simply the set of tape symbols Σ of the QTM. The only restriction on this track is that it is always bracketed.

We will require the transition rules for the QTM, which can now involve superpositions of standard basis states, to be unitary in the following sense: for each neighbouring i and j and each $a, b \in \mathscr{H}_C^{(i)}$ and $cd \in \mathscr{H}_C^{(j)}$, there are either:

- no transitions from ab to cd; or
- there exists a unitary U_{abcd} acting on $\mathscr{H}_Q^{(i)} \otimes \mathscr{H}_Q^{(j)}$ together with an orthonormal basis $\{|\psi_{abcd}^k\rangle\}_k$ for $\mathscr{H}_Q^{(i)} \otimes \mathscr{H}_Q^{(j)}$, both depending only on a, b, c, d, such that the transition rules from ab to cd are exactly $|ab\rangle\,|\psi_{abcd}^k\rangle \to |cd\rangle\, U_{abcd}\,|\psi_{abcd}^k\rangle$.

Put simply: every transition must act classically on $\mathscr{H}_C$ and as a unitary on $\mathscr{H}_Q$.

Now let δ be the transition function of the QTM. We then add the following transition rule terms for all $q \in Q\backslash\{q_f\}$ (where q_f is the final state of the QTM) and $\sigma \in \Sigma$:

$$\left|\begin{array}{c|c} \bigcirc & \overleftarrow{1} \\ \hline q & \bigcirc \\ \hline \sigma & \cdot \end{array}\right\rangle \to \sum_{\tau, q_R} \delta_t(q, \sigma, \tau, q_R, R) \left|\begin{array}{c|c} \overleftarrow{1} & \bigcirc \\ \hline \bigcirc & q_R \\ \hline \tau & \cdot \end{array}\right\rangle + \sum_{\tau, q_N} \delta_t(q, \sigma, \tau, q_N, N) \left|\begin{array}{c|c} \overleftarrow{1} & \bigcirc \\ \hline q_N & \bigcirc \\ \hline \tau & \cdot \end{array}\right\rangle + \sum_{\tau, q_L} \delta_t(q, \sigma, \tau, q_L, L) \left|\begin{array}{c|c} \overleftarrow{1} & \bigcirc \\ \hline q'_L & \bigcirc \\ \hline \tau & \cdot \end{array}\right\rangle, \tag{8.29}$$

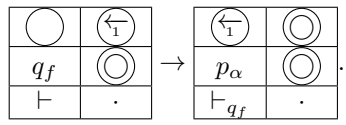
$$\left|\begin{array}{c|c} \bigcirc & \overleftarrow{1} \\ \hline \bigcirc & q'_L \\ \hline \cdot & \cdot \end{array}\right\rangle \to \left|\begin{array}{c|c} \overleftarrow{1} & \bigcirc \\ \hline q'_L & \bigcirc \\ \hline \cdot & \cdot \end{array}\right\rangle. \tag{8.30}$$

Here, the diagrams apply to tracks (1), (4) and (5), and we have written them in kets to emphasize that we are potentially working with superpositions. The sums are taken over $\tau \in \Sigma$ and $q_d \in Q_d$ with $d \in \{L, N, R\}$ depending on the sum. Note that no transitions apply when the head is in the last cell of the track. Thus the effective tape length is one less than the number of sites between the brackets.

We have not yet added any transition terms out of the final state q_f. There are also no transitions out of configurations trying to move the head over either end of the chain. Unfortunately, this violates the required unitarity conditions. Simply completing the transition map U to a unitary (in a manner similar to theorem 5), this will add additional transitions out of the final state q_f, and also out of configurations with the head at the beginning or end of the chain. This means that after reaching such a configuration, the QTM will continue to evolve in some arbitrary way, depending on the particular choice of the completion of U.

Instead, we will simply "run out the clock" after the QTM M has reached such a configuration by running another instance of the counter TM, but this time on tracks (4) and (6). This machine will certainly not halt or move of the end of the chain before the clock has run out, so this will solve the problem. To this end, we will dovetail the counter TM after the QTM M, and then encode this combined machine M' instead of the original QTM. The only caveat is that the transitions for the dovetailed counter TM will act on track (4) and (6) instead of track (4) and (5). We will also need to add the transitions which govern the transition from M to the counter TM (and the corresponding change in track). And to preserve reversibility, we will have to store the final state of M (either the halting state q_f, or the state when it attempted to write past the end of the chain). To do this, we add a symbol $\vdash_q$ to track (6) for each $q \in Q' = Q \cup Q'_L$. The counter TM will treat these subscripted symbols in the same way as the unsubscripted symbol $\vdash$.

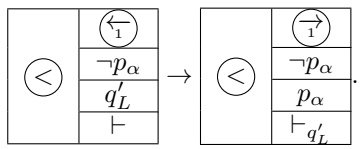
We first add a transition out of the final state q_f. We will transition into the initial state p_α of the counter TM, and write the final state to the subscript of the start-of-tape symbol on track (6). This transition rule applies to tracks (1), (4) and (6):



(8.31)

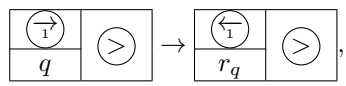
We can assume that q_f is above the start-of-tape symbol $\vdash$ because we will require the QTM M to behave properly.

We also add a transition out of configurations that try to move past the beginning of the chain. This transition rule applies to tracks (1), (2), (4) and (6):

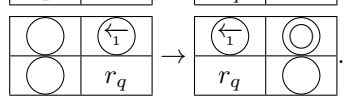


(8.32)

To deal with configurations that try to move past the end of the chain, we must first move back to the beginning of the chain. Note that such configurations are identified easily by the head being in the last cell of the track before the bracket. (recall that the effective chain length is one less). To facilitate moving back to the beginning of the chain, we add new states r_q for each $q \in Q$. Then we add the following transitions acting on tracks (1) and (4):

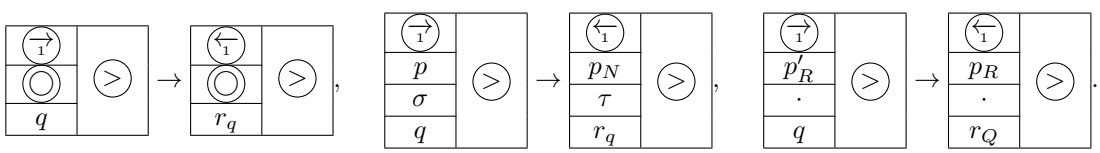


(8.33)



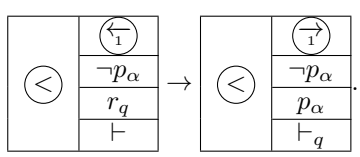
(8.34)

However, transition (8.33) overlaps with some of the transition rules on tracks (1)-(3) from (8.15) and (8.17). So instead, we add the following (composite) transitions acting on tracks (1), (2) and (4) and tracks (1), (2), (3) and (4):



(8.35)

After stepping to the beginning of the chain, the following transition on tracks (1), (2), (4) and (6) applies:



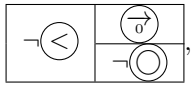
(8.36)

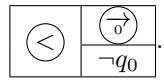
We have now covered all "edge-cases" that will arise during the evolution of the QTM, and the transition maps can now be arbitrarily completed to unitaries. For a proof that this is indeed possible, see the last part of section 7.6.2 in [CPW15b].

8.7 QTM initialisation sweep

All that remains is to ensure that tracks (4) to (6) are in the right initial configuration using the initialisation sweep. For track (4) we use the following penalty terms acting on track (1) and (4) to force the track to be

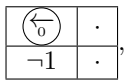
completely empty save for the initial state q_0 at the beginning of the track:

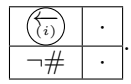


,


. (8.37)

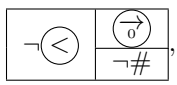
We will force track (5) to contain all 1's, except for K blank symbols at the end. The 1's are the unary input to the machine P_n from theorem 17, while the blank symbols accommodate for the space overhead of this machine (see part (ii) of theorem 17). This is done using the following penalty terms, acting on tracks (1) and (5):

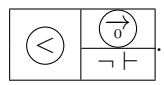


,


. (8.38)

Finally, track (6) is forced to be completely blank, save for a $\vdash$ at the beginning, using the following penalty terms acting on tracks (1) and (6):



,


. (8.39)

8.8 Formal statement

We can now present a formal statement of the results of this chapter with the following theorem, which is theorem 33 in [CPW15b].

Theorem 22 (Local Hamiltonian QTM encoding)**.** *Let* $\mathbb{C}^d = \Big(\text{span}\left\{\textcircled{<}, \textcircled{>}\right\} \oplus \mathbb{C}^C\Big) \otimes \Big(\text{span}\left\{\textcircled{<}, \textcircled{>}\right\} \oplus \mathbb{C}^Q\Big)$ *be the local Hilbert space of a 1-dimensional chain of length* L*.*

For any well-formed unidirectional Quantum Turing Machine $M = (\Sigma, Q, \delta)$ *and any constant* $K > 0$*, we can construct a two-body interaction Hamiltonian* h *on* $\mathbb{C}^d \otimes \mathbb{C}^d$ *such that the 1-dimensional translationally-invariant, nearest neighbour Hamiltonian* $H(L) = \sum_{i=1}^{L+1} h^{(i,i+1)}$ *on the chain of length* $L \geq K + 3$ *has the following properties:*

(i) d *depends only on the alphabet size and number of internal states of* M*.*

(ii) $h \geq 0$*, and the overall Hamiltonian* $H(L)$ *is frustration-free for all* L*.*

(iii) If we call $\mathscr{H}(L-2) := (\mathbb{C}^C)^{\otimes L-2} \otimes (\mathbb{C}^Q)^{\otimes L-2} =: \mathscr{H}_C \otimes \mathscr{H}_Q$ *and define* $S_{br} \subset \mathscr{H} = \text{span}\left(\left|\textcircled{<}\right\rangle \otimes \left|\textcircled{<}\right\rangle^1\right) \otimes \mathscr{H}(L-2) \otimes \text{span}\left(\left|\textcircled{>}\right\rangle \otimes \left|\textcircled{>}\right\rangle^L\right)$*, the unique ground state of* $H(L)|_{S_{br}}$ *is a computational history state encoding the evolution of* M *on input corresponding to the unary representation of the number* $L - K - 3$*, running on a finite tape segment of length* $L - 3$*.*

Moreover, if M *is proper on input given by the unary representation of* $L - K - 3$*, then:*

(iv) The computational history state always encodes $\Omega(|\Sigma \times Q|^L)$ *time-steps. If* M *halts in fewer than the number of encoded time steps, exactly one* ψ_t *has support on a state* $|\top\rangle$ *that encodes a halting state of the QTM. The remaining time steps of the evolution encoded in the history state leave* M*'s tape unaltered, and have zero overlap with* $|\top\rangle$*.*

(v) If M *runs out of tape within a time* T *less than the number of encoded time steps (i.e. in time-step* $T + 1$ *it would move its head before the starting cell or beyond cell* $L - 3$*), the computational history state only encodes the evolution of* M *up to time* T*. The remaining steps of the evolution encoded in the computational history state leave* M*'s tape unaltered.*

(vi) Finally, if M satisfies part (iii) of theorem 17, then h has the following form[9]

$$h = A + (e^{i\pi\varphi} B + e^{i\pi 2^{-|\varphi|}} C + h.c.), \tag{8.40}$$

with B and C being operators on $\mathbb{C}^d \otimes \mathbb{C}^d$, not dependent upon n, and with coefficients in $\mathbb{Z}$, and A being an operator on $\mathbb{C}^d \otimes \mathbb{C}^d$, not dependent upon n, and with coefficients in $\mathbb{Z} + \frac{1}{\sqrt{2}}\mathbb{Z}$.

A formal proof of this theorem can be found in chapter 7 of [CPW15b].

[9] h.c. stands for "Hermitian conjugate". It indicates that the Hermitian conjugate of the previous terms should be added, making the overall operator Hermitian.

Chapter 9

Robinson tiling

In this chapter we will treat the modified Robinson tiling which will ultimately be used to embed infinitely many chains (in the sense of chapter 8) of arbitrary length into the grid $\mathscr{H}^{\Lambda(L)}$. The Robinson tiling was originally introduced in 1971 by Raphael M. Robinson in 1971 ([Rob71]) to simplify Berger's proof that the Wang tiling problem is undecidable (also see chapter 4). The modified version which we will used is described in chapter 8 of [CPW15b], which this chapter is based on.

The Robinson tiling is a quasi-periodic tiling of the plane. It has a hierarchical structure of ever greater periodically repeating patterns. These patterns will be used to embed ever larger chains into the grid, and the hierarchical structure is exploited to prove certain rigidity results. These rigidity results ensure that we cannot circumvent the energy penalty of a halting encoded QTM by intentionally introducing errors into the tiling, without incurring an even greater energy penalty for these errors.

9.1 Robinson's tiles

In this section we will describe the tiling as introduced by Robinson in [Rob71], mirroring section 8.1 from [CPW15b].

Robinson's tiling consists of two layers. Intuitively, we will "glue" certain tiles from layer 1 on top of tiles from layer 2, and the resulting tile will fit into the grid only if both tiles fit into their respective layers. Mathematically, we have sets of tiles $\mathcal{T}_1$ and $\mathcal{T}_2$, and a set $\mathcal{T} \subset \mathcal{T}_1 \times \mathcal{T}_2$. We also have sets of colours $\mathcal{C}_1$ and $\mathcal{C}_2$, and colouring functions $c_k : \mathcal{T}_k \times \{R, U, L, D\} \to \mathcal{C}_k$, for $k \in \{1, 2\}$, that assign colours to each side of the tiles in each layer. We now define the set of colours for the combined tiling to be $\mathcal{C}_1 \times \mathcal{C}_2$. The corresponding colouring function is given by

$$\begin{aligned} c : \mathcal{T} \times \{R, U, L, D\} &\to \mathcal{C}_1 \times \mathcal{C}_2 \\ \big((t_a, t_b), s\big) &\mapsto \big(c_1(t_a, s), c_2(t_b, s)\big). \end{aligned} \tag{9.1}$$

The first layer (or main layer) is based on the tiles shown in figure 9.1. We also add all possible reflections

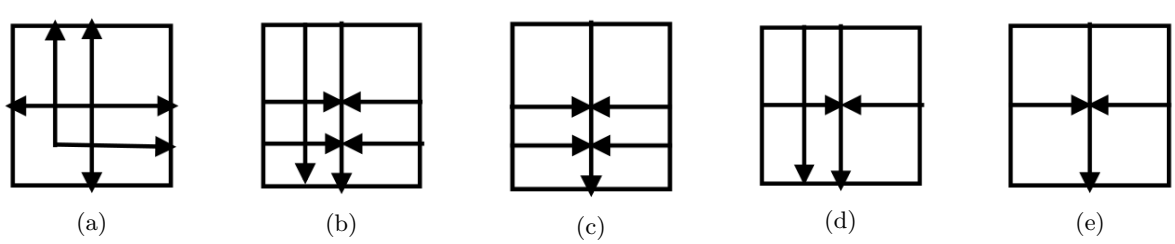
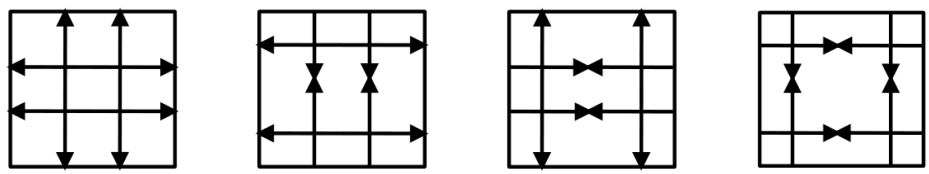


Figure 9.1: The five basic tiles used in the first layer of Robinson's tiling (figure 9 from [CPW15b]).



(a) (b) (c) (d)

Figure 9.2: The parity markings used in Robinson's tiling (figure 10 from [CPW15b]).

and rotations of these five tiles. Two tiles can be placed side-to-side if arrow heads meet arrow tails. Equivalently, we can define a unique "colour" for each configuration of arrow heads and tails on a right/bottom edge and arrow tails and heads (note the reversal) on a left/top edge. Tile 9.1a is called a "cross". We refer to crosses by their directions. The cross shown in figure 9.1 is called an up/right cross. Tiles 9.1b-9.1e are called "arms", and are referred to by the direction of their only complete central arrow. All arms drawn in figure 9.1 are down arms.

We now further enlarge this set of tiles by colouring the side arms red or green in the following manner:

- The non-central arrow (the hooked arrow) of a cross 9.1a is coloured either red or green.
- For the arms 9.1b - 9.1d, the non-central arrows must be coloured either red or green, with arrows in the same orientation (horizontal/vertical) using the same colour. For arm 9.1b one colour must be used horizontally and the other vertically.
- Arm 9.1e remains colourless.

Of course, tiles can only be place next to each other when the colours of touching arrows (head to tail) match.

This brings the number of tiles of each of the basic tile types from figure 9.1 to:

(a) 2 colourings and 4 orientations, 8 in total.

(b) 2 colourings and 8 orientations, 16 in total.

(c) 2 colourings and 4 orientations, 4 in total.

(d) 2 colourings and 8 orientations, 16 in total.

(e) 1 colouring and 4 orientations, 4 in total.

For a total of 48 different tiles in the first layer.

The second layer (or parity layer) will be used to enforce parity restrictions. The tiles used for this layer are shown in figure 9.2. For this layer we do not introduce the reflections and rotations of the tiles (this matters only for tiles 9.2b and 9.2c), nor do we introduce colourings; we only use the four tiles shown in the figure.

Note that this second layer will admit (only) a very simple tiling of the plane. Up to a translation of 1 tile in either the horizontal or vertical direction, the following pattern will occur: at odd x and odd y coordinates (odd-odd) we will find tile (a). At odd-even positions, we will find tile (b). At even-odd positions, we will find tile (c). And finally, at even-even position, we will find tile (d).

The two layers are now combined as follows:

- Parity marking 9.2a will be combined only with green crosses.
- Parity marking 9.2b will be combined only with horizontal arms (i.e. left arms or right arms) that do not have red "in-arrows". That is, horizontally oriented versions of tiles 9.1b-9.1e, where (c) and (d) have to be coloured green, and (b) has to be coloured such that the complete non-central arm is green, and the incomplete non-central arms are red.

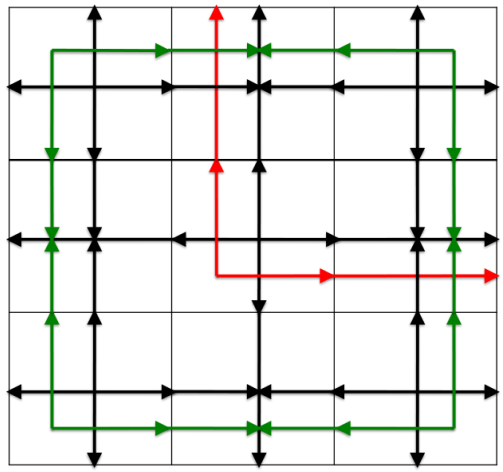


Figure 9.3: A 3×3 section of the Robinson tiling. The red cross in the centre has been arbitrarily chosen to face up/right. (Taken from figure 12 of [CPW15b].)

- Parity marking 9.2c will be combined only with vertical arms that do not have red "in-arrows".
- Parity marking 9.2d will be combined with all tiles from the first layer.

Some simple arithmetic will show that this results in a total of 68 tiles in the combined set $\mathcal{T} \subset \mathcal{T}_1 \times \mathcal{T}_2$.

Note that the way the parity markings are combined with the main layer is slightly different from that in [CPW15b]. The above has been taken directly from [Rob71] which [CPW15b] seems to have incorrectly cited. The rules which are used by [CPW15b] allow 9.2b and 9.2c to be combined only with green horizontal arms and green vertical arms respectively.

9.2 Robinson's tiling

We will now briefly describe the pattern that emerges when these tiles are used to tile the plane. A full analysis of this tiling can be found in [Ber66]. Because of the parity markings, green crosses must appear in alternating columns and rows. Any such cross will "radiate out" two green lines. These lines will be carried over an arm in the adjacent tiles, and then hit another green cross. The only valid configuration which is allowed is then a 3×3 square with green crosses in the corners facing each other. This will force the arms in the middles of the edges of this 3×3 square to face outwards. This in turn forces a cross to appear in the middle of the 3×3 square. This cross cannot be green, since this would force two green lines to cross each other in a single tile (this is not possible because tile 9.1b always has two different colours). The middle square is thus red. The only thing which is not forced is the orientation of the red cross. The resulting configuration is shown in fig 9.3. Such 3×3 squares repeat themselves (with a period of 4) over the grid. In a grid of 4 such squares (a 7×7 square), the red crosses will now be forced to face each other, and at the centre of the square, another cross will be forced. This time it is forced to be green, but we are again left free to choose its orientation. The resultant configuration is shown in figure 9.4.

We can keep going like this and get larger and larger squares of alternating colours periodically repeating over the plane. These squares are of size $(2^n - 1) \times (2^n - 1)$ repeating with period 2^n both horizontally and vertically, for $n \geq 2$. We will refer to the largest complete coloured square in a $(2^n - 1) \times (2^n - 1)$ square as a 2^{n-1}-border, and to its top edge as a 2^{n-1}-segment. So, the 8 edge tiles in figure 9.3 are a 2-border, and the 16 tiles with a red line in figure 9.4 are a 4-border. Their top edges are 2-segments and 4-segments respectively.

Depending on how we choose the orientation of the crosses (where the orientation is not forced), the tiling obtained in this way can either tile a quadrant of the plane, a half plane, or the complete plane. So a

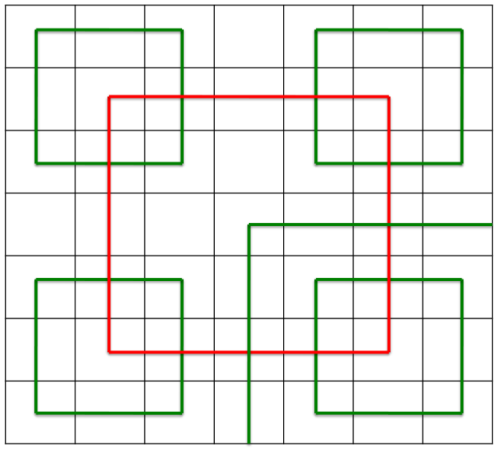


Figure 9.4: A 7×7 section of the Robinson tiling. The green cross in the center has been arbitrarily chosen to face down/right. Here, only the coloured lines of each of the tiles are shown. (Taken from figure 12 of [CPW15b].)

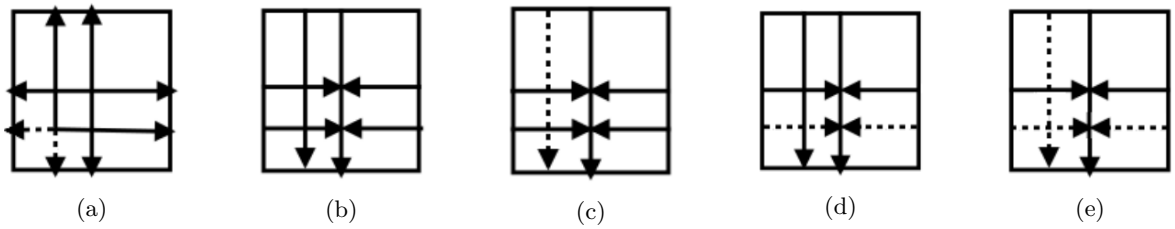


Figure 9.5: The basic tiles used in the modified Robinson tiling (figure 13 from [CPW15b]).

complete tiling of the plane can be built up from either 1, 2, 3 or 4 of such "sub-tilings". Two half tilings (which are built up from either 1 or 2 sub-tilings) can be shifted relative to each other by any even number of cells. The division line between such shifted half tilings will be called a *fault*.

9.3 Modified Robinson tiles

We will now discuss a modification for the Robinson tiles that prevents the faults described in the previous section. This section is based on section 8.2 from [CPW15b].

To remove the translation-freedom along fault lines, we add dashed side-arrows wherever the basic tiles from the Robinson tiling (figure 9.1) is missing a side arrow (see figure 9.5). We then extend to the complete set of tiles by adding rotations and translations, colours and parity markings just as with the normal Robinson tiles. The dashed arrows will not be coloured. Of course, dashed arrows must meet up for adjacent tiles (i.e. dashed heads must meet dashed tails).

Since we have only added restrictions, any valid tiling using the modified Robinson tiles gives rise to a valid tiling using the original Robinson tiles by simply removing the dashed arrows. Furthermore, any tiling using the original tiles that does not have a fault line (that is, a tiling where, for every n, the 2^n borders repeat with period 2^{n+1} both horizontally and vertically throughout the entire plane) can be made into a valid tiling of the modified tiles by adding dashed side-arrows.[10]

[10] This can be visualized as overlaying a dashed pattern on the existing pattern, the dashed pattern being obtained by "flipping" every 2^n-border "inside out". That is, rotating every corner by 180 degrees and completing the pattern these corners force.

On the other hand, if the tiling does contain a fault line, it can not be completed to a modified tiling by adding dashed arrows. To see this, assume without loss of generality that the fault line is vertical. Then there must exist two back-to-back (i.e. in the same row but not facing each other) crosses on either side of the fault line, with one pointing up and the other pointing down. Both crosses will "radiate out" a dashed line towards the fault line. The dashed line originating from the upward facing cross will run along the bottom halves of the tiles, while the one originating from the downward facing cross will run along the top halves of the tiles. Since there is no tile which can join these two lines together, this cannot lead to a valid tiling.

9.4 Rigidity

The modified Robinson tiles have a very "rigid" structure, in the sense that a tiling with a finite number of defects (i.e. non-matching adjacent tiles) will only differ from a completely valid tiling in a finite number of places in a ball around each of the defects. This will be made rigorous by the following lemmas, which will be stated without proof. For the proofs, please refer to lemmas 47 to 49 in section 8.2 of [CPW15b].

Lemma 23 (Robinson rigidity)**.** *Consider a connected region of a 2D square grid made up of square blocks of size $2^{n+1} \times 2^{n+1}$. Any tiling of such a region with modified Robinson tiles must contain a periodic pattern of 2^n-borders (some of which may be incomplete due to boundaries), repeating horizontally and vertically with period 2^{n+1}. That is, the tiling must contain the same periodic pattern of 2^n-borders as a section of the tiling of the infinite plane, up to translation and defects that do not affect the 2^n-borders.*

Lemma 24 (Segment bound)**.** *The minimum number of 2^n-segments in a tiling of an $L \times H$ rectangle (width L, height H) using modified Robinson tiles is $\lfloor H/2^{n+1} \rfloor (\lfloor L/2^{n+2} \rfloor - 1)$, and this minimum can be attained simultaneously for all n.*

Lemma 25 (Segment rigidity)**.** *In any tiling of an $L \times H$ rectangle (width L, height H) with d defects using modified Robinson tiles, the total number of 2^n-segments is at least $\lfloor H/2^{n+1} \rfloor (\lfloor L/2^{n+2} \rfloor - 1) - 2d$.*

Chapter 10

Undecidability of the spectral gap for constrained local dimension

All the ingredients for our final result have now been laid out, and it is time to combine them to complete the construction. This chapter is based on chapter 9 from [CPW15b]. Since the focus of this thesis is to give an overview of the construction, and not to give detailed formal proofs, we will state the results from that chapter without proof.

We will first introduce another definition which we will use in the results that follow.

Definition 26 (Gottesman-Irani Hamiltonian)**.** *Let $\mathbb{C}^Q$ be a finite-dimensional Hilbert space with two distinguished orthogonal states labelled $\left|\textcircled{<}\right\rangle$ and $\left|\textcircled{>}\right\rangle$. A* Gottesman-Irani Hamiltonian *is a 1D, translationally-invariant, nearest neighbour Hamiltonian $H_q(r)$ on a chain of length r with local interaction h_q on $\mathbb{C}^Q \otimes \mathbb{C}^Q$, which satisfies the following properties:*

(i) $h_q \geq 0$.

(ii) $[h_q, \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right| \otimes \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right|] = [h_q, \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right| \otimes \left|\textcircled{>}\right\rangle\left\langle\textcircled{>}\right|] = [h_q, \left|\textcircled{>}\right\rangle\left\langle\textcircled{>}\right| \otimes \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right|] = [h_q, \left|\textcircled{>}\right\rangle\left\langle\textcircled{>}\right| \otimes \left|\textcircled{>}\right\rangle\left\langle\textcircled{>}\right|] = 0$.

(iii) $\lambda_0(r) := \lambda_0(H_q(r)|_{S_{br}} < 1$, *where S_{br} is the subspace of states with fixed boundary conditions $\left|\textcircled{<}\right\rangle, \left|\textcircled{>}\right\rangle$ at the left and right ends of the chain, respectively.*

(iv) $\forall n \in \mathbb{N} : \lambda_0(4^n) \geq 0$, *and* $\sum_{n=1}^{\infty} \lambda(4^n) < 1/2$.

(v) $\lambda_0(H_q(r)|_{S_<} = \lambda_0(H_q(r)|_{S_>}) = 0$, *where $S_<$ and $S_>$ are the subspaces of states with, respectively, a $\left|\textcircled{<}\right\rangle$ at the left of the chain or a $\left|\textcircled{>}\right\rangle$ at the right end of the chain.*

Lemma 26 (Tiling + quantum layers)**.** *Let h_c^{row}, h_c^{col}, acting on $\mathbb{C}^C \otimes \mathbb{C}^C$, be the local interactions of a 2D tiling Hamiltonian H_c, with two distinguished states (tiles) $|L\rangle, |R\rangle \in \mathbb{C}^C$. Let h_q, acting on $\mathbb{C}^Q \otimes \mathbb{C}^Q$, be the local interaction of a Gottesman-Irani Hamiltonian $H_q(r)$. Then there is a Hamiltonian on a 2D square lattice with nearest-neighbour interactions h^{row}, h^{col}, acting on $\mathbb{C}^{C+Q+1} \otimes \mathbb{C}^{C+Q+1}$, with the following properties: For any region of the lattice, the restriction of the Hamiltonian to that region has an eigenbasis of the form $|T\rangle_c \otimes |\psi\rangle_q$, where $|T\rangle_c$ is a product state representing a classical configuration of tiles. Furthermore, for any given $|T\rangle_c$, the lowest energy choice for $|\psi\rangle_q$ consists of a ground state of $H_q(r)$ on segments between sites in which $|T\rangle_c$ contains an $|L\rangle$ or $|R\rangle$ and the boundary of the region, and $|0\rangle$'s everywhere else.*

We will outline the construction of the Hamiltonian whose existence the lemma asserts. For a full proof, please see lemma 51 in [CPW15b]. The construction and its motivation are given at the beginning of the proof for that lemma:

The idea is to sandwich the two Hamiltonians H_c and H_q together in two "layers", so that the overall Hamiltonian acts as H_c on the c-layer, with constraints between the layers that force low-energy configurations of the q-layer to be in the auxiliary $|0\rangle$ "blank" state, *except* between pairs of $|L\rangle$ and $|R\rangle$ states appearing in the same row of the c-layer, where the q-layer acts like H_q on that line segment.

To this end, define the local Hilbert space to be $\mathscr{H} := \mathscr{H}_c \otimes (\mathscr{H}_e \oplus \mathscr{H}_q) \simeq \mathbb{C}^C \otimes (\mathrm{span}\{|0\rangle\} \oplus \mathbb{C}^Q)$. The Hamiltonian H is defined in terms of the two-body interactions as follows:

$$
\begin{aligned}
h^{\mathrm{col},(j,j+1)} =& h_c^{\mathrm{col}} \otimes \mathbb{1}_{eq}^{(j)} \otimes \mathbb{1}_{eq}^{(j+1)} && (10.1)\\
h^{\mathrm{row},(i,i+1)} =& h_c^{\mathrm{row}} \otimes \mathbb{1}_{eq}^{(i)} \otimes \mathbb{1}_{eq}^{(i+1)} && (10.2)\\
&+ \mathbb{1}_c^{(i)} \otimes \mathbb{1}_c^{(i+1)} \otimes h_q && (10.3)\\
&+ |L\rangle\langle L|_c^{(i)} \otimes \left(\mathbb{1}_{eq} - \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right|\right)^{(i)} \otimes \mathbb{1}_{ceq}^{(i+1)} && (10.4)\\
&+ (\mathbb{1}_c - |L\rangle\langle L|_c)^{(i)} \otimes \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right|^{(i)} \otimes \mathbb{1}_{ceq}^{(i+1)} && (10.5)\\
&+ \mathbb{1}_{ceq}^{(i)} \otimes |R\rangle\langle R|_c^{(i+1)} \otimes \left(\mathbb{1}_{eq} - \left|\textcircled{>}\right\rangle\left\langle\textcircled{>}\right|\right)^{(i+1)} && (10.6)\\
&+ \mathbb{1}_{ceq}^{(i)} \otimes (\mathbb{1}_c - |R\rangle\langle R|_c)^{(i+1)} \otimes \left|\textcircled{<}\right\rangle\left\langle\textcircled{<}\right|^{(i+1)} && (10.7)\\
&+ \mathbb{1}_c^{(i)} \otimes |0\rangle\langle 0|_e^{(i)} \otimes |R\rangle\langle R|_c^{(i+1)} \otimes \mathbb{1}_{eq}^{(i+1)} && (10.8)\\
&+ |L\rangle\langle L|_c^{(i)} \otimes \mathbb{1}_{eq}^{(i)} \otimes \mathbb{1}_c^{(i+1)} \otimes |0\rangle\langle 0|_e^{(i+1)} && (10.9)\\
&+ \mathbb{1}_c^{(i)} \otimes |0\rangle\langle 0|_e^{(i)} \otimes (\mathbb{1}_c - |L\rangle\langle L|_c)^{(i+1)} \otimes (\mathbb{1}_{eq} - |0\rangle\langle 0|_e)^{(i+1)} && (10.10)\\
&+ (\mathbb{1}_c - |R\rangle\langle R|_c)^{(i)} \otimes (\mathbb{1}_{eq} - |0\rangle\langle 0|_e)^{(i)} \otimes \mathbb{1}_c^{(i+1)} \otimes |0\rangle\langle 0|_e^{(i+1)}, && (10.11)
\end{aligned}
$$

where $\mathbb{1}_c$, $\mathbb{1}_{eq}$ and $\mathbb{1}ceq$ are the identity operators on the corresponding Hilbert spaces. The Hamiltonian can be understood as follows. (10.4) and (10.5) force a $\left|\textcircled{<}\right\rangle$ in the q-layer whenever there is an $|L\rangle$ in the c-layer. (10.6) and (10.7) do the same with $\left|\textcircled{>}\right\rangle$ and $|R\rangle$. (10.8) and (10.9) force non-blank to the left and right of an $|R\rangle$ or $|L\rangle$, respectively. Finally, (10.10) and (10.11) force a non-blank to the left and right of any other non-blank in the q-layer, except when a non-blank coincides with an $|L\rangle$ or $|R\rangle$ in the c-layer. [CPW15b, p. 126]

We now introduce the modified Robinson tiling for our tiling Hamiltonian to obtain the following result.

Lemma 27 (Robinson + Gottesman-Irani Hamiltonian)**.** *Let h_c^{row} and h_c^{col}, acting on $\mathbb{C}^C \otimes \mathbb{C}^C$ be the local interactions of the tiling Hamiltonian associated with the modified Robinson tiles. For a given ground state configuration (tiling) of H_c, let $\mathcal{L}$ denote the set of all horizontal line segments of the lattice that lie between the down/right-facing and down/left facing red crosses (inclusive). Let h_q, acting on $\mathbb{C}^Q \otimes \mathbb{C}^Q$ be the local interaction of a Gottesman-Irani Hamiltonian $H_q(r)$.*

Then there is a Hamiltonian on a 2D square lattice of with L and height H with nearest-neighbour interactions h^{row}, h^{col}, acting on $\mathbb{C}^{C+Q+1} \otimes \mathbb{C}^{C+Q+1}$, such that, for any L, H, the ground state energy

$$\lambda_0(H^{\Lambda(L)}) = \min_{\mathcal{L}\in\Lambda(L)} \sum_{l\in\mathcal{L}} \lambda_0(|l|), \tag{10.12}$$

where the minimisation is over all valid tilings of the $L \times H$ rectangle.

Proof. This is essentially a straight-forward application of lemmas 26 and 25. See lemma 52 in [CPW15b] for the details. □

This then allows us to prove the following proposition. The proof of this statement is quite involved, and can be found as proposition 53 in [CPW15b].

Proposition 28 (Diverging ground state energy)**.** *There exists a family of interactions $h_u^{row}(n)h_u^{col}(n)$, acting on $\mathbb{C}^U \otimes \mathbb{C}^U$, and $h_u^{(1)}(n)$, acting on $\mathbb{C}^U$ with operator norm $\leq \frac{1}{2}$ and algebraic matrix entries, and strictly positive functions $\alpha_1^l(n), \alpha_(n), \delta(2(n), \alpha_1^u(n), \delta_1(n)$ (where the α functions are computable and the δ functions are uncomputable), such that either $\lambda_0(H_u^{\Lambda(L)}(n)) = -L\alpha_q^l(n) + \alpha_0(n)$, or $\lambda_0(H_u^{\Lambda(L)}(n)) = L^2\delta_2(n) - L[\alpha_1^u(n) + \delta_2(n)]$, but determining which is undecidable.*

Moreover, the interactions can be taken to have the following form: $h_u^{(1)}(n) = \alpha_2(n)\mathbb{1}$ with $\alpha_2(n)$ an algebraic computable number, $h_u^{row}(n)$ $\{0, \beta\}$-valued and independent of n and

$$h_u^{col}(n) = \beta\left(A + e^{i\pi\varphi}B + e^{i\pi 2^{-|\varphi|}}\right) + h.c., \tag{10.13}$$

where A, acting on $\mathbb{C}^U \otimes \mathbb{C}^U$ is independent of n and has coefficients in $\mathbb{Z} + \frac{1}{\sqrt{2}}\mathbb{Z}$, B and C, acting on $\mathbb{C}^U \otimes \mathbb{C}^U$, are independent of n and have coefficients in $\mathbb{Z}$, and $\beta \in \mathbb{Q}$ is independent of n. Recall that φ is defined as the rational number whose binary decimal expansion contains the digits of n in reverse order after the decimal.

The following corollary (corollary 54 in [CPW15b]) follows immediately by letting $L(n)$ be the minimal L in proposition 28 such that $-L\alpha_1^l(n) + \alpha_0(n) \leq 0$ and $L^2\delta_2(n) - L[\alpha_1^u(n) + \delta_1(n)] \geq 1$.

Corollary 29 (Undecidability of the ground state energy with promise)**.** *There exists a family of interactions $h_u^{row}(n), h_u^{col}(n)$, acting on $\mathbb{C}^U \otimes \mathbb{C}^U$, and $h_u^{(1)}(n)$, acting on $\mathbb{C}^U$, with operator norm $\leq \frac{1}{2}$ and algebraic matrix entries, and an (uncomputable) function $L(n)$, such that either $\lambda_0(H^\Lambda(L)_u(n)) \leq 0$ for all $L \geq L(n)$, or $\lambda_0(H^\Lambda(L)_u(n)) \geq 1$ for all $L \geq L(n)$, but determining which is undecidable. Moreover, the interactions can be taken to have the same form as in proposition 28.*

This finally allows us to prove our main result:

Theorem 30. *We construct explicitly a dimension d, $d^2 \times d^2$ matrices A, B, C, D and a rational number β so that*

(i) A is Hermitian and with coefficients in $\mathbb{Z} + \beta\mathbb{Z} + \frac{\beta}{\sqrt{2}}\mathbb{Z}$,

(ii) B, C have integer coefficients,

(iii) D is Hermitian and with coefficients in $\{0, 1, \beta\}$.

For each natural number n, define:

$$\begin{aligned} h_1(n) &= \alpha_2(n)\mathbb{1} \quad (\alpha_2(n) \text{ an algebraic number}) \\ h_{row}(n) &= D \quad (\text{independent of } n) \\ h_{col}(n) &= A + \beta\left(e^{i\pi\varphi}B + e^{-i\pi\varphi}B^\dagger + e^{i\pi 2^{-|\varphi|}}C + e^{-i\pi 2^{-|\varphi|}}C^\dagger\right). \end{aligned}$$

Then:

(i) The local interaction strength is ≤ 1. That is, all terms $h_1(n), h_{row}(n), h_{colo}(n)$ have operator norm bounded by 1.

(ii) If the UTM M halts on input n, then the associated family of Hamiltonians $\{H^{\Lambda(L)}(n)\}$ is gapped in the sense of definition 10 and, moreover, the gap $\gamma \geq 1$.

(iii) If the UTM M does not halt on input n, then the associated family of Hamiltonians $\{H^{\Lambda(L)}(n)\}$ is gapless in the sense of definition 11.

The construction of the described Hamiltonian is performed analogously to what is done in section 4.3, replacing the tiling Hamiltonian for the Hamiltonian obtained in corollary 29, but the proof is quite involved. Please refer to the proof of theorem 3 in section 9.2 of [CPW15b].

Bibliography

[Ber66] R. Berger. *The undecidability of the domino problem*. Memoirs of the Amer. Math. Soc. Vol. 66. 1966. ISBN: 9780821812662. DOI: 10.1090/memo/0066.

[BV97] E. Bernstein and U. Vazirani. "Quantum Complexity Theory". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1411–1473. DOI: 10.1137/S0097539796300921.

[CPW15a] T. S. Cubitt, D. Perez-Garcia, and M. M. Wolf. "Undecidability of the Spectral Gap". In: *Nature* 528 (Dec. 2015), pp. 207–211. DOI: 10.1038/nature16059. arXiv: 1502.04135 [quant-ph].

[CPW15b] T. S. Cubitt, D. Perez-Garcia, and M. M. Wolf. "Undecidability of the Spectral Gap (full version)". In: *ArXiv e-prints* (Feb. 2015). arXiv: 1502.04573 [quant-ph].

[Dir39] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35 (03 July 1939), pp. 416–418. ISSN: 1469-8064. DOI: 10.1017/S0305004100021162.

[FG+15] C. Fernández-González, N. Schuch, M. M. Wolf, J. I. Cirac, and D. Pérez-García. "Frustration Free Gapless Hamiltonians for Matrix Product States". In: *Communications in Mathematical Physics* 333 (Jan. 2015), pp. 299–333. ISSN: 1432-0916. DOI: 10.1007/s00220-014-2173-z. arXiv: 1210.6613 [quant-ph].

[GI09] D. Gottesman and S. Irani. "The Quantum and Classical Complexity of Translationally Invariant Tiling and Hamiltonian Problems". In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE. 2009, pp. 95–104. DOI: 10.1109/focs.2009.22. arXiv: 0905.2419 [quant-ph].

[HMU06] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to automata theory, languages, and computation*. 3rd ed. Pearson, July 2006. ISBN: 9780321455369.

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. 10th anniversary ed. Cambridge University Press, 2010. ISBN: 9781107002173.

[NO02] H. Nishimura and M. Ozawa. "Computational Complexity of Uniform Quantum Circuit Families and Quantum Turing Machines". In: *Theoretical Computer Science* 276.1-2 (Apr. 2002), pp. 147–181. DOI: 10.1016/s0304-3975(01)00111-6. arXiv: quant-ph/9906095.

[NO09] H. Nishimura and M. Ozawa. "Perfect Computational Equivalence between Quantum Turing Machines and Finitely Generated Uniform Quantum Circuit Families". In: *Quantum Information Processing* 8.1 (Jan. 2009), pp. 13–24. DOI: 10.1007/s11128-008-0091-8. arXiv: quant-ph/0511117.

[Rob71] R. M. Robinson. "Undecidability and nonperiodicity for tilings of the plane". In: *Inventiones Mathematicae* 12.3 (Sept. 1971), pp. 177–209. DOI: 10.1007/bf01418780.

[Sha94] R. Shankar. *Principles of quantum mechanics*. 2nd ed. Springer, 1994. ISBN: 9780306447907. DOI: 10.1007/978-1-4757-0576-8.

[Tur37] A. M. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society*. 2nd ser. 42.1 (1937), pp. 230–265. DOI: 10.1112/plms/s2-42.1.230.

[Wan61] H. Wang. "Proving Theorems by Pattern Recognition - II". In: *Bell system technical journal* 40.1 (Jan. 1961), pp. 1–41. DOI: 10.1002/j.1538-7305.1961.tb03975.x.